Compute (Bridgend) Ltd

SELCOPYi Reference and User Guide

Release 3.50

8 Merthyr Mawr Road, Bridgend, Wales UK CF31 3NH

Tel: +44 (1656) 65 2222
Fax: +44 (1656) 65 2227

CBL Web Site - www.cbl.com

This document may be downloaded from www.cbl.com/documentation.php

CBL Ref: z:\cd\sman\i350\SELCOPYi_3.50_Reference_and_User_Guide.pdf

Contents

Documentation Notes **1**

Summary of Changes **2**
- First Edition (October 2013) 2
- Second Edition (March 2015) 4
- Third Edition (December 2016) 5
- Fourth Edition (October 2018) 6
- Fifth Edition (July 2020) 7

About SELCOPYi **9**

Getting started with SELCOPYi **10**
- Starting the SELCOPYi program 10
- Security Considerations 10
  - VSE Systems 10
  - MVS Systems 11
- 3270 Terminal Emulation 11
  - 3270 Screen Sizes 11
  - Keyboard and Mouse Mapping 12
- Window Concepts 12
  - Window Hierarchy 12
  - Manipulating Windows 12
  - Window Format 13
  - Pressing Buttons 14
  - Window Focus 14
    - Input fields 14
    - Changing the focus window 15
  - Window Names 15
    - Viewing Window Names 15
  - Window Class 15
  - System Menu 16
  - Function Keys 16
    - Default Function Keys (KEYLISTs active) 16
    - Default Function Keys (KEYLISTs not active) 17
- SELCOPYi Main Window 18
  - SELCOPYi Main Window Menu Bar 18
- SELCOPYi Clipboard 19
- SELCOPYi Batch Execution (SDEAMAIN) 19
  - Supported Functionality 19
  - JCL Requirements 20
  - Environment Options 20
  - Rexx Macro Execution and Conditional Logic 20
- SELCOPYi Interactive Help 21
- SELCOPYi Help Topic Index Window 22
  - Searching the list of help topics 23
  - Help topic list fields 23
  - Selecting a help topic 23
  - Opening the help topic index window 23
- File Object Names 23
  - File/Dataset Name 24
  - Generic File Object Names 24
    - Generic z/OS Dataset Name 24
    - Generic z/OS HFS/ZFS fileids 25
    - Generic z/OS PDS/PDSE Library Member Names 25
    - Generic z/VM CMS File Names 25
- z/OS PDSE Library Member Generations 25
  - Generic z/OS PDSE Library Member Generations 26

Window Classes **27**
- Storage Display Windows 27
  - Storage Window Display Format 27
  - Storage Window Resizing 27
  - Storage Display Navigation 28
  - Storage Data Manipulation 28
- List Windows 28
  - List Window Status Bar 28
  - List Window Menu 29
    - View List Display 29
    - Field Descriptor Block (FDB) 29
    - Edit View 30
    - Zoom View 30
  - Selecting, Sorting and Filtering 31
    - SELECT Clause 32
    - WHERE Clause 32
    - SORT (ORDER BY) Clause 34
  - Sorting with the Cursor 34
  - List Entry Location 35

Contents

Window Classes
FIND Command....35
LOCATE Command....36
S Command....36
List Entry Display....36
EXCLUDE Command....37
FLIP Command....37
HIDE Command....38
MEMBER Command....38
ONLY Command....38
RESET Command....39
SHADOW Command....39
SRCHFOR Command....39
List Window Prefix Area....40
Interactive Panel Windows....41
Panel Window Format....42
Panel Window Size & Location....44
Panel Window Hierarchy....44
Panel Scrollable Display....44
Panel Window Views....44
Scrollable Input/Output Fields....44
Input Field Data Recall....45
Embedded Tables....45
Overview....45
Table Edit CLI (Primary) Commands....45
Table Edit Options....46
Table Edit Prefix (Line) Commands....46
Table Edit PFKeys....47
Table Editing Techniques....47
REFRESH Command....48
SELECTALL Command....48
SORT Command....49
VLMAX Command....49
VLMIN Command....50
CCOLOUR - SET/QUERY Option....50
COLHEADER - SET/QUERY Option....51
Selection Lists....51
Select Dataset Name (Catalog)....52
Select Dataset Name (VTOC)....52
Select HFS Path Name....53
Select Member Name....53

Primary Option Menu (=)....54
Primary Option Menu Panel....54
Menu Bar Items....54
Options....54
Panel Output Fields....54

Settings (=0)....55
Settings Panel....55
Options....55
Panel Input/Output Fields....55
Startup Settings (=0.1)....55
Panel Input Fields....55
System Settings (=0.2)....56
Panel Input Fields....56
Text Edit Settings (=0.3)....56
Panel Input Fields....56
Structured Data Edit (SDE) Settings (=0.4)....57
Options....57
Panel Input Fields....57
COBOL Compiler Options (=0.4.1)....58
Panel Input Fields....58
PL/1 Compiler Options (=0.4.2)....59
Panel Input Fields....59
HLASM Assembly Options (=0.4.3)....59
Panel Input Fields....60
Auxiliary Dataset Options (=0.4.4)....60
Panel Input Fields....60
Compiler work file allocation settings (=0.4.5)....60
Manage Copybook Associations (=0.4.6)....61
Panel Input Fields....61
Primary Commands....61
List Window Settings (=0.5)....62
Panel Input Fields....62
Batch Settings (=0.6)....62
Panel Input Fields....62

Contents

Settings (=0)
DB2 Settings (=0.7) ... 62
Options ... 63
Execute SQL Settings (=0.7.2) ... 63
Panel Input Fields ... 63
DB2 Subsystem Options Settings (=0.8) ... 63
Options ... 63
Search/Update Settings (=0.9) ... 64
Panel Input Fields ... 64

Text Edit (=1) ... 65
Text Edit Panel ... 65
Panel Input Fields ... 65

Data Edit (=2) ... 67
Structured Data Browse/Edit Panel ... 67
Menu Bar Items ... 67
Panel Input Fields ... 68
Structured Data Browse/Edit - Options Panel ... 70
Menu Bar Items ... 70
Panel Input Fields ... 70

List File Windows (=3) ... 72
List Menu Panel ... 72
Options ... 72
List DASD Volumes (=3.1) ... 72
Panel Input Fields ... 72
Prefix Line Commands ... 73
Columns Displayed ... 73
List VTOC Files (=3.2) ... 74
Panel Input Fields ... 75
Prefix Line Commands ... 75
Columns Displayed ... 76
List VTOC Extents (=3.3) ... 76
Panel Input Fields ... 77
Prefix Line Commands ... 77
Columns Displayed ... 78
Data Set List Utility (=3.4) ... 78
Panel Input Fields ... 79
List Catalog Entries (=3.5) ... 79
Panel Input Fields ... 80
Prefix Line Commands ... 81
Columns Displayed ... 82
List CMS Files ... 84
Panel Input Fields ... 84
Prefix Line Commands ... 84
Columns Displayed ... 85
List Dataset Details (=3.6) ... 85
Panel Input Fields ... 86
Prefix Line Commands ... 87
Columns Displayed ... 87
List Library Members (=3.7) ... 89
Panel Input Fields ... 90
Prefix Line Commands ... 90
Columns Displayed ... 91
List Library Member Generations ... 93
Panel Input Fields ... 94
Prefix Line Commands ... 94
Columns Displayed ... 94
List MVS Allocated Files (=3.8) ... 95
Panel Input Fields ... 96
Prefix Line Commands ... 96
Columns Displayed ... 97
List VSE Standard Labels ... 97
Panel Input Fields ... 97
Prefix Line Commands ... 98
Columns Displayed ... 98
List MVS Enqueues (=3.9) ... 98
Panel Input Fields ... 99
Prefix Line Commands ... 99
Columns Displayed ... 99
List MVS Job Enqueues (=3.10) ... 99
Panel Input Fields ... 100
Prefix Line Commands ... 100
Columns Displayed ... 100
List Associations (=3.11) ... 101
Panel Input Fields ... 101

Contents

List File Windows (=3)
Prefix Line Commands....102
Columns Displayed....103
List HFS Path (=3.12)....103
Panel Input Fields....103
Prefix Line Commands....104
Columns Displayed....104
List Storage Groups....105
Panel Input Fields....106
Prefix Line Commands....106
Columns Displayed....106
List Storage Group Volumes....107
Panel Input Fields....108
Prefix Line Commands....108
Columns Displayed....108

File Copy (=5)....110
Overview....110
Source and Target File Types....110
File Copy Panel....111
File Copy....111
Menu Bar Items....111
Panel Input Fields....112
File Reformat....116
Menu Bar Items....116
Panel Input Fields....116
Primary Commands....117
BROWSEINPUTCOPYBOOK....117
BROWSEINPUTFILE....117
BROWSEOUTPUTCOPYBOOK....117
BROWSEOUTPUTFILE....117
CMX....117
FILTER....118
MAP....118
JCL....118
RUN....118
SELECT....118
File Copy - PDS Copy Statistics....118
Summary Format....118
Copy Statistics Fields....119
Library Member Move/Copy....120
Menu Bar Items....120
Panel Input Fields....121
PFKeys....121

File Search/Update/Copy/Remap....122
Overview....122
Source File Types....122
Output Report....122
Unformatted File Search/Update/Copy....123
Formatted File Search/Update/Copy/Remap....124
File Search/Update/Copy/Remap Panels....125
Basic File Search Panel....126
FSU: Basic File Search Panel....126
Menu Bar Items....126
Panel Input Fields....127
Primary Commands....131
BROWSEINPUTFILE....131
CMX....131
FILTER....131
JCL....131
RUN....131
SELECT....131
Extended File Search/Update/Copy/Remap Panels....132
FSU: Extended File Search, Update, Copy & Remap Tasks....132
Menu Bar Items....132
Panel Options....132
FSU: Input File(s) Specification....135
Menu Bar Items....135
Panel Fields....135
FSU (unformatted): Record Selection....137
Menu Bar Items....137
Panel Fields....137
FSU (formatted): Record Selection....138
Menu Bar Items....138
Panel Fields....138
FSU (unformatted): Search records using WHERE expression....140

Contents

File Search/Update/Copy/Remap
Menu Bar Items....140
Panel Fields....140
FSU (formatted): Search records using WHERE expression....141
Menu Bar Items....141
Panel Fields....141
FSU (unformatted): Search Records using the FIND Command....142
Menu Bar Items....142
Panel Fields....142
FSU (formatted): Search Records using the FIND Command....143
Menu Bar Items....143
Panel Fields....143
FSU (unformatted): Change record data using the CHANGE command....145
Menu Bar Items....145
Panel Fields....145
FSU (formatted): Change record data using the CHANGE command....148
Menu Bar Items....148
Panel Fields....148
FSU: Copy Selected records to an Output File....151
Menu Bar Items....151
Panel Fields....151
FSU: Remap Fields in Selected records to an Output File....153
Menu Bar Items....153
Panel Fields....153
FSU: Copy Selected records to Output Library Members....155
Menu Bar Items....155
Panel Fields....155
FSU: Remap Fields in Selected records to Output Library Members....157
Menu Bar Items....157
Panel Fields....157
FSU: Report File....159
Menu Bar Items....159
Panel Fields....159
Primary Commands....161
BROWSEINPUTCOPYBOOK....161
BROWSEINPUTFILE....161
BROWSEOUTPUTCOPYBOOK....161
BROWSEOUTPUTFILE....161
CMX....161
FILTER....162
MAP....162
JCL....162
RUN....162
SELECT....162
File Selection Panels....163
Multiple Fileid Masks....163
Table View....163
Single Row (Zoomed) View....163
Input Values....164
Select Files to Process....166
Table View....166
Single Row (Zoomed) View....166
Input Values....167
Select Input Members....168
Table View....168
Single Row (Zoomed) View....168
Input Values....169
Select Input Members - FIND....170
Input Values....170
Primary Commands....172
FIND....172
RUN....172
SELECT....172
Generate Formatted Record Expression Panels....173
Major OR Sub-Expressions....173
Table View....173
Single Row (Zoomed) View....173
Input Values....174
AND Sub-Expressions....175
Table View....175
Single Row (Zoomed) View....175
Input Values....176
Minor OR Sub-Expressions....177
Table View....177
Single Row (Zoomed) View....177
Input Values....178
SDE: Select Record-Type....180

Contents

File Search/Update/Copy/Remap
SDE: Select Field Name....180
Primary Commands....181
SELECT....181
Multiple FIND & CHANGE Specification Panels....182
Multiple Find Commands (unformatted)....182
Table View....182
Single Row (Zoomed) View....182
Input Values....183
Multiple Find Commands (formatted)....184
Table View....184
Single Row (Zoomed) View....184
Input Values....185
Multiple Change Commands (unformatted)....186
Table View....186
Single Row (Zoomed) View....186
Input Values....187
Multiple Change Commands (formatted)....189
Table View....189
Single Row (Zoomed) View....189
Input Values....190
File Search/Update/Copy/Remap Output....192
Report Format....192
Record Type: Command....193
Record Type: Summary....193
Record Type: Hit....194
Record Type: IOError....196
Record Type: Alias....196
Record Type: Record....196
Function Keys....196
File Update Undo....198
Overview....198
File Update Undo Panel....199
File Update Undo Output....199
Report Format....199
Report Fields....199
Report Data....200
Summary Block....200
Sample Terse Report Output....201
Sample EXTENDED Output....202

Compare Files/Libraries Menu (=7)....204
Options....204
Compare Files (=7.1)....204
Overview....204
Source File Types....204
Output Report....204
Unformatted Compare....205
Formatted Compare....205
Hierarchical Compare....206
Record Synchronisation....206
Terminology....206
Overview....207
1-TO-1 Synchronisation....207
Read-Ahead Synchronisation....207
Key Synchronisation....208
Hierarchical Key Synchronisation....209
Compare Files Panels....211
Basic Unformatted Compare Panel....212
Compare Files: Basic Options....212
Compare Type - Formatted/Unformatted ?....214
Extended Unformatted Compare Panels....216
Compare Files (unformatted): New file details and options....216
Compare Files (unformatted): Old file details and options....217
Compare Files (unformatted): Re-synchronisation options....218
Compare Files (unformatted): Specify Key fields....221
Compare Files: Specify Key Pos/Len....223
Compare Files (unformatted): Output Files....223
Compare Files (unformatted): Options / Action....225
Formatted Compare Panels....226
Compare Files (formatted): New file details and options....226
Compare Files (formatted): Old file details and options....227
Compare Files (formatted): Re-synchronisation options....229
Compare Files (formatted): Specify Key fields....232
Compare Files: Specify Key Pos/Len....232
SELCOPYi Compare Files - KEY Columns (Record Types List)....233
SELCOPYi Compare Files - Select from Field Names List....234

Contents

Compare Files/Libraries Menu (=7)
Compare Files (formatted): Output Files....234
Compare Files (formatted): Options / Action....236
SELCOPYi Compare Files - Select Record Types to Compare....237
SELCOPYi Compare Files - Select Field Names to Compare....238
SELCOPYi Compare Files - Select Field Names List....239
Compare Files Output....239
Report Format....239
Record Type: Command....243
Record Type: Files....243
Record Type: Compare | Compare-record_type....243
Record Type: Field....245
Record Type: Summary....245
Record Type: Key....246
Record Type: Gap....246
Function Keys....246
Compare Libraries (=7.2)....246
Overview....246
Compare Libraries Panel....247
Menu Bar Items....247
Panel Input Fields....247
Compare Libraries Output....249

SELCOPY Debug & Development (=8.1)....250
SELCOPY Debug Startup....250
Supply JCL (=8.1.1)....250
JCL Job Input Panel....250
JCL Job Step Selection List Panel....251
JCL Job Panel Primary Commands....253
Supply SYSIN (=8.1.2)....253
Control Statement Dataset Input Panel....253
SELCOPY Debug Load Library Search Chain....255
SELCOPY Debug IMS/DL1....256
IMS/DL1 Restart....256
SELCOPY Debug Loop Break-in....257
SELCOPY Debug Windows....257
SELCOPY Debug Main window....257
SYSIN Window....258
SYSPRINT Window....259
SQL Log Window....260
WTO Log Window....260
Work Area/Current Input Record Window....261
POS Expression Window....261
@ Pointer Window....262
Columns Displayed....262
Equates Window....262
Columns Displayed....263
PCB Window....263
TRACE Window....263
Watch List Window....264
Watch List Columns....264
Watch List Prefix Area (Line) Commands....265
Add WATCH Var/PosExp Panel....266
Panel Input Fields....266
Operations List....266
Columns Displayed....267
Point-and-Shoot Popup Menu....267
SELCOPY Debug Commands....270
BREAKPOINT....270
EOJ....271
GO....271
LIST OPERATIONS....271
RERUN....272
STEPINTO....272
STEPOVER....272
TRACK....273
WATCH....274
WCOMMAND....277
WINDOW....278
SELCOPY Debug SET/QUERY/EXTRACT Options....279
BREAKIN - SET/QUERY/EXTRACT Option....279
DEBUGCOLOUR, DEBUGCOLOR - SET/QUERY/EXTRACT Option....280
SELCOPY Debug WATCH Sub-commands....281
BOTTOM....281
CLOSE....281
DELETE....281
DOWN....282

Contents

SELCOPY Debug & Development (=8.1)
INSERT....283
LEFT....283
LOCATE....284
POSWINDOW....285
RESET....285
RIGHT....285
SPACE....286
TOP....286
UP....287
SELCOPY Debug WATCH SET/QUERY/EXTRACT Options....288
COLOUR, COLOR - SET/QUERY/EXTRACT Watch List Option....288
COLUMN - SET/QUERY/EXTRACT Watch List Option....290
DATATYPE - SET/QUERY/EXTRACT Watch List Option....290
HEX - SET/QUERY/EXTRACT Watch List Option....291
POINT - SET/QUERY/EXTRACT Watch List Option....291
PREFIX - SET/QUERY/EXTRACT Watch List Option....292
SCALE - SET/QUERY/EXTRACT Watch List Option....293
SHADOW - SET/QUERY/EXTRACT Watch List Option....293
SELCOPY Debug Function Keys....294

Utilities Menu (=8)....295
Options....295
CBLVCAT Interactive (VCI) (=8.2)....295
CBLVCAT Interactive Window....295
Panel Input Fields....297
Prefix Commands....297
Columns Displayed....298
Raw Data Window....298
Prefix Line Commands....299
Columns Displayed....300
Execute IDCAMS (=8.3)....302
Panel Fields....302
Prefix Commands....302
Columns Displayed....302
Execute POWER....303
Panel Fields....303
Prefix Commands....303
Columns Displayed....304
Define Catalog ALIAS (=8.4)....305
Menu Bar Items....305
Panel Fields....305
Create Library ALIAS (=8.5)....306
Panel Fields....306
Execute IEBCOPY (=8.6)....307
Panel Fields....307
Favourite Datasets/Commands (=8.7)....309
Overview....309
Favourite Datasets/Commands Panel....309
Panel Input Fields....309
System Information Menu (=8.8)....312
Options....312
Operating System Window (=8.8.1)....312
Menu Bar Items....312
LPA Modules Window (=8.8.2)....313
Columns Displayed....313
Link List Window (=8.8.3)....313
Columns Displayed....314
APF List Window (=8.8.4)....314
Columns Displayed....314
Task List Window (=8.8.5)....315
Columns Displayed....315
Allocated Storage Windows....315
Columns Displayed....316
Loaded Programs Window (=8.8.6)....316
Columns Displayed....316
SELCOPYi Storage Statistics Window....317
Heap....317
Stack....317
Lists....317
SELCOPYi Module List Window....318
CBLVCAT SVC window (=8.8.7)....318
CBLNAME Window....319
About SELCOPYi (=8.8.8)....319
File Search (=8.9)....320
Panel Fields....320
Prefix Line Commands....321

Contents

Utilities Menu (=8)
Columns Displayed....321
Search for Library Members (=8.10)....322
Overview....322
Search for Library Members Panel....322
Menu Bar Items....323
Panel Input Fields....323
Search for Library Members Output....324
Calendar Window (=8.13)....325
Calculator Window (=8.14)....325
Create New Datasets Menu (=8.15)....326
Options....326
Allocate NonVSAM (=8.15.1)....326
Menu Bar Items....326
Define VSAM KSDS/ESDS/RRDS/LDS (=8.15.2/3/4/5)....327
Menu Bar Items....327
Define GDG Base (=8.15.7)....328
Menu Bar Items....328
Panel Input Fields....328
Generate XML (=8.16)....330
XML Generation Panels....330
Menu Bar Items....330
Panel Fields - Source Structured Data File....331
Panel Fields - Output XML Text File....333
Primary Commands....337
Function Keys....337
Select Record-Types Panel....338
Select Field Names Panel....338
Choose Record-Type....338
Select Field Names List....338
Generate CSV (=8.17)....339
CSV Generation Panels....339
Menu Bar Items....339
Panel Fields - Source Structured Data File....340
Panel Fields - Output CSV Text File....342
Primary Commands....343
Function Keys....344
Generate JSON (=8.18)....345
JSON Generation Panels....345
Menu Bar Items....345
Panel Fields - Source Structured Data File....345
Panel Fields - Output JSON Text File....348
Primary Commands....349
Function Keys....349
Merge Datasets....351
Merge Datasets Panel....351
Menu Bar Items....351
Panel Input Fields....351
Primary Commands....352
Function Keys....352
List/Delete PDSE v2 Orphaned Member Generations (=8.17)....353
Menu Bar Items....353
Panel Input Fields....353
Data Set Information....355
Dataset Information - Non-VSAM....355
Menu Bar Items....355
Panel Fields - Dataset Information Non-VSAM (Page 1)....356
Panel Fields - Dataset Information Non-VSAM (Page 2)....357
Panel Fields - Dataset Information Non-VSAM (Page 3-n)....357
Dataset Information - VSAM....359
Menu Bar Items....359
Panel Fields - Dataset Information VSAM (Page 1)....359
Panel Fields - Dataset Information VSAM (Page 2)....361
Panel Fields - Dataset Information VSAM (Page 3)....362
Panel Fields - Dataset Information VSAM (Page 4)....362
Panel Fields - Dataset Information VSAM (Page 5-n)....362
DB2 Table Information....364
General Table Information....364
Table Statistics....365
Table Columns....365
Column Attributes....365
Table Indexes....368
Parent Tables....368
Dependent Child Tables....368

Contents

Create Structure (SDO) Menu (=9) **370**
Create Structure Menu Panel 370
Menu Bar Items 370
Options 370
Create Structure from Copybook(s) 371
Create Structure from COBOL/PL1 copybook(s) Panel 371
Options 371
Panel Input Fields 371
Create Structure - Copybook Library List 372
Create Structure - Define Record-Types 372
Panel Input Fields 373
Create Structure from XREF File 375
Create Structure from XREF File Panel 375
Panel Input Fields 375
Display Record Layout 377
Display Record Layout Panel 377
Panel Input Fields 377
Display Record Layout Output 377
Columns Displayed 378
File to Copybook Associations (=9.4) 379
List Loaded Structures (=9.5) 380
List Tables Panel 380
Prefix Line Commands 380
Columns Displayed 380
Generate SELCOPY DCL/EQU Statements (=9.7) 381
Generate DCL/EQU Statements from a Copybook Mapping Panel 381
Panel Input Fields 381

Create File Filter (=10) **382**
Create File Filter Panel 382
Panel Input Fields 382
Unformatted Selection Criteria 384
Unformatted Selection Criteria Panel Table View 384
Unformatted Selection Criteria Panel Single View 384
Panel Input Fields 385
Formatted Selection Criteria 386
Filter (formatted) - INCLUDE/EXCLUDE record-types 386
Filter (formatted) - Selection Criteria 386
Panel Input Fields 388

Print/Report Features Menu (=11) **390**
Options 390
Print Data File (=11.1) 390
Print Data File - Input 390
Print Data File - Input Panel Fields 391
Print Data File - VFMT Options 392
Print Data File - VFMT Options Primary Commands 392
Print Data File - VFMT Options Panel Fields 393
Print Data File - FMT Options 394
Print Data File - FMT Options Primary Commands 395
Print Data File - FMT Options Panel Fields 395
Print Data File - Select Record-Types 396
Print Data File - Select Field Names 397
Print Data File - Field Names List 397
Print Data File - CHAR Options 398
Print Data File - CHAR Options Panel Fields 399
Print Data File - UNFMT Options 399
Print Data File - UNFMT Options Panel Fields 400
Print Data File - Destination 400
Print Data File - Destination Panel Fields 401
Formatted Report Utility (=11.2) 402
Sample Report Definition 402
Sample Report Output 402
Panel Input Fields 403

DB2 Utilities **406**
DB2 Primary Option Menu 406
Menu Bar Items 407
Options 407
Panel Input/Output Fields 407
Execute DB2 Commands 408
Menu Bar Items 408
Field Entries 408
Execute SQL Statements 409
Menu Bar Items 409
Options 409
ExecSQL 409

Contents

DB2 Utilities
Menu Bar Items....409
Panel Input Fields....410
Output Options Panel Fields....411
dSQL....412
Menu Bar Items....413
Field Entries....413
SQL....413
Field Entries....415
Edit Tables and Views....416
Edit DB2 Table Panel....416
Menu Bar Items....416
Panel Fields - Edit DB2 Table....417
Edit DB2 Table Options....418
Panel Fields - Edit DB2 Table Options....418
Edit DB2 Table SQL Clauses....419
Panel Fields - Edit DB2 Table SQL CLauses....420
Primary Commands....420
CMX....420
COLUMNS....420
INDEX....420
JCL....420
OPTIONS....421
SELECT....421
SQL....421
WHERE....421
Browse Tables and Views....422
Browse DB2 Table Panel....422
Menu Bar Items....422
Panel Fields - Browse DB2 Table....422
Browse DB2 Table SQL Clauses....423
Panel Fields - Browse DB2 Table SQL CLauses....423
Primary Commands....424
CMX....424
COLUMNS....424
INDEX....424
JCL....424
SELECT....424
SQL....425
WHERE....425
Create DB2 Objects....426
Create DB2 Objects Panel....426
Menu Bar Items....426
Options....426
Create Storage Group....427
Storage Group Values....427
Generate SQL....428
Create Work File Database....429
Work File Database Values....429
Generate SQL....430
Create User Database....431
User Database Values....431
Generate SQL....432
Create Work File Table Space....433
Work File Table Space Values....433
Generate SQL....434
Create User Table Space....435
Table Space Name & Type....435
Table Space Type Attributes....436
Table Space Partition Attributes....438
Table Space Options (1/2)....440
Table Space Options (2/2)....441
Generate SQL....442
Primary Commands....442
Create Table....443
Table Name & Location....443
Table Definition....445
Columns & Constraints....446
Column Definitions....447
Column Definition - FIELDPROC Parameters....454
Primary Key Definition....454
Unique Constraint Definitions....455
Unique Constraint Key Columns....456
Referential Constraint Definitions....457
Referential Constraint - Parent Key Columns....459
Referential Constraint - Foreign Key Columns....460
Check Constraint Definitions....461

Contents

DB2 Utilities

Model using LIKE....462
Model on Result Table....463
Result Table Column Name Remap....464
Model on Tables/Views (Load Values)....465
Materialized Query Definition....467
Implicit TableSpace Options....468
Table Options (1/2)....469
Range Partitioning Key Columns....471
Tablespace Range Partitions....473
Range Partitioning Key Values....473
Table Options (2/2)....475
Generate SQL....475
Primary Commands....477
Create Index....478
Index & Table Name....478
Index Type....479
Index Key Columns....481
Index Key Columns & Expressions....483
Index Options....485
Index Space....486
Partitioned Index Space....488
Index Partition/Partitioning Attributes....489
Partitioning Key Column Values....491
XML Index....493
XML Index Namespace Prefixes....494
Generate SQL....495
Primary Commands....495
Create View....497
View Definition....497
Result Table Column Name Remap....498
Common Table Expression Definitions....499
Generate SQL....500
Primary Commands....500
Create Alias....502
Alias Values....502
Generate SQL....503
Create Synonym....504
Synonym Values....504
Generate SQL....505
Create Distinct Type....506
Distinct Type Values....506
Generate SQL....507
Create Trigger....508
Trigger Name & Type....508
Trigger Action....509
Trigger Update Columns....511
Triggered SQL Statements....512
Generate SQL....513
Primary Commands....513
Create Sequence....515
Sequence Values....515
Generate SQL....517
Create Role....518
Role Values....518
Generate SQL....518
Create Clone Table....519
Table Names....519
Generate SQL....520
Alter User Table Space....521
Table Space Name....521
Menu Bar Items....521
Table Space Name & Type - Panel Fields....521
Table Space Type Attributes....522
Table Space Type Attributes - Panel Fields....522
Table Space Partition Attributes....523
Table Space Partition Attributes - Panel Fields....524
Table Space Options (1/2)....524
Table Options (1/2) - Panel Fields....524
Table Space Options (2/2)....525
Table Options (2/2) - Panel Fields....525
Generate SQL....526
Primary Commands....526
CMX....526
JCL....526
RUN....526
SELECT....526

Contents

DB2 Utilities
Drop DB2 Objects....527
Drop DB2 Objects Panel....527
Menu Bar Items....527
Options....527
Panel Input/Output Fields....527
Drop DB2 Storage Group....527
Field Entries....527
Drop DB2 Database....528
Field Entries....528
Drop DB2 Table Space....528
Field Entries....528
Drop DB2 Table....528
Field Entries....528
Drop DB2 View....529
Field Entries....529
Drop DB2 Alias....529
Field Entries....529
Drop DB2 Index....529
Field Entries....529
Drop DB2 Synonym....530
Field Entries....530
Drop DB2 Distinct Type....530
Field Entries....530
Drop DB2 Function....530
Field Entries....530
Drop DB2 Stored Procedure....531
Field Entries....531
Drop DB2 Trigger....531
Field Entries....531
Drop DB2 Sequence....531
Field Entries....531
Drop DB2 Role....532
Field Entries....532
Drop DB2 Trusted Context....532
Field Entries....532
Drop DB2 Package....532
Field Entries....532
Drop DB2 Clone Table....533
Field Entries....533
Generate SQL....533
Generate SQL - Panel Fields....533
List DB2 Objects....535
List DB2 Objects Menu Panel....535
Menu Bar Items....535
Options....535
List Storage groups....536
Panel Input Fields....536
Prefix Line Commands....536
List Databases....537
Panel Input Fields....537
Prefix Line Commands....537
List Table spaces....538
Panel Input Fields....538
Prefix Line Commands....538
List Tables....539
Panel Input Fields....539
Prefix Line Commands....539
List Related Tables....540
Panel Fields....540
Prefix Line Commands....541
List Views....542
Panel Input Fields....542
Prefix Line Commands....542
List Aliases....543
Panel Input Fields....543
Prefix Line Commands....543
List Indexes....544
Panel Input Fields....544
Prefix Line Commands....544
List Index Keys....545
Panel Input Fields....545
Prefix Line Commands....545
List Synonyms....546
Panel Input Fields....546
Prefix Line Commands....546
List Distinct Types....547

Contents

DB2 Utilities
Panel Input Fields....547
Prefix Line Commands....547
List Triggers....548
Panel Input Fields....548
Prefix Line Commands....548
List Global Temporary Tables....549
Panel Input Fields....549
Prefix Line Commands....549
List Sequences....550
Panel Input Fields....550
Prefix Line Commands....550
List Roles....551
Panel Input Fields....551
Prefix Line Commands....551
List Trusted Contexts....552
Panel Input Fields....552
Prefix Line Commands....552
List Columns....553
Panel Input Fields....553
Prefix Line Commands....553
List Volumes....554
Panel Input Fields....554
Prefix Line Commands....554
List Table Space Parts....555
Panel Input Fields....555
Prefix Line Commands....555
List Packages....556
Panel Input Fields....556
Prefix Line Commands....556
List Package Dependencies....557
Panel Input Fields....557
Prefix Line Commands....557
List Package Privileges....558
Panel Input Fields....558
Prefix Line Commands....558
Select DB2 Objects....559
Select Storage Group....559
Panel Input Fields....559
Prefix Line Commands....560
Select Database....561
Panel Input Fields....561
Prefix Line Commands....561
Select Tablespace....562
Panel Input Fields....562
Prefix Line Commands....562
Select Table....563
Panel Input Fields....564
Prefix Line Commands....564
Select Alias....565
Panel Input Fields....565
Prefix Line Commands....565
Select Index....566
Panel Input Fields....566
Prefix Line Commands....566
Select Synonym....567
Panel Input Fields....567
Prefix Line Commands....567
Select Distinct Type....568
Panel Input Fields....569
Prefix Line Commands....569
Select Function....570
Panel Input Fields....570
Prefix Line Commands....570
Select Stored Procedure....571
Panel Input Fields....571
Prefix Line Commands....571
Select Trigger....572
Panel Input Fields....572
Prefix Line Commands....572
Select Role....573
Panel Input Fields....573
Prefix Line Commands....573
Select Trusted Context....574
Panel Input Fields....574
Prefix Line Commands....574
Select Package....575

Contents

DB2 Utilities
Panel Input Fields....575
Prefix Line Commands....575
Select Unique Key Constraint....576
Panel Input Fields....576
Prefix Line Commands....577
Select Column....578
Panel Input Fields....578
Prefix Line Commands....578
Select XML NameSpace....579
Panel Input Fields....579
Prefix Line Commands....579
Select Server Location....580
Panel Input Fields....580
Prefix Line Commands....580
Select Catalog Alias....581
Panel Input Fields....581
Prefix Line Commands....581
Select Sequence....582
Panel Input Fields....582
Prefix Line Commands....582
Audit Trail Functions....583
Audit Trail Functions Panel....583
Menu Bar Items....583
Options....583
Audit Log Dataset Options....583
Panel Input Fields....583
Print Audit Report....584
Menu Bar Items....584
Panel Input Fields....585
List Audit Datasets....585
Compare DB2 Tables Panels....586
Compare DB2 Base/Results Tables - New Table details and options....586
Panel Input Fields....586
Compare DB2 Base/Results Tables - Old Table details and options....587
Panel Input Fields....587
Compare DB2 Base/Results Tables - Re-synchronisation options....588
Panel Input Fields....588
Compare DB2 Base/Results Tables - Specify Key Columns....590
Panel Input Fields....591
Compare DB2 Base/Results Tables - Options / Action....591
Panel Input Fields....592
Compare DB2 Base/Results Tables - Ancillary Output Files....592
Panel Input Fields....593
Rename DB2 Objects....595
Rename DB2 Objects Panel....595
Menu Bar Items....595
Options....595
Rename DB2 Table....595
Panel Field Entries....595
Rename DB2 Index....596
Panel Field Entries....596
Generate SQL....596
Panel Field Entries....596
Create/Edit DB2 Structure (SDO)....598
Create/Edit DB2 Structure Panel....598
Menu Bar Items....598
Panel Fields - Create/Edit DB2 Structure....598
Create/Edit DB2 Structure Options....599
Panel Fields - Create/Edit DB2 Structure Options....599
Create/Edit DB2 Structure SQL Clauses....600
Panel Fields - Create/Edit DB2 Structure SQL CLauses....600
Primary Commands....600
CMX....600
COLUMNS....601
INDEX....601
JCL....601
OPTIONS....601
SELECT....601
SQL....601
WHERE....601
DB2 Utilities....602
List DB2 Utilities Menu Panel....602
Menu Bar Items....602
Options....602
DB2 UNLOAD Utility....602
UNLOAD Utility Table Name & Location....602

Contents

DB2 Utilities
UNLOAD Utility Output Datasets....604
UNLOAD Utility Options....605
UNLOAD Utility Floating Point Options....607
UNLOAD Utility Delimited Output Options....608
UNLOAD Utility Primary Commands....609
DB2 LOAD Utility....611
LOAD Utility Table Name & Location....611
LOAD Utility Input Dataset....612
LOAD Utility Options....613
LOAD Utility Floating Point Options....614
LOAD Utility Delimited Output Options....615
LOAD Utility Primary Commands....616
Remap record Layout....618
Create DB2 Table Report....618
Options....618
Setup SELCOPYi DB2 Training Material....619

SMF Features (=13)....620
SMF Features Menu Panel....620
Options....620
SMF Features Further Information....620
SMF Field Mapping....620
HELP Key (F1) gives Field Info....620
Primary/Secondary Segments....620
Layout Source Text....621
Useful Commands....621
VBASE....621
NEXT/PREV....621
ZoomW Key....621
SELECT....622
VIEW....622
PRINT / XMLGEN / CSVGEN / JSONGEN....622
SMF Formatted Browse Utility (=13.1)....622
Panel Input Fields....622
SMF Extract Utility (=13.2)....626
Panel Input Fields....626
SMF Report Utility (=13.2)....630
Sample Report Definition....630
Sample Report Output....631
Panel Input Fields....631
SMF Report Definition Control Statements....636
SMF Report Definition Sections....636

Window List (=W)....640

SELCOPYi Command Reference....641
ABOUT....643
ALIAS....643
AMS....644
AMSDIALOG....644
APE....645
AUDPRINT....645
BOTTOM....646
BROWSE....646
CALENDAR....646
CALC....647
CBLI....647
CBLICANCEL....648
CBLNAME....648
CFOUT....648
CLOSE....648
CMDS....649
COMMANDLINE....649
COMPFILE....650
COMPLIB....664
COMPTABLE....666
CRETRIEV....667
CSVGEN....667
CURSORSELECT....671
DB2....672
DCMD....673
DOWN....673
DRAGBORDERMINUS....674
DRAGBORDERPLUS....675
DSINFORMATION....675
DSQL....676

Contents

SELCOPYi Command Reference

EDIT....676
EO....677
ERASE....678
EXECSQL....680
FAV....683
FCOPY....683
FS....688
FSU....689
FSUEND....703
FSUOUT....704
FSUUNDO....704
GETXML....705
HELP....706
HELPINDEX (HIX)....706
HOME....707
IEBCOPYDIALOG....707
ISPF....707
ISPFUTIL....708
JBOTTOM....708
JLEFT....708
JRIGHT....709
JSONGEN....709
JTOP....713
KEYS....713
KEYS Dialog....714
KEYS Dialog (KEYLISTs ON)....714
KEYS Dialog (KEYLISTs OFF)....715
KEYLIST....716
KEYLIST PUSH, PUSHFKA and POP....716
KEYLIST Dialog....717
Prefix Line Commands....717
LA....718
LAS....718
LC....720
LD....722
LEFT....724
LJQ....725
LL....725
LLS....726
LLX....727
LP....729
LQ....730
LSG....730
LSGV....731
LV....731
LVOL....732
LVR....733
LX....733
MAXIMISE....734
MDINEXT....734
MDIPREV....735
MERGE....735
MINIMISE....736
MOVEWINDOW....736
NEXTMAINWINDOW....738
NEXTWINDOW....738
PFSHOW (FKA)....738
PFSHOWSTYLE (PFS)....739
POWER....739
PREVMAINWINDOW....740
PREVWINDOW....740
PRINT....741
PUTXML....744
QUICKREF....744
QUIT....745
RECOVER....745
RENAME....745
REPORT....747
RESTORE....748
RETRIEVE....749
RIGHT....749
SDATA....750
SDE....751
SDSF....751
SELCOPY....751
SETCOLOUR....754

Contents

SELCOPYi Command Reference
SETFOCUS....755
SHOWPOPUPMENU....755
SHOWWATTR....756
SIZEWINDOW....758
SMFB....759
SMFEXTRC....764
SMFRPT....768
SQL....773
STRUCTURE....774
SVC....774
SYSAPF....775
SYSCOMMAND....775
SYSI....776
SYSLL....776
SYSLPA....776
SYSMENU....776
SYSPGM....777
SYSSTOR....777
SYSTASK....777
TASK....778
TOP....778
TRACE....779
UP....780
VCAT....781
VIEW....781
VOLSTATS....782
WINDOWLIST....783
WINDOWNAMES....783
XMLGEN....784

Unix System Services (USS) Commands....792
USS CHDIR....792
USS GETCWD....792
USS LINK....793
USS MKDIR....793
USS REALPATH....793
USS RENAME....794
USS RMDIR....794
USS STAT....794
USS UNLINK....795

SELCOPYi VTAM commands....796
MESSAGE....796
QUERY....796
STOP....797

SELCOPYi Dump Files....798

Appendix A - SELCOPYi Window Classes....799

Appendix B - List File Prefix Command Summary....800
Command Cross-Reference....802

Glossary....803

Documentation Notes

Fifth Edition, November 2020

Information in this document details general features and functionality of the **SELCOPY Product Suite** component, **SELCOPYi**.

Copyright in the whole and every part of this document and of the SELCOPY Product Suite system and programs, is owned by Compute (Bridgend) Ltd (hereinafter referred to as CBL), whose registered office is located at 8 Merthyr Mawr Road, Bridgend, Wales, UK, CF31 3NH, and who reserve the right to alter, at their convenience, the whole or any part of this document and/or the SELCOPY Product Suite system and programs.

SELCOPY Product Suite for z/OS, z/VM (CMS) and z/VSE operating systems, which includes SELCOPY, SLC, SELCOPYi and CBLVCAT, is available for download and install from **www.cbl.com/selcdl.php**.

The following publications for SELCOPY Product Suite and its component products are available in Adobe Acrobat PDF format at CBL web page **www.cbl.com/documentation.php**:

- SELCOPY Product Suite Customisation Guide
- SELCOPY User Manual
- SELCOPY C++ (SLC) Language Reference
- CBLVCAT User Manual
- SELCOPYi Reference and User Guide
- SELCOPYi Text Editor
- SELCOPYi Data Editor (SDE)
- SELCOPYi Quick Reference
- SELCOPYi REPORT Utility
- SELCOPYi SMF Utilities
- SELCOPYi Training Manual

No reproduction of the whole or any part of the SELCOPY Product Suite system and programs, or of this document, is to be made without prior written authority from Compute (Bridgend) Ltd.

At the time of publication, this document is believed to be correct. Where the program product differs from that stated herein, Compute (Bridgend) Ltd reserve the right to revise either the program or its documentation at their discretion. CBL do not warrant that upward compatibility will be maintained for any use made of this program product to perform any operation in a manner not documented within the user manual.

The following generic terms are used throughout this document to indicate all available versions and releases of IBM mainframe operating systems:

z/OS	-	z/OS, OS/390, MVS/ESA, MVS/XA, MVS/SP, OS.
z/VSE	-	z/VSE, VSE/ESA, VSE/SP, DOS.
z/VM CMS	-	z/VM, VM/ESA, VM/XA, VM/SP.
All	-	All z/OS, z/VSE and z/VM CMS operating systems.

Summary of Changes

First Edition (October 2013)

This section is a summary of significant new features provided in SELCOPYi Release 3.20.

SELCOPYi Training Material & Manual

Primary Options panel updated to include "Training" item (T) to guide new users through initialisation of sample test data, copy books and record selection (filter) data sets for use in conjunction with the "*SELCOPYi Training Manual*" publication.

For details, see:

◊ "*Primary Option Menu (=)*"

ISPF style KeyLists

Introduce definition and management of function keys within SELCOPYi key list tables in a manner which is analogous to ISPF panel key lists. Use of key lists is default and is now the preferred alternative to key assignments based on a SELCOPYi window's window class.

For details, see:

◊ "*Function Key Settings (=0.8)*"
◊ "*Function Keys*"
◊ KEYS
◊ KEYLIST

ISPF SPLIT

SELCOPYi windows within the application ISPF panel now resize based on the depth of the display area available to the panel when it is one of the logical sessions involved in an ISPF SPLIT operation. This change overcomes scrolling issues and potential loss of SELCOPYi command prompt display when ISPF SPLIT has been actioned at a location within the 3270 display area.

FCOPY and FSU Utility Panels

Both the File Search, Update, Copy and Remap (FSU) utility and File Copy (FCOPY) utility have undergone extensive changes to improve usability and functionality in what have become very powerful and potentially complex tools.

For details, see:

◊ "*File Copy (=5)*"
◊ "*File Search/Update/Copy/Remap (=6)*"

Data Set Information

Detailed information on all types of file objects may now be displayed using data set information panels which have different contents depending on the type and oraganisation of file object.

For details, see:

◊ "*Data Set Information*"
◊ "*Dataset Information - Non-VSAM*"
◊ "*Dataset Information - VSAM*"
◊ DSINFORMATION

Regular Expressions

Regular expressions may be used as arguments to the FIND and CHANGE parameters of the FSU primary command and equivalent fields in the File Search/Update/Copy/Remap panel views. These provide string search complex pattern matching.

For details, see:

◊ "*Basic File Search Panel*"
◊ "*Extended File Search/Update/Copy/Remap Panels*"

XML Generation

Introduce support for primary command XMLGEN to generate XML source for formatted data.

For details, see:

◊ XMLGEN

DB2 Object Creation

Introduce new panels for DB2 Tablespace and Table creation. Object modelling has also been introduced for existing DB2 object creation panels.

For details, see:

◊ *"Create Work File Table Space"*
◊ *"Create User Table Space"*
◊ *"Create Table"*

DB2 SQL Execution

New DB2 SQL execution facility (EXECSQL) introduced. Like SPUFI, primary command EXECSQL will execute SQL statements in a text file and display the output in a text edit view. However, EXECSQL is a lot more flexible than SPUFI supporting a number of additional options.

For details, see:

◊ EXECSQL

GDG Base Definition

Support panel to define a new GDG Base cayalog entry.

For details, see:

◊ "*Define GDG Base (=8.15.7)*"

Settings Panels

Settings panels have been updated to include text edit ACTION facility options and Function Key (KeyList) management.

For details, see:

◊ "*Text Edit Settings (=0.3)*"
◊ "*Function Key Settings (=0.8)*"

VTOC Extents List

The List VTOC extents panel and primary command (LX) have been enhanced to include a DSN filter.

For details, see:

◊ "*List VTOC Extents (=3.3)*"
◊ LX

Print File Data Utility

Introduce new panels for Printing formatted or unformatted file data. Includes ammendements to the Primary Options Menu and Utilities panels.

For details, see:

◊ "Print File Data Utility"

Text Edit Panel

Text Edit panel (=1) reworked. Original version moved to List panel option 4 (=3.4) as the Data Set List utility.

For details, see:

◊ "*Text Edit (=1)*"
◊ "*Dataset List Utility (=3.4)*"

Structured Data Edit Panel

Data Edit panel (=2) reworked to provide easier implementation of the different edit techniques and record selection (filtering) functionality.

For details, see:

◊ "*Data Edit (=2)*"

Second Edition (March 2015)

This section is a summary of significant new features provided in SELCOPYi Release 3.30.

Library Member MOVE/COPY

Introduce new panels for multiple member MOVE (M or //M...//) and COPY (C or //C...//) in Library Member List window.

For details, see:

◊ "*Library Member Move/Copy Panel*"

XML Generation Panel

Support XML generation panel opened from SELCOPYi Utilities panel.

For details, see:

◊ "*Generate XML Panel*"
◊ XMLGEN

CSV Generation Panel

Support CSV (Comma Separated Variable) generation for structured data (Copybook, DB2 table, etc.)

For details, see:

◊ "*Generate CSV Panel*"
◊ CSVGEN

DSINFORMATION for DB2 Tables

Data set information report output (DSI) support for DB2 tables.

For details, see:

◊ DSINFORMATION primary command

EXECSQL Panel

Introduce EXECSQL panel for DB2 SQL excution.

For details, see:

◊ ExecSQL

DB2 List Related Tables

In conjunction with enhanced support for DB2 table referential constraint obeyance, DB2 related table list is introduced.

For details, see:

◊ "*List Related Tables*"

DB2 Create Objects

Support introduced for DB2 Index and DB2 Clone Table creation.

For details, see:

◊ "*Create Index*"

SELCOPY Debug JCL Input

Support JCL job stream input to the SELCOPY Debug utility with selective job step forground execution and SELCOPY job step debug.

For details, see:

◊ "*Supply JCL*"

Third Edition (December 2016)

This section is a summary of significant new features provided in SELCOPYi Release 3.40.

SELCOPY Debug

Expand functionality of the SELCOPY Debug Utility to support debug of SELCOPY control statements using the SELCOPY C++ program (SLC). Also includes support for the Operations List and Watch List application windows.

For details, see:

◊ "*SELCOPY Debug & Development*"
◊ "*Watch List Window*"
◊ "*Operations List*"

JSON Generation

Support JSON (JavaScript Object Notation) generation for structured data (Copybook, DB2 table, etc.)

For details, see:

◊ "*Generate JSON Panel*"
◊ JSONGEN primary command

Concatenated Library Directories

COMPLIB, FCOPY and FSU utilities support processing of first matching member name found in a DDname library concatenation.

For details, see:

◊ "*File Copy Panel*"
◊ FCOPY primary command
◊ "*Basic File Search Panel*"
◊ FSU primary command
◊ "*Compare Libraries Panel*"
◊ COMPLIB primary command

Print FSU/COMPFILE Output

Option to automatically execute the PRINT utility against report output generated by the File Search and Update (FSU) and Compare Files (COMPFILE) utilities.

For details, see:

◊ FSU primary command
◊ COMPFILE primary command

Compare Libraries

Support execution in a batch environment (SDEAMAIN) and support uncataloged library datasets via a volume id specification.

For details, see:

◊ "*Compare Libraries Panel*"

DB2 XML Column Data

GETXML and PUTXML utility commands supported to import and export XML documents between datasets and DB2 XML column entries.

For details, see:

◊ GETXML primary command
◊ PUTXML primary command

DB2 LOAD/UNLOAD Utility Panels

DB2 panel interfaces to generate DB2 LOAD and UNLOAD batch jobs for DB2 table data.

For details, see:

◊ "*DB2 Unload*"
◊ "*DB2 Load*"

DB2 Object Rename and Alter Panels

DB2 panel interfaces to execute SQL RENAME on a DB2 table or index, and SQL ALTER on a DB2 tablespace.

For details, see:

◊ "*Rename DB2 Objects*"
◊ "*Alter User Table Space*"

DB2 Create Structure Panels

DB2 panel interface to create a SELCOPYi SDO structure from an SQL Query.

For details, see:

◊ "*Create/Edit a DB2 Structure*"

Diagnostic Trace

Generate diagnostic trace output for CBL SELCOPYi support desk.

For details, see:

◊ TRACE primary command

Window Position Justify

Justify the focus window's position relative to the parent window's display area limits.

For details, see:

◊ JBOTTOM primary command
◊ JLEFT primary command
◊ JRIGHT primary command
◊ JTOP primary command

Fourth Edition (October 2018)

This section is a summary of significant new features first introduced in SYSMOD RS34003 for SELCOPYi Release 3.40.

SMF Record Browse and Reporting

New panels for SMF record formatted display and report generation. The feature introduces new structured data report writing control syntax that may be used to generate basic reports from SMF source records or indeed any formatted data.

For details, see:

◊ "*SMF Utility Panels (=13)*"
◊ "*SMF Report Definition Control Statements*"

PDSE V2 Member Generations

Support Edit/Browse/Recover/Search/Compare of PDSE Version 2 library member generations. (PDSE V2 defined with MAXGENS.)

For details, see:

◊ "*z/OS PDSE Library Member Generations*"
◊ "*Generic z/OS PDSE Library Member Generations*"

Fifth Edition (July 2020)

This section is a summary of significant new features provided in SELCOPYi Release 3.50.

Report Utility

The SMF Report Utility tool (SMFRPT) has been further developed to become a more general purpose report generation tool (REPORT). The REPORT utility is capable of processing not only SMF records with pre-defined layouts, but also data records formatted by a supplied structure (e.g. a COBOL copybook) as well as DB2 table rows.

New REPORT primary command and utility panels introduced.

For details, see:

◊ "*SELCOPYi REPORT Utility Manual*"
◊ "*Create Formatted Record Report (=11.2)*"
◊ "*Create DB2 Table Report (=12.13 or =11.3)*"
◊ REPORT primary command

Utility Field to Field Mapping

File Copy, File Search & Update and File Compare utilities each support processing records in 2 data sets where data from each data sets is formatted using different structures. This new feature allows a map between a field from one structure with a field from the other despite potentially having different field names, data types and record layout names.

For details, see:

◊ "*File Copy Panel*"
◊ "*Extended File Search/Update/Copy/Remap Panels*"
◊ "*Compare File - Formatted Compare Panels*"
◊ COMPFILE primary command
◊ FCOPY primary command
◊ FSU primary command

File Search Report Context Records

Like the Compare File utility, the File Search & Update utility now supports display of "context" records in its formatted report output. The report will display a number of records that occur immediately before and after each record that contains a match for the search string.

For details, see:

◊ "*Basic File Search Panel*"
◊ FSU primary command

File Search & Update I/O Records

For multiple input and output files, the File Search & Update utility is now able to restrict the number of records read in total from **all** input files and the number of records written in total to **all** output library members.

For details, see:

◊ "*Basic File Search Panel*"
◊ FSU primary command

File Compare Record Masking and Blank Suppression

For unformatted record compare, the Compare Files utility now supports conditional masking of areas within the input records. A mismatch that would otherwise be reported in these areas will not be detected. The utility also supports a parameter to suppress blanks in the compare operation so that a record pair matches if the only difference is the number of blanks between words in the text.

For details, see:

◊ "*Compare File - Extended Unformatted Compare Panels*"
◊ COMPFILE primary command

XML and JSON Generation

The XML and JSON generation utilities have been updated so that the character case of alpha characters in markup tags may match that of the field names from which they are generated. Previously, tag names were always upper cased.

For details, see:

◊ "*XMLGEN (=8.16)*"
◊ "*JSONGEN (=8.18)*"
◊ JSONGEN primary command
◊ XMLGEN primary command
◊ NAMECASE option in the Data Editor (SDE) manual.

SELCOPY Debug

To expand support of SELCOPY debug for a SELCOPY exceution running in an IMS DLI region, the options -DBRC and -IRLM have been added to the SELCOPY primary command syntax.

For details, see:

◊ SELCOPY primary command

List Window Enhancements

The List class of windows supports a number of additional commands relating to exclusion of list entry lines. Additionally, Catalog entry and Dataset lists include a new column "VolX", which displays volume or entry type values, and a VSAM Data and Index selection field.

For details, see:

◊ "*List Entry Display*"
◊ EXCLUDE Command
◊ FLIP Command
◊ HIDE Command
◊ MEMBER Command
◊ ONLY Command
◊ RESET Command
◊ SHADOW Command
◊ SRCHFOR Command
◊ "*List Catalog Files (=3.5)*"
◊ "*List Dataset Details (=3.6)*"

About SELCOPYi

System Environment

SELCOPYi is a full screen 3270 application which executes in any of the following environments:

◊ z/OS TSO/E.
◊ z/OS TSO/E as an ISPF application.
◊ z/OS as a stand-alone, multi-user VTAM application. (See Disaster Recovery below)
◊ z/VSE as a stand-alone, multi-user VTAM application.
◊ z/VM CMS.

General Functionality

The SELCOPYi environment includes a compendium of tools and facilities that operate within a windowed environment within the 3270 display. The following is a selection of the supported features:

◊ Function rich text editor (CBLe) with both ISPF Edit and XEDIT compatibility.
◊ Structured data editor supporting COBOL and PL1 Copybooks.
◊ DB2 table editor.
◊ DB2 SQL execution.
◊ List facility includes DB2 objects, Datasets, DASD Volumes, VTOC files/extents, HFS files and ENQs.
◊ File Search and Update including support for copybook map.
◊ File Copy supporting mixed data set organisations and copybook remap.
◊ File and PDS/PDSE library Compare.
◊ SELCOPY interactive development environment and debugger.
◊ CBLVCAT interactive reports and VSAM data set tuning.

SELCOPYi Environment

SELCOPYi is a full screen 3270 interface which provides the user with a working environment whereby all tools and facilities (dialog panels, edit views, data set lists, etc.) are presented in windows within the 3270 display. Like PC workstation and UNIX based operating systems that support a windowed environment, SELCOPYi includes the following features:

◊ Multiple overlapping window views.
◊ Window title and menu bars.
◊ Minimise, Maximise and Resize of displayed windows.
◊ Repositioning of displayed windows.
◊ Point-and-shoot buttons.
◊ Drop down and popup menus.

Disaster Recovery

SELCOPYi has the same functionality when executing as a VTAM application as it does when running in user's TSO/E address space. Therefore, customising SELCOPYi to execute as a stand-alone VTAM application from a recovery volume, would allow functions such as data editing, job submission, data set allocation and system navigation, even if ISPF and TSO/E are unavailable.

In addition to functionality included in the SELCOPY and CBLVCAT batch utilities, the systems programmer is provided with a powerful set of tools to assist in the data recovery process.

Getting started with SELCOPYi

This chapter introduces end users to some basic SELCOPYi concepts.

Starting the SELCOPYi program

How SELCOPYi is started depends on the environment in which it is to be executed.

OS	Environment	Command
MVS	TSO/E	Enter **SELCOPYI** at the READY prompt.
	ISPF	Enter **TSO %SELCOPYI** on an ISPF command line. Alternatively, SELCOPYi may have been included as a selectable item in an ISPF menu and/or as an ISPF command (e.g. SI) by your systems programmer as part of the product install. ISPF screen management is used and so SELCOPYi must have first been defined as an ISPF application. See *SELCOPY Products Suite Customisation Guide* for further information on configuring access to SELCOPYi in ISPF.
	VTAM	Enter **LOGON APPLID(SELCOPYI)** on a VTAM USS screen. **Note:** The SELCOPYi VTAM session controller program (CBLIVTAM) must be running and the SELCOPYi VTAM applid, SELCOPYI, must be active. See *SELCOPY Products Suite Customisation Guide* for further information on configuring access to SELCOPYi as a stand-alone VTAM application.
	Batch	Execute program SDEAMAIN with primary command syntax supplied in DD SDEIN and list output to DD SDEPRINT.
VSE	VTAM	Enter **LOGON APPLID(CBLIVTAM)** on a VTAM USS screen. Alternatively, SELCOPYi may have been included as a selectable item in the VTAM Application Selection Menu, VTMUSSTR (SNA) or VTMUSSTB (non-SNA), by your systems programmer as part of the product install. **Note:** The SELCOPYi VTAM session controller program (CBLIVTAM) must be running in a static or dynamic partition and the SELCOPYi VTAM applid, CBLIVTAM, must be active. See *CBL Software Install Guide for VSE Systems* for further information.
VM	CMS	Enter **SELCOPYI** on a CMS command line to execute the SELCOPYi startup Rexx EXEC.

Security Considerations

VSE Systems

By default, it is assumed that a Basic or Extended Security Manager (BSM or ESM) is operational (SEC=YES) and so SELCOPYi authenticates the user logon id and password at startup, and thereafter performs resource access checking for the userid as required. e.g. For LIBR library lists and member edit.

Before an attempt is made to perform an operation on a resource, SELCOPYi first checks whether the user has sufficient access authority for that resource and, if not, does not attempt the operation but instead returns an error message to the user's session.

When running with a security manager, SELCOPYi processing of POWER queue list entries, displayed in the Execute POWER command window, is allowed only if the user's logon id matches either the TO or FROM field values.

If no security manager is in effect, then the SELCOPYi System INI variable SYSTEM.VSESMLogon=No must be set in order to bypass userid authentication and resource access checking. The user is prompted for a logon id, which gets assigned to the environment variable %user%, but no password is required.

With no security manager, access to resources (LIBR libraries, members, etc.) will be unrestricted with the exception of POWER queue list entries where the user may process only queue entries that are password protected and for which the password is known to the user.

Whether or not a security manager is operational, SELCOPYi may be customised to restrict its use to only a specified group of trusted users. Following logon, these users will be prompted for a SELCOPYi password which may differ from the user's security manager password.

If the trusted user facility is activated, then no other users will be able to successfully start SELCOPYi. Trusted users may process any POWER queue list entry without restriction.

Trusted users may process any POWER queue list entry without restriction.

See section *Security Manager* in document *CBL Software Install Guide for VSE Systems* for further information.

MVS Systems

On MVS systems, users login to SELCOPYi using their RACF, or equivalent security package, login id. Under TSO, no SELCOPYi login is performed as the user's TSO login id is used instead.

Thereafter, the user's access privilage is verified prior to performing any action on potentially protected resources. e.g. listing PDS(E) library contents and editing data sets.

In addition to this, the security administrator can restrict users' access to the following SELCOPYi features using RACF profiles.

Resource Name	SELCOPYi Feature
System	Access to the z/OS Operating System information available by selecting 'Operating System' from the Utilities/System menu in the CBLe main window menu bar, or via line commands `SYSI`, `SYSLPA`, `SYSLL`, `SYSAPF`, `SYSTASK`, `SYSSTOR`, `SYSPGM` and `CBLNAME`.
UserTSO	Log on to SELCOPYi under TSO and ISPF.
UserVTAM	Log on to SELCOPYi as VTAM application.
SELCOPY	Use of the SELCOPY Debug application.
CBLVCAT	Use of the CBLVCAT Interactive (VCI) application.
DB2	Use of DB2 features.

If these resources have been customised then users must have at least READ access to the specific resource in order to be able to perform the equivalent operation.

See section *Security Considerations* in document *SELCOPY Product Suite Customisation Guide* for further information.

3270 Terminal Emulation

Before SELCOPYi is started, 3270 terminal session is required. Most installations now use a 3270 emulation software executing on a workstation rather than a real 3270 terminal.

3270 Screen Sizes

The CBL 3270 screen manager can operate in any 3270 screen size up to a total area of 16384 (16K) with a maximum width or depth of 255. This 16K area limit is imposed by the 14 bit address format of the 3270 data stream used by SELCOPY/i. The 255 width or depth limit is the result of some components using just 1 byte to store these dimensions.

All 3270 emulation software packages allow the user to configure a 3270 session to emulate hardware terminal models 2/3/4/5 having rows x columns screen sizes of 24x80, 32x80, 43x80 and 27x132 respectively.
Most good 3270 emulators also support the ability to define non-standard terminal sizes (dynamic TN3270 terminals) that allow users to obtain 3270 screen sizes with dimensions much larger than the standard hardware models.

Note: To configure z/OS non-standard screen sizes, a VTAM dynamic logmode must be defined to your system (IBM supply VTAM dynamic logmode D4C32XX3 in VTAMLIB.) This logmode may optionally be configured as the default for TN3270 sessions via a TELNETDEVICE DYNAMIC entry in the TN3270 server configuration data set.
See the IBM Technotes and Tips web page entitled *Creating dynamic 3270 screen size definitions for increased productivity* for further information.

When using 3270 emulation software, it is strongly recommended that the largest possible screen size be used to take full advantage of SELCOPY/i's ablility to display multiple overlapping windows. At CBL, a screen size of **96x160** is regularly used for TSO sessions (160 being the maximum number of columns supported by ISPF) and **86x190** used for CMS and VTAM sessions (MVS and VSE).

A selection of popular 3270 emulator packages have been installed and tested at CBL to determine support for dynamic terminal sizes and other features considered useful to SELCOPY/i operation (Keyboard macros, etc.) A synopsis of results for each product tested to date may be found at the *SELCOPYi and 3270 Terminal Emulation Software* web page.

Sample 3270 emulator configuration files that provide non-standard terminal sizes have been generated for selected 3270 emulation products may also be downloaded from this location.

Keyboard and Mouse Mapping

Configuration of the 3270 emulator software keyboard and mouse is recommended to greatly enhance the user's experience of the SELCOPYi window environment. Most good 3270 emulator software packages provide the facility to do this.

Traditionally, 3270 terminal keyboards provide only the 24 programmable function keys (PF01-PF24) whereas 3270 emulation software provides the facility to map functions and macros to a much wider range of key combinations. (e.g. Ctrl-S may be mapped to execute a SAVE operation.)

Of most benefit to the use of SELCOPYi, is the ability to map the mouse left button double-click action to be "position the cursor here and press <Enter>". Configuring this emulator feature provides quicker window manipulation, button pressing, menu selection, etc., so allowing the user to operate on SELCOPYi windows using the mouse in a manner that is intuitive to users of PC or UNIX workstations.

In addition to sample 3270 session configuration files, CBL provides a library of keyboard macros with recommended keyboard and mouse map files for 3270 emulator products, IBM Personal Communications and Tom Brennan's Vista3270.

For further information, see the CBL web pages *SELCOPYi and 3270 Terminal Emulation Software* and *SELCOPYi Downloads*.

Window Concepts

The SELCOPYi window environment is managed by the CBL3270 screen manager.

In general, CBL3270 managed windows behave in a similar fashion to window GUI environments provided by PC and UNIX operating systems.

This section provides technical detail on characteristics and concepts shared by all SELCOPYi windows. It also provides instructions on how to work with windows in the SELCOPYi environment.

Window Hierarchy

All windows exist in a hierarchy. At the top of the hierarchy is the desktop window which is automatically created during initialisation. The desktop window occupies the entire screen and cannot be moved, resized or destroyed. All other windows, including the SELCOPYi main window, are dependents of the desktop.

When an application creates a window the new window has to be dependent on an existing window, the parent or owning window. If the application does not supply an existing window then the desktop window is used by default. This dependency relationship has two forms:

- Owned window. The dependent window is owned by the existing window which is called its owner. The owned window can be moved all over the display surface but is always in front of (cannot be obscured by) its owner. Typically owned windows are used for more complex entities such as MDI frame windows, dialog boxes and help windows.
- Child window. The dependent window is not only owned by, but is also a child of the existing window which is called its parent. The child window can only exist within the rectangle defined by its parent's client area. Typically, child windows are used for low level entities such as buttons and input fields.
- MDI (Multiple Document Interface) Child window. The dependent window is an MDI child of the existing MDI parent (frame) window. Like child windows, the MDI child window can only exist within the rectangle defined by its parent's MDI client area. However, unlike child windows, each MDI child window has a sizing border, a title bar, a window menu, a minimise button, and a maximise button.

  SELCOPYi MDI applications include the CBLe text editor and SELCOPY Debugger, each supporting various types of MDI child windows. e.g. text edit views, SDE edit views, list windows, help windows and IPO panels.

When a window is destroyed, so are all of its dependent owned and child windows.

Manipulating Windows

Moving a window

If a window has a title bar it can be moved with the following procedure:

1. Place the cursor in the title bar of the window.
2. Press the <Enter> key. The window border will be highlighted.
3. Move the cursor to a new position.
4. Press the <Enter> key. The window will move by an amount equal to the displacement of the cursor.

Note that, if configured, double-clicking the left mouse button on the window title bar, then doing the same at the new location will perform the same action. Also see line command MOVEWINDOW and SET WINPOS for Text Editor document window views.

Resizing a window

If a window has a border it can be resized with the following procedure:

1. Place the cursor in the border of the window. If the cursor is in the top or bottom border, the window will have its depth changed. If the cursor is in the left or right border it will have its width changed. If the cursor is in a corner of the border it will have its width and depth changed.
2. Press the <Enter> key. The window border will be highlighted.
3. Move the cursor to a new position.
4. Press the <Enter> key. The window will be resized by an amount equal to the displacement of the cursor.

Note that, if configured, double-clicking the left mouse button on a window border, then doing the same at the new location will perform the same action.

Also see line commands SIZEWINDOW, DRAGBORDERMINUS, DRAGBORDERPLUS and, for CBLe text edit document windows only, SET WINSIZE.

Maximising a window

If a window has a maximise button it can be maximised by moving the cursor to the maximise button and hitting the <Enter> key or, if configured, double-clicking the left mouse button on the maximise button. The window will then take up the whole of the 3270 screen. The maximise button will change from a plus sign to a solid vertical bar (representing restore).

Also see line command MAXIMISE and, for CBLe text edit document windows only, WINDOW MAX.

Minimising a window

If a window has a minimise button it can be minimised by moving the cursor to the minimise button and hitting the <Enter> key or, if configured, double-clicking the left mouse button on the minimise button. The window will then be removed from the display and replaced by a small iconic window showing just a portion of its title bar near the botton of the SELCOPYi main window. The minimise button will change from a minus sign to a solid vertical bar (representing restore).

Also see line command MINIMISE, for CBLe text edit document windows only, WINDOW MIN.

Restoring a window

When a window has been maximised or minimised, the maximise/minimise button is replaced restore button.

A maximised or minimised window may be restored to its former location and size by moving the cursor to the restore button and hitting the <Enter> key or, if configured, double-clicking the left mouse button on the restore button.

Also see line command RESTORE, for CBLe text edit document windows only, WINDOW REST.

Closing a window

A window can be closed by moving the cursor to the close button and hitting the <Enter> key or, if configured, double-clicking the left mouse button on the close button.

Also see line command CLOSE and, for CBLe text edit document windows only, WINDOW CLOSE.

Window Format

SELCOPYi windows have a standard format which consists of the following components. Not all windows have all these components, but where present they occupy the same relative position in the window and have the same function.

Title Bar

The title bar contains the title of the window.

System Menu Button

The system menu button is at the left end of the title bar. If pressed the options of the system menu are displayed in a popup menu.

Minimise Button

The minimise button is at the right end of the title bar. It is represented as a single "-" (minus sign) in white reverse video. If the window is minimised then the minus sign is replaced by a solid vertical bar representing restore.
Note that the Minimise button is not present in the SELCOPYi main window.

Restore Button

The Restore button is displayed in place of the Minimise or Maximise button when a window is in a minimised or maximised state respectively. It is represented by a solid vertical bar in white reverse video.
Note that the Restore button is not present in the SELCOPYi main window.

Maximise Button

The maximise button is at the right end of the title bar. It is represented as a single "+" (plus sign) in white reverse video. If the window is maximised then the plus sign is replaced by a solid vertical bar representing restore.
Note that the Maximise button is not present in the SELCOPYi main window.

Close Button

The close button is at the right end of the title bar. It is represented as an "x" in red reverse video.

Menu Bar

The menu bar occupies one or more lines below the title bar. It contains optional items that can be selected by positioning the cursor on the menu item text and hitting <Enter> or, if configured, double-clicking the mouse left button on the menu item.

Menu items may display pop-up sub-menu items which may be selected in the same fashion. All menu items have the 3270 unprotected attribute so they can be navigated using the <Tab> key which tabs to enterable fields. Any data entered into a menu item is ignored.

Menu items may have been enabled or disabled by the SELCOPYi application. When enabled they are displayed in white, when disabled they are displayed in blue in which case their selection will have no effect.

Client Area

The client area of a window is the main body of the window and its contents vary depending on the class of the window.

Command Line

The command line is an area of the window into which text commands may be entered. Most menu items have a command line equivalent text command.

Pressing Buttons

The 3270 architecture is such that the host is only informed of user input when a certain class of key is pressed on the keyboard. These keys are those with attention identifiers and typically consist of the function keys, the enter key, the Program Attention (PA) keys and a few other specialised keys. Even when operating under the control of a workstation 3270 emulator, the 3270 host application is not sensitive to mouse movements (except in the case when the emulator allows the user to assign a particular 3270 function to the mouse buttons).

Because of this, 3270 window buttons cannot be pressed in the same way as workstation window buttons. 3270 window buttons are pressed by positioning the cursor on the button and hitting <Enter> or, if configured, double-clicking the left mouse button on the button icon.

Window Focus

The focus window is the window which contains the cursor when the 3270 screen is displayed. If the window has a title bar then the fact that it is the focus window is indicated by colouring the title bar area with blue reverse video. All other windows have a white reverse video caption bar.

Input fields

The focus window defines the input rectangle which is the only area of the screen where input is enabled. When the focus window is not a child window the input rectangle is the window itself.

When the focus window is a child window the input rectangle is that defined by its parent. Any input fields outside of the input rectangle are temporarily disabled. Each field in the input rectangle can be visited by using the cursor tab key (shift+cursor tab key to reverse the direction).

Changing the focus window

The user can change the focus window using the following methods:

- By placing the cursor in a window and hitting <Enter> or, if configured, double-clicking the left mouse button in the window display area. This sets the focus to this window.
- WINDOW + (+ is default) or WINDOW - line commands. These commands cycle through all windows in creation sequence. By default, WINDOW is assigned to function key F4.
- By selecting 'All Windows', from the Window menu in the CBLe main window menu bar, or executing the WINDOWLIST command to open the Window List window and then selecting the required window entry. The selected window becomes the focus window.
- By using the SETFOCUS line command to explicitly name the focus window.

Window Names

All the windows defined by a CBL3270 application have a name. The name is supplied by the application when the window is created and may be changed later during the window's life.

If the name is not supplied by the application then a default name is supplied by CBL3270 made up of the window class name suffixed with a three digit number which is incremented by 1 for each window of the class created during the CBL3270 session.

The main use of window names is to allow commands entered on the command line of a window to refer to other windows which are currently part of the application.

Viewing Window Names

The window name associated with each window in the SELCOPYi session may be displayed in that window's title bar (and subsequently hidden from view) via the following:

- Select 'Display/Hide Window Names' from the system menu belonging to any open window.
- Enter the line command WINDOWNAMES on the command line of any window.

Either of these operations will display or hide the window names for **all** open windows in the SELCOPYi session.

By default, display of window names is suppressed to avoid overcrowding the title bar.

Alternatively, the Window List window may be used to obtain a window's associated window name. The window list displays all open windows and their associated window names (including MDI window names.)

Window Class

All the windows defined by a CBL3270 application are members of a window class.

A window class is identified by a 1 to 8 character name and defines window behaviour, appearance and functions that are commmon to windows belonging to that window class.

CBL3270 uses the window class name to associate a processing module (called a Window Procedure) with a window.

All windows in the same class are managed by the same window procedure. The window procedure is called by CBL3270 whenever an event occurs which affects the window. It is the window procedure's responsibility to:

1. Paint the window client area.
2. Establish PFKey definitions for the individual Class Function Keys table.
3. Process any data entered into the window.
4. Respond to any commands issued from the window's menu or command line.

A complete list of SELCOPYi Windows Classes may be found in *Appendix A - SELCOPYi Window Classes*.

System Menu

The system menu is a menu of functions which is available on all windows within a CBL3270 environment.

You access the system menu by pressing the system menu button at the top left of the main window (or any subordinate window that has a system menu button). The following options are available from the system menu:

Layout	For Storage Display Windows only, displays the options popup menu. (As for SHOWPOPUPMENU.)
Restore	Restore a maximised or miminised window to its original size and position.
Minimise	Minimise the window with the system menu.
Maximise	Maximise the window with the system menu.
Close	Close the window with the system menu.
Quit	Quit the application.
Window List	Open a window showing the current list of windows. You can select a window from this list by placing the cursor on a list element and hitting <Enter> or, if configured, double-clicking the left mouse button. The selected window will be given the focus.
Next window	Give the focus to the next window in the hierarchy of open windows.
Previous window	Give the focus to the previous window in the hierarchy of open windows.
Command line	Open the command line dialog.
Function keys	Open the functions keys dialog.
Show/Hide window names	Toggle the status of the window names display. Window names are unique window identifiers which can be used in commands to identify a particular window. If displayed, the window name is displayed in the title bar as a prefix to the window caption.
Use ISPF/TSO	When running under TSO/ISPF, this menu command toggles the display from ISPF to TSO format.

Function Keys

For SELCOPYi Rel 3.20 onward, on z/OS systems SELCOPYi maintains its function key definitions in **KEYLIST**s, analogous to ISPF keylists.

Use of KeyLists may be controlled using the Function Keys Settings (=0.8) panel, or the KEYLIST primary command.

To view and update PFKey settings in any application window, enter primary command KEYS. The format of the KEYS dialog display is dependent on whether KeyLists are active.

Default Function Keys (KEYLISTs active)

The SELCOPYi supplied default **Caption** (title bar) function keys, defined by **KEYLIST=@WINTITL** are:

F7	MoveWindow by y=-1	Drag window up 1 line
F5	JTopBottom	Justify the window location to the top of screen, unless already at the top, in which case justify to the bottom.
F6	JLeftRight	Justify the window location to the left of screen, unless already at the left, in which case justify to the right.
F8	MoveWindow by y=+1	Drag window down 1 line.
F10	MoveWindow by x=-1	Drag window left 1 column.
F11	MoveWindow by x=+1	Drag window right 1 column.
F13	WinX S	Save current browse/edit window size/position.
F14	WinX R	Restore current browse/edit window size/position.
F15	SizeWindow by w=-20	Decrease window width by 20 columns.
F16	SizeWindow by w=+20	Increase window width by 20 columns.
F17	SizeWindow by d=-20	Decrease window depth by 20 rows.
F18	SizeWindow by d=+20	Increase window depth by 20 rows.
F19	MoveWindow by y=-5	Drag window up 5 lines
F20	MoveWindow by y=+5	Drag window down 5 lines.
F22	MoveWindow by x=-5	Drag window left 5 colums.
F23	MoveWindow by x=+5	Drag window right 5 columns.
F24	MaxRes	Toggle between maximied and non-maximised window state.

The SELCOPYi supplied default **Borders** function keys, defined by **KEYLIST=@WINBORD** are:

F7	DragBorderMinus	Drag top/bottom border up 1 line.

F5	JTopBottom	Justify the window location to the top of screen, unless already at the top, in which case justify to the bottom.
F6	JLeftRight	Justify the window location to the left of screen, unless already at the left, in which case justify to the right.
F8	DragBorderPlus	Drag top/bottom border down 1 line.
F10	DragBorderMinus	Drag left/right border left 1 column.
F11	DragBorderPlus	Drag left/right border right 1 column.
F13	WinX S	Save current browse/edit window size/position.
F14	WinX R	Restore current browse/edit window size/position.
F15	SizeWindow by w=-20	Decrease window width by 20 columns.
F16	SizeWindow by w=+20	Increase window width by 20 columns.
F17	SizeWindow by d=-20	Decrease window depth by 20 rows.
F18	SizeWindow by d=+20	Increase window depth by 20 rows.
F19	DragBorderMinus 5	Drag top/bottom border up 5 lines.
F20	DragBorderPlus 5	Drag top/bottom border down 5 lines.
F22	DragBorderMinus 5	Drag left/right border left 5 columns.
F23	DragBorderPlus 5	Drag left/right border right 5 columns.
F24	MaxRes	Toggle between maximied and non-maximised window state.

Default Function Keys (KEYLISTs not active)

For non-z/OS systems (or with KEYLIST OFF in effect), SELCOPYi maintains function key tables at 5 levels (i.e. WINDOW, CLASS, DEFAULT, TITLEBARS and BORDERS).

WINDOW function keys are unassigned by default. WINDOW key settings apply to an instance of a particular window, therefore user updates via the KEYS dialog are not be remembered by the system.

CLASS function keys are assigned defaults according to their class, with user updates via the KEYS dialog being remembered by the system.

DEFAULT function keys are assigned to the following (unless overridden by the SELCOPYi System/User INI file):

F1	Top	Display data at the top of a scrollable window.
F2	Bottom	Display data at the bottom of a scrollable window.
F3	Close	Close the window in which the cursor is positioned. (N.B. not necessarily the current focus window)
F4	Action	Issue command at the cursor position if prefixed by "<" (less than), otherwise place text on the command line.
F7	Up	Scroll up the file so that the current focus line becomes the last line of the display.
F8	Down	Scroll down the file so that the current focus line becomes the first line of the display.
F9	NextMainWindow	Place focus on the next main window in the SELCOPYi window list.
F10	Left	Scroll the display to the left so that the current focus column becomes the last column of the display.
F11	Right	Scroll the display to the right so that the current focus column becomes the first column of the display.
F12	Retrieve -	Retrieve the last command issued and place it at the command line.
F13	ShowPopupMenu	For storage windows only, display the options popup menu.
F16	Action Edit	Issue command at the cursor position if **not** prefixed by "<" (less than), otherwise place text on the command line.
F21	PrevMainWindow	Place focus on the previous main window in the SELCOPYi window list.
F24	Retrieve +	Retrieve the next command issued and place it at the command line.

The SELCOPYi **Caption** (title bar) function keys are assigned to the following:

F7	MoveWindow by y=-1	Drag window up 1 line
F8	MoveWindow by y=+1	Drag window down 1 line.
F10	MoveWindow by x=-1	Drag window left 1 column.
F11	MoveWindow by x=+1	Drag window right 1 column.

The SELCOPYi **Borders** function keys are assigned to the following:

F7	DragBorderMinus	

		Move the border on which the cursor is position 1 column and/or row towards the top left corner of the 3270 display.
F8	DragBorderPlus	Move the border on which the cursor is position 1 column and/or row away from the top left corner of the 3270 display.
F10	DragBorderMinus	
F11	DragBorderPlus	

SELCOPYi Main Window

On starting SELCOPYi the main window, in which all SELCOPYi applications execute, is displayed in a maximised state. Note that this window cannot be resized.

Since the CBLe text editor application is executed automatically on SELCOPYi startup and the CBLe main window is also opened in a maximised state, the SELCOPYi main window is not normally visible until all CBLe text editor windows are closed.

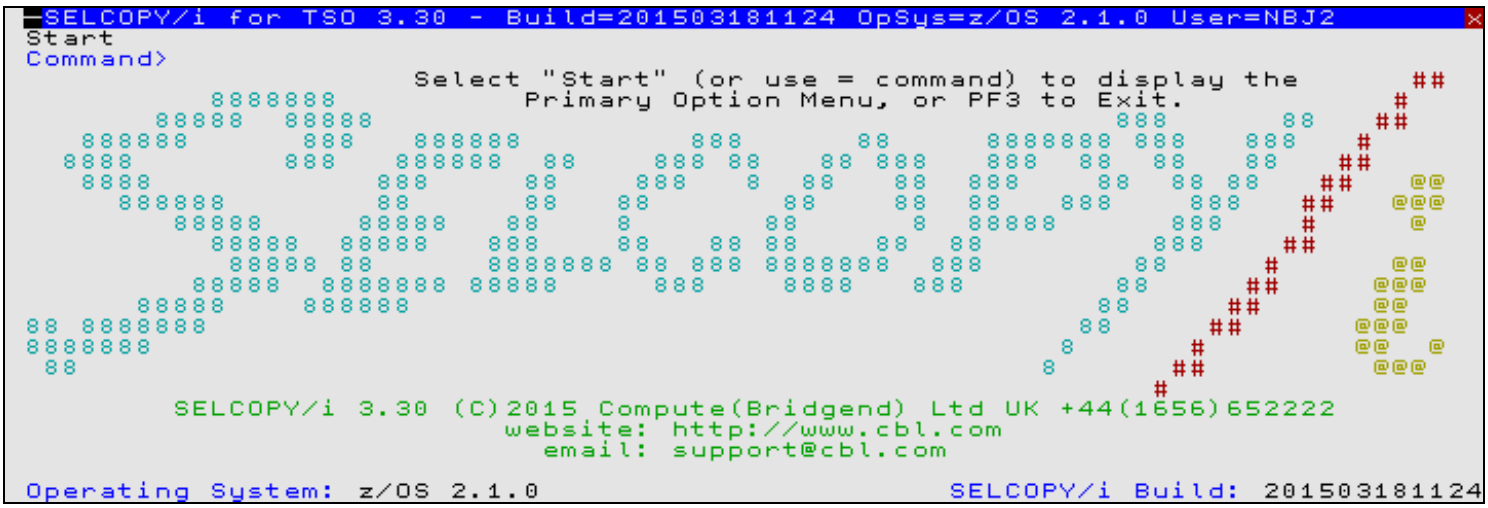


Figure 1. SELCOPYi Main Window.

The main window contains:

- A title bar. Located at the top of the window, it includes, from left to right:
  - A system menu button at the extreme left. This button accesses the system menu options.
  - If WINDOWNAMES is active, the name of the window (VCIWMAIN).
  - The SELCOPYi product name, operating environment, release and build level.
  - The operating system release.
  - The user's logon id.
  - A close button as the first character from the right.
- The menu bar. Located immediately below the title bar, it lists the SELCOPYi main window menu bar items.
- The client area. Occupying the body of the window, it contains:
  - The SELCOPYi logo, release, copyright notice and CBL contact details.
  - The operating system name and version.
  - The user id.
  - The build level of SELCOPYi. This information is useful when raising a product query with CBL.
- The command line. This may be positioned at either the top or bottom of the main window. Commands may be entered at the **Command>** prompt to invoke SELCOPYi facilities that duplicate or extend the menu facilities.

You can launch SELCOPYi facilities by doing the following:

- Select a SELCOPYi main menu bar item by positioning the cursor on the menu item text and hitting <Enter> or, if configured, double-clicking the left mouse button on the menu item.
- Executing a line command.

SELCOPYi Main Window Menu Bar

The SELCOPYi main menu bar is located at the top of the SELCOPYi main window.

The main menu consists of the following item:

Start

Open the Text Editor application main window and, depending on options specified in the Startup Settings panel (=0.1), edit the user's HOME command centre file and/or open the SELCOPYi Primary Options Menu.

When running interactively, all SELCOPYi facilities are started from the Text Editor application and run as a child windows of the Text Editor main (parent) window. The exception to this is the SELCOPY Debug application which, although started from within the Text Editor environment, operates as a separate, special instance of the Text Editor. Unless executing SELCOPYi batch (SDEAMAIN), all SELCOPYi facilities require the Text Editor environment to operate successfully. e.g. Structured Edit (SDE), DB2, File Search, Update & Copy.

Because of this, menu items for all available SELCOPYi facilities are located in the SELCOPYi Primary Options Menu and the Text Editor main window menu bar as opposed to the SELCOPYi main window menu bar.

SELCOPYi Clipboard

SELCOPYi supports a clipboard facility to allow users to Copy, Cut and Paste data between windows running in the SELCOPYi environment that support clip board functions.

At this time, clipboard facilities are only supported in CBLe and SDE edit and browse views. e.g. Data copied to the clipboard from an edit view in the SELCOPY Debug application may be pasted to an edit view in the CBLe text edit application.

Note that the SELCOPYi clipboard is not associated with any other clipboard facility offered by the system.

SELCOPYi Batch Execution (SDEAMAIN)

For z/OS systems only, a selection of SELCOPYi applications and utilities may be executed in a batch environment using program **SDEAMAIN**.

Note that SELCOPYi batch should not be confused with the **SELCOPY** and **SLC** batch programs. SELCOPY and SLC are powerful batch utilities that execute SELCOPY programming language statements to process multiple input and output data sources on a record by record basis. See the *"SELCOPY User Manual"* for details on the SELCOPY language and execution.

Supported Functionality

The SDEAMAIN program may be used as an alternative to other batch utilities, to perform such tasks as dataset record selective update, compare, copy and also for XML, CSV generation for data using COBOL copybooks.

SDEAMAIN supports execution of all SELCOPYi Data Editor edit and browse operations and also all utilities are based on data edit operations. Specifically, these are:

- Data browse / edit of VSAM, GDG, physical sequential, HFS and library member files.
- Data browse / edit of DB2 tables and views.
- CREATE STRUCTURE - SELCOPYi SDO structure creation.
- FCOPY - Advanced file copy utility.
- FSU - File search, update, copy and remap utility.
- COMPFILE - File and DB2 table compare utility.
- COMPLIB - PDS/PDSE Library compare utility.
- EXECSQL - DB2 SQL execution utility.
- CSVGEN - Comma separated variable (CSV) generation utility.
- XMLGEN - Extensible Markup Language (XML) generation utility.
- PRINT - File and DB2 table print utility.

Unless otherwise specified, the syntax defined for each of these utility primary commands is supported in its entirity by the SELCOPYi batch environment. This means that features such as record filtering and formatting using SDO structures or COBOL/PL1 copy books, may be applied to the input records or DB2 table rows processed by the utility.

When data is edited or browsed interactively using the data editor, a data edit or browse view is opened to display the data. When data edit/browse is executed in batch, no view is displayed although the data may be scrolled and altered (for edit) as though it were being viewed. All primary commands supported by the Data Editor may be executed on this virtual display of the data, including FIND, CHANGE, EXCLUDE, ONLY, INSERT, REPLACE, SORT, WHERE, etc.

Multiple datasets or DB2 result tables may be edited/browsed within the same SDEAMAIN invocation and so the concept of current data edit/browse view also exists in batch. Just as for interactive operation, executed commands will operate on the dataset or table view data referenced by the last EDIT or BROWSE command.

JCL Requirements

Although additional datasets may need to be allocated for use by the input command syntax, program SDEAMAIN requires only 2 DDNames to be allocated as follows:

SDEIN

Identifies the Data Editor primary command input source statements for SDEAMAIN.

SDEIN may be of any RECFM and LRECL. However, for RECFM=F input, the last 8-bytes of the SDEIN input records are ignored.

This input is simply a sequence of primary commands that are each executed once only in the order specified. If required, conditional command execution may be implemented by executing a Data Editor Rexx macro (see Rexx Macro Execution below.)

SDEIN input is read as a continuous stream of text, so allowing commands to span several SYSIN input lines as required. If more than one command is specified, all but the last command must be terminated with ";" (semi-colon). Text may contain charaters in mixed case which will be preserved when processed by SDEAMAIN.

Text that occurs between unquoted, balanced pairs of slash-asterisk (/*) and asterisk-slash (*/) combinations is treated as being a comment. Note that quoted text may occur within balanced pairs of quotation marks (") or apostrophes (').

SDEPRINT

Identifies the destination of SDEAMAIN processing and message output.

SDEPRINT may be of any RECFM and LRECL.

```
//FSUPRINT EXEC PGM=SDEAMAIN,REGION=0M
//FSURPT   DD DSN=JGE.FSU.D2012165.T115202,          FSU Report DSN.
//            UNIT=SYSDA,SPACE=(CYL,(5,5),RLSE),
//            DCB=(DSORG=PS,RECFM=VB,LRECL=16384,BLKSIZE=0),
//            DISP=(NEW,CATLG,DELETE)
//*
//SDEPRINT DD DISP=SHR,DSN=CBL.SYSPRINT(FSUPRINT)
//SDEIN    DD *

 /*
 |  Search all Assembler source members for "CSECT",
 |  print the formatted FSU report selecting only required columns.
 */

  FSU
        INPUT   (      CBL.CBLI330.ASM(SDE*)    )
        FIND    (      C'CSECT'                 )
      REPORT           JGE.FSU.D2012165.T115202    /* FSU Report DSN. */

 ; PRINT
       INDSN    (      JGE.FSU.D2012165.T115202
                 USING JGE.FSU.D2012165.T115202.SDO
                 SELECT ZDSN, ZMEMBER, ZRECNO, ZRECORD, *  FROM HIT
                )
       FILE
/*
```

Figure 2. Sample SDEAMAIN - SELCOPYi SDE batch job stream.

Environment Options

For both interactive and batch execution of SELCOPYi, the environment settings (e.g. MACROPATH and Data Editor options) are established by initialisation options supplied in the System and User INI datasets.

Note that there is only one System INI dataset which identifies a generic dataset name from which each user's individual User INI dataset name is derived. SELCOPYi first establishes initialisation options from the System INI dataset before applying User INI dataset option overrides.

System INI dataset initialisation options affect all SELCOPYi sessions and should only be setup by authorised users using the FIRSTUSE utility. Similarly, User INI dataset initialisation options are automatically set when a user closes an interactive SELCOPYi session. The User INI dataset should never be updated using an editor. See *Appendix C.* of the *"SELCOPY Product Suite Customisation Guide"* for detailed information on the INI file relationships and initialisation options.

Since SDEAMAIN establishes environment options based on entries in the User INI file, the output from a SELCOPYi batch job may differ when submitted by different users. The name of the derived User INI dataset is overridden by the DSN allocated to DDName **ZZSUSERI**. Therefore, if a batch job is to be user independent, ZZSUSERI may be allocated to a specific User INI dataset name or to DUMMY, if no user environment option overrides are to occur.

Rexx Macro Execution and Conditional Logic

Macros written in the Rexx programming language may be executed in any interactive Data Editor browse or edit window view. The named environemnt, CBLSDATA, may be addressed in order to execute Data Editor primary commands, including utility

commands that require the SDE Data Editor environment (FCOPY, FSU, COMPFILE, etc.)

SDEAMAIN processing also support execution of Data Editor Rexx macros. The macro source may be specified within the SDEIN input, either as a fully qualified fileid (DSN and member name) or simply as a 1 to 8 character name. In the latter case, the name will first be treated as a library member name as found in the defined MACROPATH library concatenation. If not found in the macro path, the name is treated as a DDname of a macro library dataset member.

Support for a Rexx macro DDname in batch execution provides a method by which conditional logic may be applied to the SDEAMAIN SDEIN input. e.g.

```
//SELC0001 EXEC PGM=SDEAMAIN,REGION=0M
//ZZSUSERI DD DUMMY            Suppress use of a User INI file.
//SDEPRINT DD SYSOUT=*
//SDEIN    DD *
   macro SDEMAC1;
/*
//SDEMAC1  DD DATA,DLM='%%'
/* Temporary REXX macro SDEMAC1 */

  address CBLSDATA;       'extract /region/'

  if region.3 > 64*2**20     /* 64M */
    then   do; 'edit       USER123.SELCTRN.ZZST2DAT ',
               '     using USER123.SELCTRN.SDO(ZZST2)  noprofile'
               'print file  limit 1 page'
               rv = 0
           end
    else   do; say 'Edit cancelled. Auxiliary Edit Suppressed.'
               rv = 22
           end

  return rv
%%
```

Figure 3. Sample SDEAMAIN - SELCOPYi SDE batch job stream using a Rexx macro.

SELCOPYi Interactive Help

The SELCOPYi Interactive Help windows are basic HTML browse windows started with links to the SELCOPYi suite of HTML help files located via the SELCOPYi INI variable Help.DefaultPath.

SELCOPYi Help may be obtained via the following:

- Select 'Help' from the the CBLe main window menu bar.
- Enter command HELP on the command line of any window.

Context sensitive help windows are displayed by performing either of these functions within the appropriate SELCOPYi window. e.g. Executing HELP in a Library List window opens the help window for the topic List Library Members.

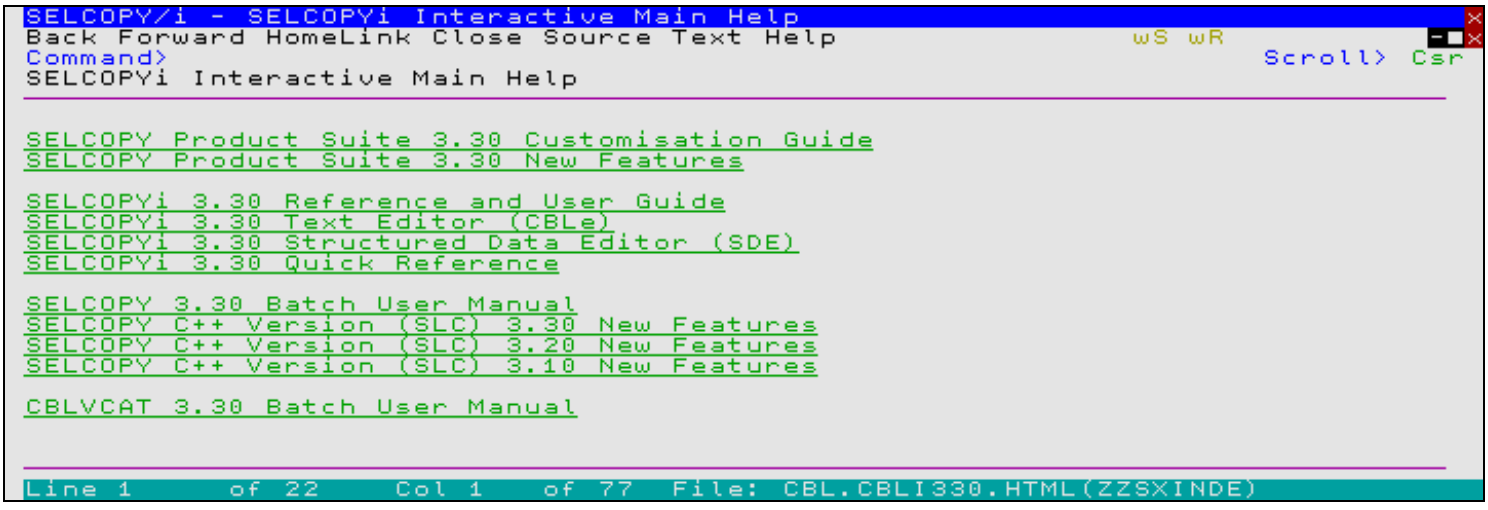


Figure 4. SELCOPYi Main Help Menu window.

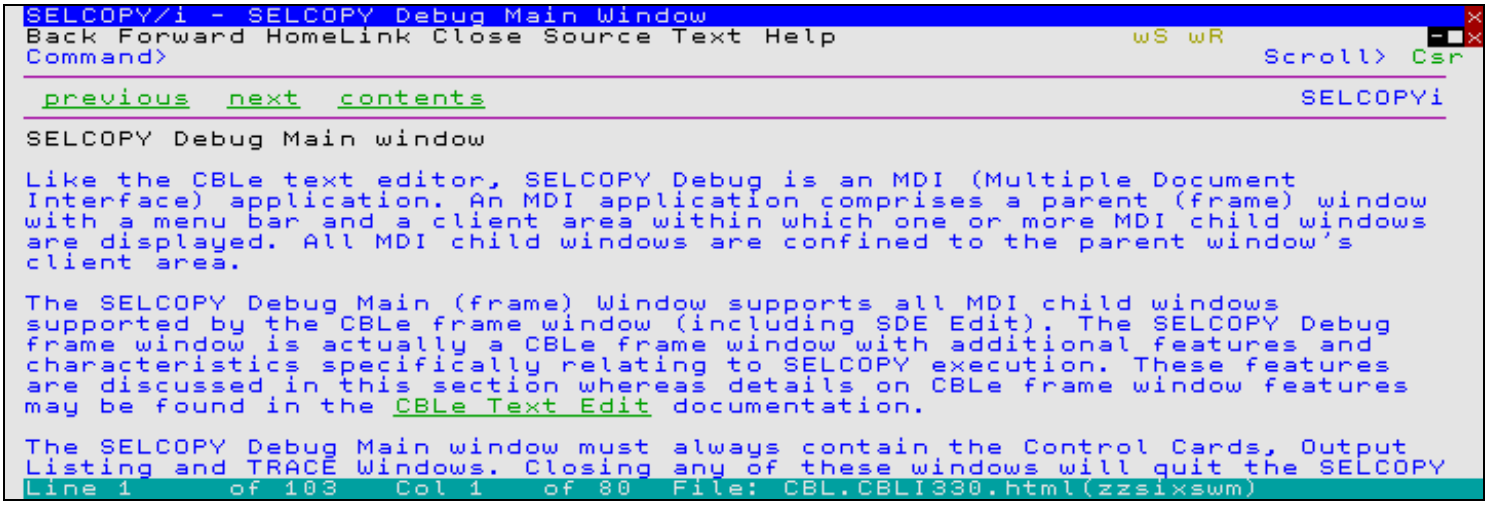


Figure 5. Help window for SELCOPY Debug.

`Back`
: Display the HTML page that occurs immediately prior to the current page in the stack of viewed pages.

`Forward`
: Display the HTML page that occurs immediately after the current page in the stack of viewed pages.

`Home`
: Display the defined Home page.

`Close`
: Close the current HTML browse window.

`Refresh`
: Refresh (reload) the current HTML page.

`Find`
: Open a dialog window to locate lines in the current HTML page that contain a specified string.
Not yet supported.

`Source`
: Open a CBLe text editor window to edit the source for the current HTML page.

`Options`
: Tailor options for the current HTML browse window.
Not yet supported.

`Text`
: Open a CBLe text editor window to edit the current HTML page as plain text with file name UNTITLED.

`Help`
: Display this help page.

`Location>`
: Specify an explicit HTML source fileid. If only a file name is specified, then the HTML browse window will search the Help.Defaultpath library for that file name.

  This allows the user to display any file containing basic HTML tags that is not necessarily associated with the SELCOPYi suite of help files.

SELCOPYi Help Topic Index Window

The SELCOPYi help topic index window displays a list of the titles of all the distributed HTML help files from which a help topic can be selected for viewing.

Note that the Help Topic Index List window is a standard SELCOPYi List Window and as such supports all the features of this type of window.

Searching the list of help topics

The Help Topic Index List window has a search input field which can be used to filter the list of help topics. Filtering and sorting can also be done from the command line as for all types of SELCOPYi List Windows.

If the search string consists of more than one blank delimited sub-string, only those help topics will be displayed which have titles containing all of the sub-strings.

The search string is used to generate a filter command with a WHERE clause. For example if the following string is entered at the **Search>** prompt:

```
unix command
```

then the generated filter command is:

```
where title<<unix & title<<command
```

and only those help topics which have titles containing both the strings *unix* and *command* will be displayed.

Help topic list fields

A Help Topic List row contains the following fields:

`Component`
: The software component to which the help topic applies:

SELC/batch	Batch SELCOPY.
SLC	Batch SLC.
SELCOPYi	The interactive window environment.
DataEdit	The structured Data Editor.
TextEdit	The Text Editor.
CBLVCAT	CBLVCAT.
Custom	Product Suite Customisation.
QuickRef	SELCOPYi Quick Reference.
NewFeature	Product Suite new features.
Misc	Additional miscellaneous topics.

`Title`
: The title of the HTML help document.

`Member`
: The member name of the HTML help document in the help library.

Selecting a help topic

A Help Topic can be selected and viewed by placing the cursor anywhere in the list row containing the topic and pressing the enter key.

Opening the help topic index window

The help topic index list window can be opened in the following ways:

- Select 'Help Topic Index ...' from the the CBLe main window menu bar.
- Enter command HELPINDEX (HIX) on the command line of any window.
- If the HELP command is issued for a topic which does not exist, then the help topic index list window is opened and the given topic is used as a search string.

File Object Names

SELCOPYi is predominantly used to edit, browse, search, update, copy, compare and reformat data stored in file objects on DASD volumes.

SELCOPYi utilities support for data stored in DB2 database tables is detailed in chapter DB2 Utilities (=12).

File/Dataset Name

On all the mainframe operating systems in which SELCOPYi operates, file data objects are referenced using standard naming rules and conventions as defined by the operating system for the particular file system.

```
BROWSE      NBJ2.SMPL001.X1IQ023.KSDS
INFO        SYSA.OEM.SELC340.SZZSLOAD
FSU   INPUT(SYSA.OEM.SELC340.SZZSLOAD(SELCOPY))   FIND(X'0A0D')
EDIT        NBJ2.SELC340.INSTALL.JCLLIB(ZZSSMPE)

EDIT        /home/nbj/selcopy/330/ssdcl04.s330.log
```

Quotation marks (") or apostrophes (') may be used to enclose a z/OS file object name though are usually unnecessary. An instance where it is necessary is in a z/OS HFS or ZFS filesystem fileid which includes blank characters.

```
EDIT       "/home/nbj/test input data"
```

For edit or browse of uncataloged z/OS datasets and libraries, the file name specification must include the serial id of a volume on which the dataset exists. The volume id is specified before the dataset name, separated by a ":" (colon) character. For uncataloged, multi-volume datasets, the edit view will display only data that exists on the specified volume. e.g.

```
VIEW        CBLM15:TEST.UNCATLG.DATA001
```

For CMS fileids, SELCOPYi also supports use of a single "." (dot/period) in place of blank characters between the file name, type and mode.

```
VIEW        CMSPROF EXEC A5
EDIT        PRONBJ.XEDIT.A
```

Generic File Object Names

Where generic names are appropriate (e.g. as input to file lists, search tools and panel file selection fields), SELCOPYi supports specification of one or more wildcard characters ("%", "*" and/or "**") within the name. Use of these wildcards allows the user to specify a fileid mask that may be matched by zero or more file names. e.g.

```
LC     NBJ%.*.*IQ*.KSDS
LLX    SYSA.OEM.**.*LOAD
LP     /home/nbj/selcopy/330/ss*.s330.l%%

FL     PROF%%% * *
```

Note that SELCOPYi for **z/VSE** does not support this form of generic file name specification. Generic z/VSE file names may be used only for displaying VSAM catalog lists, in which case the CBLVCAT generic file name syntax must be used. (See LISTVCAT KEY operand in the *"CBLVCAT User Manual"*.)

Generic z/OS Dataset Name

For **z/OS** data set names, the following wildcard characters may be used:

*	A single asterisk represents a DSN qualifier, or zero or more characters within a DSN qualifier.
**	A double asterisk represents zero or more qualifiers within a DSN. Double asterisk must be preceded or followed by either a "." (dot/period) or a blank. It may not precede or follow an alphanumeric character.
%	A single percent sign represents exactly one character, other than "." (dot/period), within a DSN qualifier. Up to 8 percent signs can be specified in each qualifier.

If the DSN mask ends with "." (dot/period), then the trailing "." is stripped and the rest of the mask is unchanged. e.g.

```
LISTCAT       DEV.                  becomes: LISTCAT     DEV
LC            DEV*.                 becomes: LISTCAT     DEV*
LISTDATASET   DEV.OEM.TRSPAN*.      becomes: LISDATASET  DEV.OEM.TRSPAN*
LD            DEV.*.*SAMP%%.        becomes: LISDATASET  DEV.*.*SAMP%%
```

If the DSN mask does **not** end with "." (dot/period), a default trailing wild card suffix of ".**" or "*.**" is automatically appended to the mask.
Suffix ".**" is used if the DSN mask ends with "*" (asterisk) or is a single qualifier. e.g.

```
LISTCAT       DEV                   becomes: LISTCAT     DEV.**
LC            DEV*                  becomes: LISTCAT     DEV*.**
LISTDATASET   DEV.OEM.TRSPAN*       becomes: LISDATASET  DEV.OEM.TRSPAN*.**
LD            DEV.*.*SPA*           becomes: LISDATASET  DEV.*.*SPA*.**
```

Suffix "*.**" is used otherwise. e.g.

```
LD            DEV.OEM.TRSPAN        becomes: LISDATASET  DEV.OEM.TRSPAN*.**
LD            DEV.*.*SPA%           becomes: LISDATASET  DEV.*.*SPA%*.**
LC            SYS1.*.Z19            becomes: LISTCAT     SYS1.*.Z19*.**
```

Generic z/OS HFS/ZFS fileids

For files in a z/OS HFS or ZFS file system, wildcard characters are recognised within the file name only. i.e. "%" and "*" are not treated as wildcard characters if specified in a directory name within a file path specification.

As for DSN masks, wildcard character "%" represents exactly 1 character in the file name mask and character "*" represents zero or more characters. Unlike DSN masks, these wilcard characters may represent character "." (dot/period) in the file name mask because a file name is not restricted to a number of up to 8 character qualifiers.
Note that wildcard "**" is equivalent to "*". e.g.

```
LISTPATH     /etc/ssh/*host*
```

Generic z/OS PDS/PDSE Library Member Names

z/OS PDS and PDSE library members have up to 8 character names and are specified in parentheses immediately following the library data set name.

As for DSN masks, wildcard character "%" may be used to represent exactly one character within a SELCIOPYi generic member name mask and wildcard character "*" to represent zero or more characters in the mask.
Note that wildcard "**" is equivalent to "*". e.g.

```
LISTLIBRARY  SYSA.OEM.SELC340.SZZSLOAD(%ZZS*)
LL           NBJ.SELCTEST.JCLLIB(*DCL*)

LLX          NBJ.**.JCLLIB(*DCL*)
```

Furthermore, multiple blank or comma delimitted member masks may be specified in parentheses after the library DSN. If the up to 8 character name of a library member matches any of the member masks provided, then it will be selected. e.g.

```
LLX          NBJ.**.JCLLIB(*DCL*  %%CAL%  SVC*)
LLX          SYSA.OEM.**.*LOAD(%ZZS*  SELC*  *BL%%)
```

Generic z/VM CMS File Names

z/VM **CMS** file names comprise a filename, filetype and filemode. SELCOPYi supports specification of a CMS file name with filename, filetype and filemode qualifiers separated by either one or more separating blank characters or a single "." (dot/period) with no intervening blank characters.

As for DSN masks, wildcard "%" represents exactly 1 character and wildcard "*" represents zero or more characters in the filename or filetype masks only. Wildcard "*" may be used in place of the filemode to imply all accessed minidisk letters of any filemode number.

In most instances, filemode may be omitted and defaults to "*". Depending on the SELCOPYi utility being used, omitting the filetype will either imply a default of "*" or, in the case of an edit or browse operation, the same filetype as the CMS file being viewed in the focus edit window. e.g.

```
FILELIST     *       EXEC C1
FL           SELCOPY.*              same as: FILELIST  SELCOPY * *
FL           *AV%    *    A5
```

z/OS PDSE Library Member Generations

In z/OS 2.1, PDSE version 2 libraries were introduced which include support for member generations (via the MAXGENS library data set allocation attribute).

When changes are made to member data belonging to a PDSE V2 library that supports generations, a new generation is automatically created for the member when the data is saved. The image of the member data before the change occurred is referenced as relative generation number -1 and the relative generation number of each previous image of the member data is decremented by 1.

See IBM publication *"z/OS DFSMS Using Data Sets"* for information on PDSE version and member generations.

To reference an individual member generation, SELCOPYi supports specification of an absolute or relative generation number following the member name with a separating "." (dot/period) in between. Note that blank characters are not permitted within a member generation specification. e.g.

```
EDIT    NBJ.GEN25.EXEC(APESUB.-5)     relative generation number.
E       NBJ.GEN25.EXEC(APESUB.56)     absolute generation number.
BROWSE  NBJ.SELCI.SDO(DB2FUNC1.-1)
VIEW    NBJ.SELCI.SDO(DB2FUNC1)       same as: VIEW NBJ.SELCI.SDO(DB2FUNC1.0)
```

The base (prime) generation may be identified a s generation 0. In SELCOPYi, generation 0 is treated as the base generation's relative generation value. Although not yet assigned to the base member generation in the PDSE, SELCOPYi attributes a positive absolute generation value to the base generation. This value is the next absolute number in the member generation sequence.

As with all member generations, SELCOPYi allows reference to the base generation via its relative or (SELCOPYi attributed) absolute generation number. e.g. Where 12 previous generations have been created for member DB2FUNC1, the base generation may be referenced in SELCOPYi as DB2FUNC1.13.

```
VIEW    NBJ.SELCI.SDO(DB2FUNC1.13)   same as: VIEW NBJ.SELCI.SDO(DB2FUNC1.0)
```

Generic z/OS PDSE Library Member Generations

Where applicable, SELCOPYi supports specification of a generic PDSE member generation.

A null or wildcard character "*" may be specified in place of the generation number to indicate that all generations must be selected. Alternatively, a relational operator may be inserted between the "." (dot/period) and the (absolute or relative) generation number.

Supported relational operators are:

Operator	Description
=	Equal.
\= ¬= != <>	Not equal, less than or greater than.
>	Greater than.
<=	Less than or equal.
<	Less than.
>=	Greater than or equal.

Member generations that satisfy both the member mask and generation mask will be selected. e.g.

```
LL      NBJ.JCLLIB(*.*)              list all member generations.
LL      NBJ.JCLLIB(*.)               same as:  LL  NBJ.JCLLIB(*.*)
LL      NBJ.JCLLIB(*.<*)             same as:  LL  NBJ.JCLLIB(*.*)

LL      NBJ.JCLLIB(CALL*.>=12)       list all member generations whose member
                                     name begins with CALL and whose
                                     absolute generation number is greater
                                     than or equal to 12.

ERASE   NBJ.JCLLIB(NBJX.<0)          erase all generations except the base
                                     (i.e. relative generation 0).


LL      NBJ.JCLLIB(SS*.-1)           list all member generations whose member
                                     name begins with SS and whose
                                     relative generation number is -1.

LL      NBJ.JCLLIB(SS*.>=-3  ADA%%%%.>=-5   CBL*.0)
                                     list all member generations whose
                                     entries match one of the member
                                     and generation masks.
```

Window Classes

Every SELCOPYi window belongs to a named window class. Window classes define a set of window characteristics that are common to all windows assigned the same window class.

These characteristics include the window's appearance such as the existance of a command prompt, menu bar, scroll field, input fields; if present, the contents of the menu bar; non-KeyList PFKey assignments and the presentation format of data to be displayed in a window's display area.

A comprehensive list of SELCOPYi window classes may be found in *Appendix A - SELCOPYi Window Classes*.

Characteristics of CBLe Text Edit window classes, including Structured Data Edit document views, are detailed in the *SELCOPYi Text Editor* documentation.

Many window classes are one-to-one with an individual SELCOPYi feature and their characteristics are documented with that particular feature. (e.g. window class EDTWSORT is specific to the Text Edit SORT dialog window.) Characteristics and behaviour of those window classes that are common to more than one SELCOPYi feature are documented in this chapter.

Storage Display Windows

Storage display windows (window class HEXDUMP, STORAGE and EDTWHEXE) provide a view of an area of storage in dump format.

SELCOPY/i windows that utilise storage display windows are:

- CBLNAME display window.
- SELCOPY Debug Workarea windows.
- SELCOPY Debug POS windows.
- CBLe text edit line HEX dump views.

Storage Window Display Format

The length of storage data displayed may be restricted by the type of window opened. e.g. The amount of data displayed in a CBLe HEX window is restricted to be the length of the focus line.

Each row of a storage display window has the following format:

Field Width	Type	Description
8	Hex	Address in storage of the displayed data.
6	UInt	Displacement from the start address of the displayed data.
8,16,32 or 64	Hex	Data in storage in hexadecimal format. (4, 8, 16 or 32 bytes depending upon window size).
4,8 ,16 or 32	Char	Data in storage in character format. (Field width adjusts automatically to match hexadecimal display width.)

The format of the display may be updated using the options popup menu which may be opened using the SHOWPOPUPMENU command or via the system menu button of the storage display window. By default, the SHOWPOPUPMENU command is assigned to PF5 in storage display windows.

Storage Window Resizing

User resizing of a storage display window's width will always be adjusted by SELCOPYi to display one of the valid data display widths. i.e. 1, 2, 4 or 8 words of hexadecimal data plus its equivalent character representation if required.

If the window's width is increased or decreased by one column, the window's width is rounded up or down respectively to equal the next valid display width. e.g. When using DRAGBORDERPLUS and DRAGBORDERMINUS (assigned to PF8/PF11 and PF7/PF10 respectively) on either of the window's vertical borders.
If the window's width is increased or decreased by more than one column, the window's width is always rounded down to equal the next valid display width.

Storage Display Navigation

The displacement field in the first row of a storage display window is an enterable field (highlighted in red by default.) This field may be overwritten with a displacement, from the start address, of the byte that should be displayed first in the storage window's display area.

Line commands UP CURSOR and DOWN CURSOR may be used to navigate the storage display window. By default, UP CURSOR is assigned to PF7 and DOWN CURSOR is assigned to PF8.

Storage Data Manipulation

Some storage display window invocations allow the data to be updated simply by overtyping the existing character or hexadecimal representation and then, to commit the change, hitting <Enter>.

Only areas of storage that are not write protected may be altered. Beware when altering storage as this could adversely affect programs that utilise the updated area of storage.

List Windows

A list window is a child window with a parent window class of LISTFRAM, LISTFILE, VCIWEXEC, WINWIPO0 of WINWHIX0.

List windows display information as a table of rows and columns. The content of the list columns is described by data elements, each having a name and data type, and defined by a Field Descriptor Block (FDB).

SELCOPYi windows that have child list windows include DB2 Object lists, DASD lists, Dataset lists, Execute CBLVCAT and Execute IDCAMS windows, APF Authorised Library lists and FDB lists.

In addition to any features defined by a list window's parent window class, list windows provide support for features described in this section.

- List Window Status Bar
- List Window Menu
- Selecting, Sorting and Filtering
- Sorting with the Cursor
- List Entry Location
- List Entry Display
- List Window Prefix Area

List Window Status Bar

All list windows have a status bar that occupies the last row of the window display area.

The list window status bar displays the following information:

Status bar display	Description
Line *c_row* of *n_rows*	Identifies the list current row value (*c_row*) and the total number of rows in the list (*n_rows*).
Col *c_col* of *n_cols*	Identifies the list current column value (*c_col*) and the total number of list columns, including non-scrollable list columns, in the list (*n_cols*).
Views *n_views*	Identifies the number of saved list views (as described by SELECT, WHERE, SORT clause combinations) that exist for the current list data. (*n_views*)
select_clause <*where_clause* <*sort_clause*> >	Identifies the current view of the list data.

List Window Menu

All list windows have the following menu items:

View	This is a popup menu which lists all the views which have been made of the current list. You can select a view from this menu.
Back	Select the previous view. This is equivalent to executing command **BAck** in the current list window.
Forward	Select the next view. This is equivalent to executing command **FOrward** in the current list window.
FDB	Display the Field Descriptor Block for the list. This is equivalent to executing command **FDB** in the current list window.
Text	Open a CBLe text edit window containing a text version of the list. This is equivalent to executing command **TEXT** in the current list window.
Refresh	Refresh the contents of the list window so that all column fields reflect the current status. This is equivalent to executing command **REFRESH** in the current list window.
Help	Open the help window for the list. This is equivalent to executing command **HELP** in the current list window.

View List Display

A SELECT clause and/or WHERE clause, executed as a CLI command, creates a new view of the list data. On each execution of one of these CLI commands, the command stream is recorded as a single entry at the end of the list window's View List Display.

This allows the user to select and filter list columns and rows and then easily recall any previous view of the data.

The View List Display is a drop down menu available by selecting the "View" List menu bar item. Any previous view may be selected by positioning the cursor on the required SELECT and/or WHERE clause entry and hitting <Enter> or, if configured, double-click the left mouse button on the entry.

Alternatively, the display's view of the data may be switched to view immediately prior to or following the current view by selecting "Back" or "Forward" respectively from the List menu bar.

Field Descriptor Block (FDB)

The FDB window may be opened using the following:

- Select 'FDB' from the list window menu bar.
- Enter command **FDB** on the list window command line.

Information about the data displayed under each column of a List window is referenced via an internal SELCOPY/i data structure. This structure includes, or addresses, fields that define the column data attributes. e.g. column name, column data type, column data length, etc.

The Field Descriptor Block (FDB) maps this internal structure and so provides information for all fields in the List window.

FDB is primarily used as an aid to performing List window SELECT, SORT and WHERE clause commands.

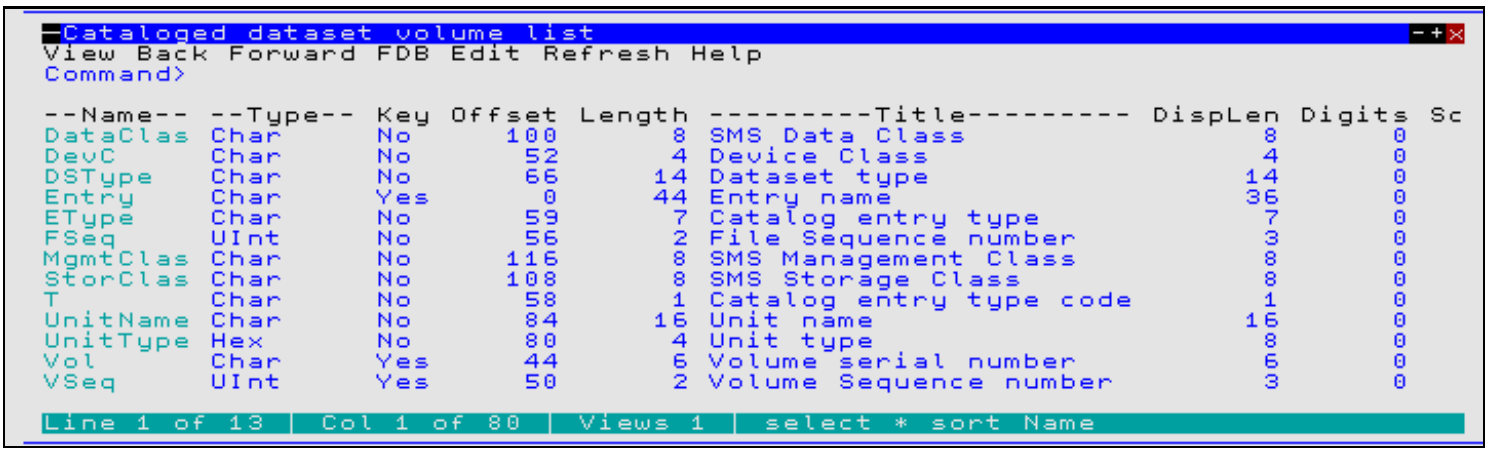


Figure 6. FDB for Catalog List window.

`Name`

Specifies the field names that constitute the List window column headers. These entries are used when selecting columns and sorting/filtering rows to generate new list views.

`Type`

The data format in which data for that column is stored.

`Key`
: Identifies whether or not the column is a key column.

  If the column is a key column and is either the first column in the list or immediately follows another key column, then it is always in view even when scrolling the list view left or right.

  Furthermore, if a key column selected as the first column of a list contains duplicate entries, then, when these entries are sorted together, only the first, in view occurrence of the key field value will be displayed. Subsequent list rows containing the duplicate key field values are displayed with ditto/quotation marks (") in the key field. If sorting on a different column separates these duplicate key field values so that they no longer appear on consecutive list rows, then the field values are once again displayed replacing the ditto mark. The display of values in all other columns are unaffected by this list window feature.

`Offset`
: The offset within the structure of the column data.

`Length`
: The length of the column data field within the structure. Note that if one or more levels of indirection to the column data field exist, then the structure contains an address field length 4.

`Title`
: Descriptive name for the field column.

`Displen`
: The length of the longest entry displayed in the column.

`Digits`
: The number of decimal digits (precision) displayed for column data of numerical data type.

`Scale`
: The number of scale digits (fraction digits) displayed for column data of numerical data type.

Because the FDB window is itself a List window with its column data attributes referenced by an internal data structure, it too may be described by an FDB and is subject to all supported list window functions (SELECT, SORT WHERE, etc.)

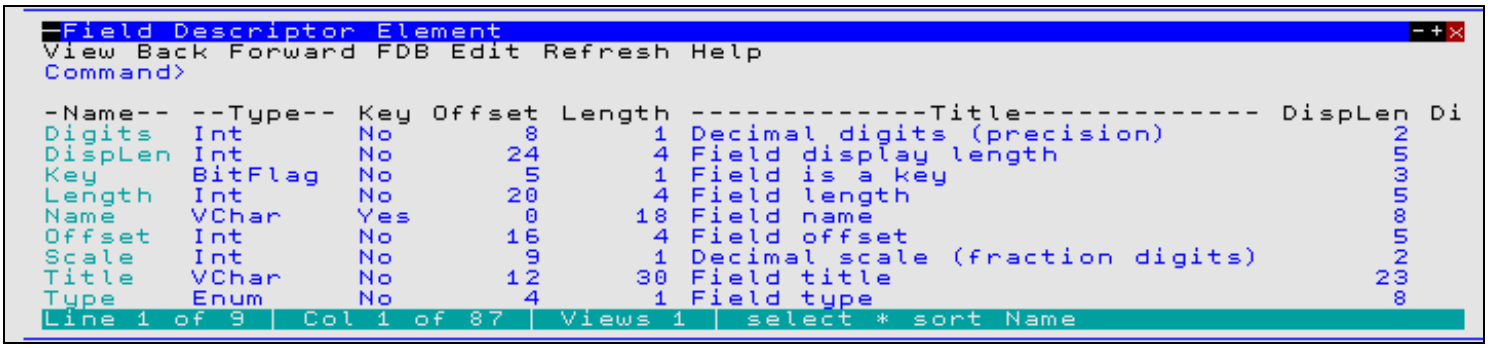


Figure 7. FDB for FDB window.

Edit View

The contents of the current list view may be edited and saved to disk.

- Select 'Text' from the list window menu bar.
- Enter command **TEXT** or **EDIT** on the list window command line.

A CBLe text editor window is opened, the list is loaded into CBLe storage and edited. The edited view is given a generic title "UNTITLED". On saving the text for the first time, the user is prompted to provide a valid "Save as" fileid.

Note that if INSTANCE=SINGLE is set in the (Edit) section of the SELCOPY/i INI file and a CBLe editor window is already open, then the list will be edited in a new document window of the existing CBLe window.

Zoom View

The contents of a list entry may be displayed vertically in a single entry, zoomed view format. i.e each column value belonging to the selected entry is displayed in a separate row of the zoomed window view.

- Enter prefix command **>** against the required list window entry.

The zoomed window view is a separate list window which displays the selected list entry's field names, values are field descriptions (as reported in the FDB list window view.)

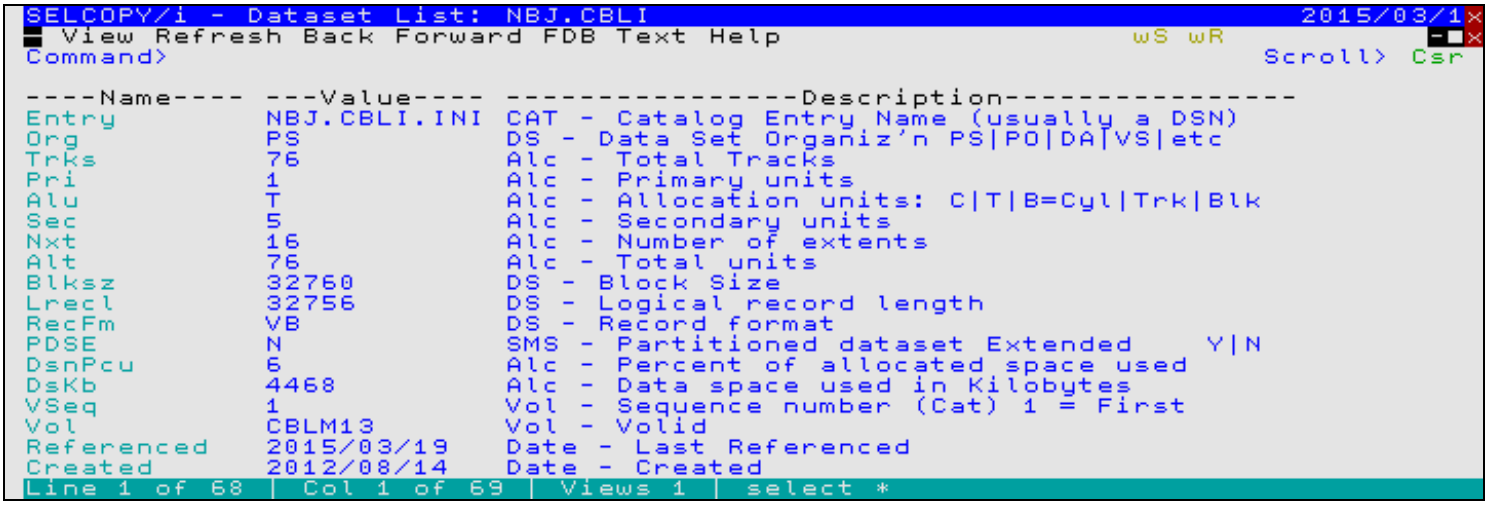


Figure 8. Zoom Window View.

Columns Displayed

Name	Type	Description
Name	ALPair	Field name
Value	ALPair	Field Value
Description	ALPair	Field description

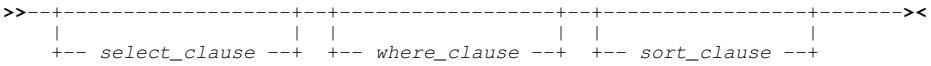
Selecting, Sorting and Filtering

Each list has a basic set of column data which is defined by the Field Descriptor Block (FDB) associated with the list. You can view the list of columns by selecting the FDB menu option or entering the FDB command on the list window command line.

You can modify the display by selecting your own list of columns from this basic list, specifying a filter to restrict the rows displayed and specifying a sort order. You do this by entering the select command on the list window command line. The select command syntax is similar to the SQL SELECT statement. It consists of one or more of the following:

- A SELECT clause which determines the columns displayed.
- A WHERE clause which filters the rows displayed by imposing a condition on the values in one or more columns.
- A SORT clause (ORDER BY clause) which displays the rows in an order determined by the values in one or more columns. (See also Sorting with the Cursor.)

Unlike SQL, these clauses can be given in any order, and any of them can be ommitted. The general syntax is:



Each time you issue a select command with a select or where clause you create a new view of the list. If you issue only a sort clause no new view is created, but the sort order is changed for the current view.

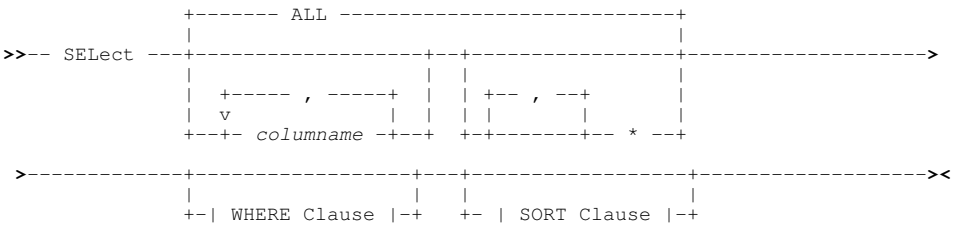
SELECT, WHERE and SORT clauses are not cumulative. Therefore, their execution replace any previously executed clause of the same type. e.g. Execution of a WHERE clause will not perform a logical AND with any previously executed WHERE clause, but will replace it instead. In addition to this, the clauses are hierarchical so that execution of a SELECT clause will undo any active WHERE and/or SORT clause, execution of a WHERE clause will undo any active SORT clause.

The set of views which you have is listed in the View menu option, from which you can select a view from the ones you have created with the select command.

SELECT Clause

SELECT is also a Text Edit CBLe option used to set the selection level of a group of edited records, and also an SDE Data Edit primary command used to select formatted columns for display.

Syntax:



Description:

Specified as a CLI command or as a parameter on WHERE or SORT, the SELECT clause is used to identify field columns for display and the order in which they appear.

Use of a SELECT clause is not cumulative and so will replace those columns currently selected for display. It also resets any prevailing WHERE and/or SORT clause specifications.

Note that the last execution of a SELECT clause CLI command is stored and applied to any new List window of the **same type** (e.g. Library List window). This occurs whether the list window is opened within the same SELCOPY/i session or opened after the current SELCOPY/i session is ended and a new session started.

Executing a SELECT clause will add a new entry to the View List Display drop down menu.

Parameters:

`ALL`
: ALL returns all columns.

`columname`
: Name of the list window column to select.

  Multiple column names must be separated by commas and/or one or more blanks. Column names may be specified in any order and any number of times.

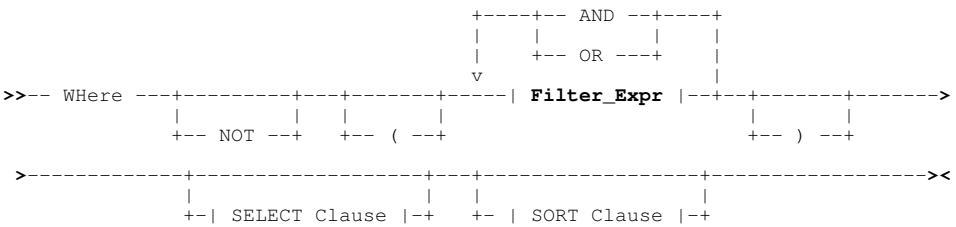
`*`
: Display all remaining column names that have not already been selected, in their default order of display. This is the same as ALL if no other column names are specified.

`WHERE Clause`
: Any valid WHERE clause used to filter list rows.

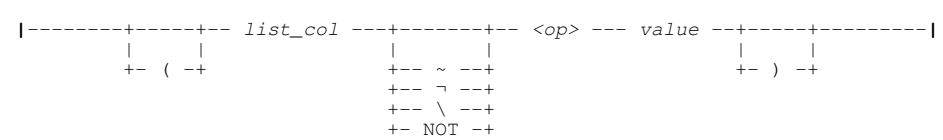
`SORT Clause`
: Any valid SORT clause used to sort list rows.

WHERE Clause

Syntax:



`Filter_Expr:`



Description:

Specified as a CLI command or as a parameter on SELECT or SORT, the WHERE clause is used to restrict the rows displayed in the list.

A WHERE clause is a logical combination of one or more simple filter expressions that define search criteria used to select the required list rows. When applied to an individual list row, each filter expression in the where clause resolves to either 1 (true) or 0 (false). If the logical result of all the filter expressions is 1, then the row satisfies the where clause criteria. Only those rows which satisfy the where clause are displayed in the view of the list.

Use of a WHERE clause is not cumulative and so will replace any prevailing filter of rows in the display. It also resets any prevailing SORT clause specification. Columns displayed by the prevailing SELECT clause, are unchanged.

Executing a WHERE clause will add a new entry, which includes the prevailing SELECT clause, to the View List Display drop down menu.

Parameters:

`( ) AND OR NOT`
Logical operators supported by the where clause and supported symbolic equivalents are as follow:

`( )`	Left and Right Brackets (parentheses - X'4D' and X'5D').
`AND`	`& (ampersand - X'50')`
`OR`	`¦ (broken bar - X'6A'),	(vertical line - X'4F')`
`NOT`	`~ (tilde - X'BC'), ¬ (not sign - X'5F'), \ (backslash - X'E0')`

Notes:
1. To avoid confusion, it is recomended that parentheses be used where more than two filter expressions are specified in order to establish logical AND/OR precedence.
2. Parentheses must be balanced so that there are an equal number of left and right parentheses in the where clause.

`Filter_Expr`
A filter expression that tests a single column within the list display. The contents of *list_col* are tested against a test value specified by *value* to establish a TRUE (1) or FALSE (0) condition.

`list_col`
The name of list column whose contents will be tested in this filter expression.

Where *list_col* is a numeric field, *value* must be numeric and an arithmetic compare is performed. Similarly, if *list_col* is a character field, then *value* is treated as a character string with blank padding on the right. Check the field Type in the FDB to determine whether the field contains numeric or character data.

If the data type of the test value is not compatible with *list_col* then an error will be returned.

`~ | ¬ | \ | NOT`
The symbols ~ (tilde), ¬ (not sign) and \ (backslash) represent the logical NOT operator and reverses the TRUE or FALSE condition established by the comparison operator <op>.

`<op>`
Comparison operator specified as one of the following:

`=`	Equals.
`<>`	Not Equals.
`<`	Less than.
`>`	Greater than.
`<=`	Less than or equals.
`>=`	Greater than or equals.
`<<`	Contains. Applicable to character columns only, returns TRUE if *value* is a sub-string of *list_col*.
`>>`	Begins. Applicable to character columns only, returns TRUE if *value* is a sub-string at the start of *list_col*.

`value`
The value to be tested against the contents of *list_col*.

If *list_col* is a character field, *value* is treated as a case insensitive character string unless it is enclosed in (') apostrophes or (") quotation marks. Apostrophes or quotation marks ensure case sensitivity and are mandatory if *value* is the same as a list column heading or if it contains any of the comparison operators or blanks.

`SELECT Clause`
Any valid SELECT clause used to select column headers.

`SORT Clause`
Any valid SORT clause used to sort list rows.

Example:

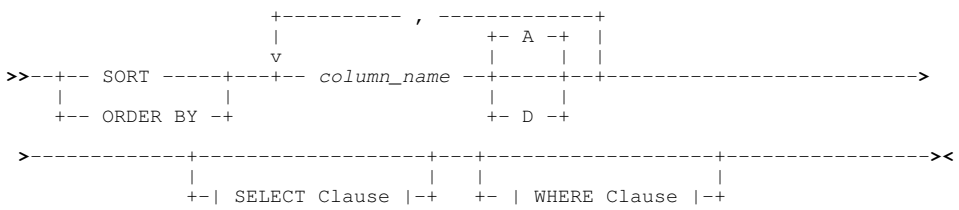
```
where    col1 = 'A'   and   (col2 << 'C' or col3 > 4)
```

List only those rows for which col1 is equal to A and either col2 contains C as a substring or col3 is greater than 4.

Note: the quotes around the literal strings are not needed unless there are columns in the list with name A or C and that the blanks separating the elements of the expression are optional.

SORT (ORDER BY) Clause

Syntax:



Description:

See also Sorting with the Cursor.

Specified as a CLI command or as a parameter on SELECT or WHERE, a SORT clause is used to modify the order in which the rows are displayed in the list.

Use of a SORT clause is not cumulative and is based on the list's default sort order. Therefore any prevailing sort order is ignored. The prevailing SELECT and WHERE clauses are unchanged.

Unlike SELECT and WHERE, a SORT clause specified as a CLI command is not added to the View List Display drop down menu.

Parameters:

column_name
: Name of the List Window column on which to sort.

  Column names must be supplied in the order in which they appear in the list heading (or in the FDB).

`A | D`
: The sort order is specified with a list of column sort specifications which consist of a column name followed by a sort direction. The sort direction is given as A for ascending or D for descending.

  If the sort direction is not provided, it defaults to ascending.

  Commas must be used to separate multiple column sort specifications whereupon intervening blanks are permitted.

`SELECT Clause`
: Any valid SELECT clause used to select column headers.

`WHERE Clause`
: Any valid WHERE clause used to filter list rows.

Sorting with the Cursor

A quick way of sorting a list view is to place the cursor on the heading of the column on which you want to sort and then press the <Enter> key. Alternatively, if configured, simply double-click the left mouse button on the list column header.

For fields that contain timestamps and/or datestamps, the first time the sort is actioned on the column header, the data is arranged in descending order. For all other column fields, the data is initially arranged in ascending order.

Subsequent sorting on a column header using this method will reverse the order in which data in that column was last sorted.

List Entry Location

If the first field column of the list meets the requirements detailed below, then the occurrence of this field in the first row of the display area (list current row) is marked as being enterable (text is yellow underscore). Simply overtyping this field with the required field string value will scroll the display so that the first row containing this value becomes the new current row.

1. The column is keyed. (Key=Yes)
2. The column contains character data. (Type=Char)
3. The column's fields are sorted in ascending order.

This information is available from the FDB.
Unless a SELECT clause has been executed which changes the first column of the display, these criteria are usually always met by list windows of class LISTFILE.

Partial strings may be entered but residue from the overtyped field should be deleted or blanked over, otherwise it is included as part of the typed string. Similarly, strings entered in lower case are automatically upper cased.

Starting at the row following the list current row, field values occupying the first column are each compared with the string value entered by the user until one is found which is greater than or equal to this string value.

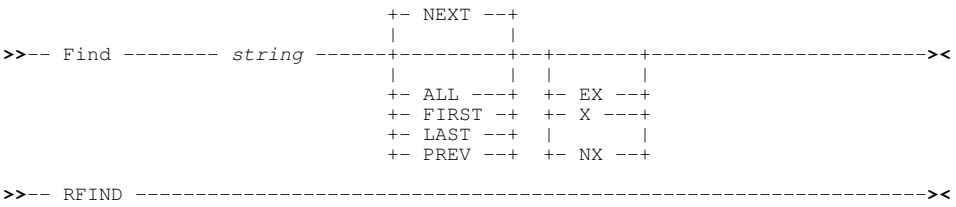
The target row is the row containing an exact match for the specified field string value or else the row immediately before a row containing a field value which is greater than the specified value. The list is scrolled so that the target row becomes the list current row.

Location of list entries is also achieved using the following:

- FIND/RFIND Command
- LOCATE Command
- S Command

FIND Command

Syntax:



Description:

Find is used to scroll the display to a list entry that contains a match for the specified search string **anywhere** within the field occupying the first column of the display. If no match is found for **string** then no scrolling occurs.

Note that a character logical compare is performed on the field's source data, not on the displayed value. Therefore, FIND may not be effective if the first column field has a source data type which is not character. (Check the FDB for Field Type=Char.)

Following a successful FIND operation, `RFIND` (assigned to PF5 by default) may be used to repeat the search for the remaining list entries. If the string is found in an excluded line, then the line will no longer be excluded in the display.

Parameters:

`string`
: A character search string used in the compare against data within fields in the first column of the display. The search will be case insensitive unless the list contains HFS/ZFS files and is enclosed in apostrophes (') or quotation marks ("). In this case only, the search will be case sensitive.

`ALL`
: If none of the list lines are excluded or operand NX is specified, then FIND ALL is equivalent to FIND FIRST. If NX is not specified, then **all** excluded lines that match the search string are no longer excluded from the list display.

`FIRST`
: Search forwards from the first list entry to find the first list entry that contains a match for the string.

`LAST`
: Search backwards from the last list entry to find the last list entry that contains a match for the string.

`NEXT`
: Search forwards from the first list entry in the current display to find the next list entry containing a match for the string.

`PREV`

Search backwards from the first list entry in the current display to find the previous list entry containing a match for the string.

`EX`
`X`
: Search only excluded list entries.

`NX`
: Search only list entries that have not been excluded.

LOCATE Command

Syntax:

```
>>-- Locate ------ string ----------------------------------------------------><
```

Description:

Starting at the first entry and proceding downwards, LOCATE will compare *string* against data at the **start** of each field in the the first column of the display until either a match is found or the field data is greater than *string*.

If the strings are equal, then the display is scrolled so that this list entry becomes the first row in the display. Otherwise, if the list entry field data is greater than *string*, the display is scrolled so that the first row in the display is that which immediately precedes this list entry.

`LOCATE` *`string`* is appropriate only if the first list column is in ascending sort order and is only valid if the first column in the display is defined as being a character key field. (Check the FDB for Type=Char and Key=Yes).

Parameters:

`string`
: A character search string used in the compare against data at the start of fields in the first column of the display.

S Command

Syntax:

```
>>-- S ----------- member ----------------------------------------------------><
```

Description:

Supported as a CLI line command for Library Lists only, S *member* will perform the default operation (i.e. Edit) on the specified library member.

S is also supported as a List window prefix command which applies to all types of List window. In this case, S will execute the default operation for the particular list entry type. See the relevant prefix command table <Dflt> entry for for each of the supported Execute CBLVCAT, List and File Search windows.

Parameters:

`member`
: The library member name.

List Entry Display

A list window display area contains a line for each entry in the generated list.

In order to restrict the scope actions performed on list items, list entries may be excluded and re-included in the list display as required. For example, to delete a number of files whose names occupy consecutive lines in a list, you may enyer the line command "D" with "//" grouping markers. To omit deleting a file that falls between the pair of line markers, the list line may be excluded from the display, thus removing it from the scope of the delete operation.

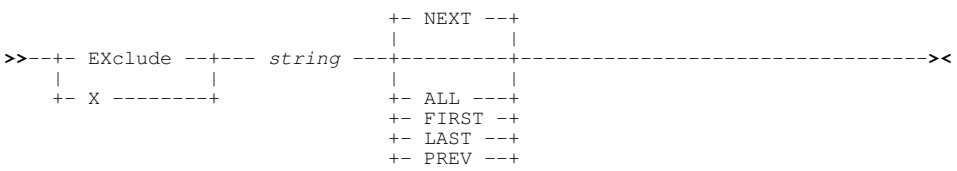
The following commands relate to excluding and then re-including entry lines in the list display.

- EXCLUDE Command
- FLIP Command
- HIDE Command

- MEMBER Command
- ONLY Command
- RESET Command
- SHADOW Command
- SRCHFOR Command

EXCLUDE Command

Syntax:



Description:

EXCLUDE is used to exclude display of a list entry that contains a match for the specified search string **anywhere** within the field occupying the first column of the display. Entries that are already excluded remain excluded following execution of EXCLUDE. Therefore, execution of successive EXCLUDE commands has a cumulative effect.

Note that a character logical compare is performed on the field's source data, not on the displayed value. Therefore, EXCLUDE may not be effective if the first column field has a source data type which is not character. (Check the FDB for Field Type=Char.)

Following a successful EXCLUDE operation, the excluded list entry (or entries) will be excluded from the display. If shadow line display is set on, then a single shadow line will be displayed in place of groups of consecutive excluded entries.

RESET EXCLUDED will redisplay excluded lines.

Parameters:

string
: A character search string used in the compare against data within the field occupying the first column of the display. The search will be case insensitive unless the list contains HFS/ZFS files and is enclosed in apostrophes (') or quotation marks ("). In this case only, the search will be case sensitive.

ALL
: Exclude **all** list line entries that match the search string.

FIRST
: Search forwards from the first list entry to exclude the first list entry that contains a match for the string.

LAST
: Search backwards from the last list entry to exclude the last list entry that contains a match for the string.

NEXT
: Search forwards from the first list entry in the current display to exclude the next list entry containing a match for the string.

PREV
: Search backwards from the first list entry in the current display to exclude the previous list entry containing a match for the string.

FLIP Command

Syntax:

Description:

FLIP will reverse the status of excluded and non-excluded lines in the list window so that excluded line entries are displayed and non-excluded line entries are excluded.

HIDE Command

Syntax:

```
>>-- HIDE ------------------------------------------------------------------><
```

Description:

HIDE will hide all shadow lines that represent groups of one or more excluded line entries in the list window. HIDE is equivalent to command SHADOW OFF.

MEMBER Command

Syntax:

```
>>-- MEMBER --- member_mask ---------------------------------------------><
```

Description:

Applicable only to lists with key columns containing data set names, MEMBER will exclude all list entries for data sets that are not libraries or do not contain at least one member whose name matches the specified member mask.

If shadow line display is set on, then a single shadow line will be displayed in place of groups of consecutive excluded entries.

Parameters:

member_mask
A member name mask which must match at least one member name within a listed library. The *member_mask* may include one or more of the following wild card symbols:

*	A single asterisk represents zero or more characters within a member name mask.
%	A single percent sign represents exactly one character within a member name mask. Up to 8 percent signs can be specified in each member name mask.

ONLY Command

Syntax:

```
                              +- NEXT --+
                              |         |
>>---- Only --------- string ---+---------+-----------------------------------><
                              |         |
                              +- ALL ---+
                              +- FIRST -+
                              +- LAST --+
                              +- PREV --+
```

Description:

ONLY will first exclude all list line entries and then include entries that contain a character match for the specified search string **anywhere** within the field occupying the first column of the display.

Note that a character logical compare is performed on the field's source data, not on the displayed value. Therefore, ONLY may not be effective if the first column field has a source data type which is not character. (Check the FDB for Field Type=Char.)

If shadow line display is set on, then a single shadow line will be displayed in place of groups of consecutive excluded entries.

Parameters:

string
A character search string used in the compare against data within the field occupying the first column of the display. The search will be case insensitive unless the list contains HFS/ZFS files and is enclosed in apostrophes (') or quotation marks ("). In this case only, the search will be case sensitive.

ALL
Exclude **all** list line entries that do not match the search string.

FIRST
Search forwards from the first list entry to exclude all but the first list entry that contains a match for the string.

LAST
: Search backwards from the last list entry to exclude all but the last list entry that contains a match for the string.

NEXT
: Search forwards from the first list entry in the current display to exclude all but the next list entry containing a match for the string.

PREV
: Search backwards from the first list entry in the current display to exclude all but the previous list entry containing a match for the string.

RESET Command

Syntax:

Description:

RESET will reset display of list entry lines.

Parameters:

EXCLUDED | X
: RESET EXCLUDED (or just RESET) will include all, currently excluded entries so that all entries are redisplayed.

HIDE
: RESET HIDE is equivalent to SHADOW ON and will re-display shadow lines that have been hidden by a HIDE or SHADOW OFF command.

SHADOW Command

Syntax:

Description:

SHADOW will control display of shadow lines that represent groups of consecutive, excluded entry lines in the list window.

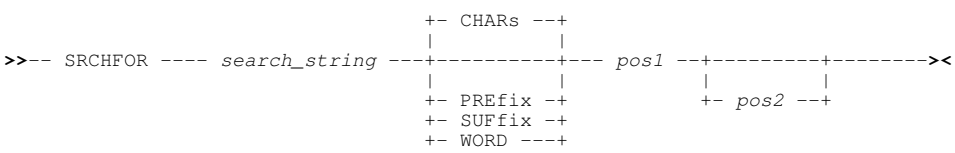
Parameters:

ON | OFF
: SHADOW OFF will remove shadow lines from the display and is equivalent to command HIDE.

  SHADOW ON will re-display shadow lines in the display and is equivalent to command RESET HIDE.

SRCHFOR Command

Syntax:



Description:

Applicable only to lists of data sets, library members or HFS/ZFS files, SRCHFOR will execute the File, Search & Update utility to perform an unformatted text search for the specified search string in all files identified by non-excluded entries in the list.

Following successful execution of SRCHFOR, the focus will be placed on the formatted report window generated by the utility.

Parameters:

search_string
: The search string used by the FIND operation performed on each of the files referenced in the list. The *search_string* format may be in any of the formats supported by the Data Editor FIND command. These are:

  ◊ An unquoted string with no commas or blanks. e.g. Book
  ◊ A quoted string for case-insensitive search. e.g. 'Book'
  ◊ A character string for case-sensitive search. e.g. C'Book'
  ◊ A hexadecimal string. e.g. X'C1D240'
  ◊ A picture string. e.g. P'B#o='
  ◊ A regular expression. e.g. R'B[aeiou]?@'

CHARS
: CHARS indicates that a successful match occurs if the search string is found anywhere within the text being searched.

PREFIX
: PREFIX indicates that a successful match only occurs if the search string is found at the start of a word within the text being searched. i.e. the matched text must precede an alphanumeric character and either be preceded by a non-alphanumeric character or be at the start of a text line.

SUFFIX
: SUFFIX indicates that a successful match only occurs if the search string is found at the end of a word within the data being searched. i.e. the matched text must be preceded by an alphanumeric character and either precede a non-alphanumeric character or be at the end of a text line.

WORD
: WORD indicates that a successful match only occurs if the search string is found to be a complete word within the data being searched. i.e. the matched text must either be preceded by a non-alphanumeric character or be at the start of a text line, and either precede a non-alphanumeric character or be at the end of a text line.

pos1
: The first position of a range of positions within the lines of text to be searched.

  pos1 must be a positive integer value (not zero) and must be a value that is less than or equal to the maximum length of the data records.

pos2
: The last position of a range of positions within the lines of text to be searched.

  Like *pos1*, *pos2* must be a positive integer value. If *pos1* references a position which is higher than that referenced by *pos2*, then the *pos1* and *pos2* values are swapped.

  If *pos2* is greater than the maximum record length then *pos2* is set equal to the maximum record length.

  Default is *pos1* plus the length of the search string minus 1.

List Window Prefix Area

List windows provide support for a prefix area which is displayed as a one to eight character enterable field occupying the first column of a list. This field is displayed with underscore characters to indicate that it is enterable.

A command entered in the prefix area of a list entry is actioned when a PFKey or <Enter> is hit. i.e. On a single 3270 I/O. (If configured, double-clicking the left mouse button will also action the command.)

Only one prefix command may be actioned against a list entry in a single 3270 I/O. However, multiple prefix commands may be actioned for multiple list entries in a single 3270 I/O. Where prefix commands are entered against multiple list entries, each command is executed in order from the top of the list to the bottom.

Supported prefix area commands depend on the individual list window application. A decription of supported prefix commands is documented in the help for each list window application.

A summary of all list prefix commands supported by parent windows of window class LISTFILE and VCIWEXEC, is found in Appendix B - List File Prefix Command Summary.

Default Action

Whether or not a list has a prefix area, list windows usually have a default action performed when a list entry is selected as if an item in a menu. i.e. When the cursor is positioned on a list entry and <Enter> is hit or, if configured, the left mouse button is double-clicked on the list entry. This is equivalent to executing prefix command "S" against a single list entry.

Help documentation for a list window applications that supports this default action includes a description of the default action as the <Dflt> entry in the application's table of supported prefix commands. e.g. CBLVCAT Interactive, List Dataset and File Search window prefix command tables.

Note that the default action is disabled if a prefix command is entered against any entry in the list.

Unrecognised Prefix Commands

Entering an unrecognised prefix command will return error ZZSW002E.

However, for file lists, where prefix command "E" (Edit) is supported, command, edit macro or procedure names may be entered in the prefix area. In this case, where the command entered in the prefix area is not one of the supported prefix commands, it is passed back through the window hierarchy in the following order. The command is processed by the first window or environment in which the command is recognised.

1. The CBLe text editor, if the list window is a CBLe List document window. (i.e. a CBLe synonym, command or macro name.)
2. The SELCOPYi environment command processor. (i.e. a SELCOPYi command.)
3. TSO, if SELCOPYi is started in ISPF or native TSO. (i.e. a TSO command, CLIST or EXEC.)

The full fileid of the entry against which the command is executed, is passed as the only input parameter to the command.

Repeating Prefix Commands

The last prefix command executed in the current list window may be repeated for any list entry by entering "=" in the prefix command area.

If other list entry prefix commands are executed in the same 3270 I/O before attempting to execute the command for this entry, then the command executed will be the last command executed for an entry in the list that occurs before this entry. Otherwise, the prefix command to be repeated is the last one executed on a previous 3270 I/O for the current list window.

Executing Prefix Command on a Block of List Entries

Any command entered in the prefix area may be repeated for each entry in a block of list entries marked by "//" in the prefix area of the first and last entries of the block. The command (or macro, etc.) to be executed must be specified on either the first or last entry of the block, following the "//" marker. e.g. "//d" in the prefix area of a file entry and "//" in any subsequent (or previous) file entry's prefix area defines the start and end of a group of file list entries that will be deleted.

Multiple, non-overlapping blocks of list entries may be marked, each block executing a specified command. If an unbalanced pair of "//" block markers is found or no command is specified on a block of list entries, execution of all blocked entry commands is delayed until there are and even number of block markers, each block having an accompanying command.

Executing Prefix Commands using a PFKey

PFKey assignment (e.g. via primary command KEYS) in a list window supports specification of a prefix command area target, denoted by a preceeding plus (+) symbol. Pressing the assigned PFkey on an individual list entry, executes the prefix command against that entry as if entered in the prefix command area.

In addition to the standard prefix commands defined for a list, any valid command or macro name of up to 8 characters may be specified. Where supported by the list, the list entry value will be passed as input to the command. e.g. Edit macro SELJ accepts an input JCL library member against which the SELCOPY Debug utility will be executed. Therefore, +SELJ may be assigned to a PFKey to act upon an entry in a library member list.

Interactive Panel Windows

Interactive panel (IPO) windows are of window class, WINWIPO0, and are used primarily for option, dialog and list windows.

SELCOPY/i functions that utilise interactive panel windows include:

- DB2
- File Copy
- File Compare
- File Search/Update/Copy/Remap
- Favorite Datasets/Commands
- SDE Edit

In addition to standard window features, windows of window class WINWIPO0 support a number of unique features that are described in this section.

- Panel Window Format
- Panel Window Size & Location
- Panel Window Hierarchy
- Panel Scrollable Display
- Panel Window Views
- Scrollable Input/Output Fields
- Input Field Data Recall
- Embedded Tables
- Selection Lists

Panel Window Format

In addition to the Command> and Scroll> prompts, an IPO window view may contain any of the following panel elements.

Panel ID

Every IPO window has a unique panel id which is displayed in a non-scrollable line of data at the top of the window display, below the window menu bar (if present).

Menu Bar

A menu bar may be displayed at the top of the panel display area below the window title bar. If the panel view includes an embedded list, then these menu items will be as described by the list window menu for the List window class.

Lines *n_first*-*n_last* of *n_total*

If a panel view does not include an embedded table or list, this information is displayed, right-adjusted, on the same non-scrollable line as the panel id. It identifies the line numbers of the first (*n_first*) and last (*n_last*) lines currently on display, and also the total number of lines (*n_total*) in the panel view.
e.g. "Lines 1-12 of 24"

This information is displayed because, by default, IPO panel windows are scrollable. The exception occurs where panel views contain an embedded table or list in which case the table/list entries are scrollable, not the panel display.

Input Fields

Input fields are enterable fields which allow the user to configure the functions performed by the panel. A value entered in a field must conform to that field's defined data type (e.g. numeric or text).

Where valid input field text can exceed the width of the input field area within the panel display, the input field may be scrolled and expanded as required. (See Scrollable Input/Output Fields)

Input Fields - Fixed Selectable Values

Text input fields may have been defined so that the specified input value can be only one of a pre-determined list of values.

If the text entered into an input field of this type does not exactly match one of these values, then the first value in the list that begins with the inputted text will be selected. If no match is found, then the input text is considered to be invalid and so a pop-up window is opened from which the user can select one of the valid input values from a list of possible values.

Input Fields - Variable Selectable Values

One or more interrelated input fields may require values which identify an existing object (e.g. Library DSN and Member) or values which relate to other information unique to the current operating environment.

For these types of input field, a selection may be made from a list of valid entries. Unlike input fields containing fixed selectable values, the list of possible input field values is generated at run-time and displayed as a modal panel window containing an embedded list. Panels of this type are called Selection Lists.

A selection list of valid entries is displayed if either of the following are true:

1. The input field value includes a wild card character '%' (percent) or '*' (asterisk), each representing zero or more characters.

2. Interrelated input fields represent a file object for which a PDS/PDSE library DSN has been specified without a member name.

The generated selection list entries are filtered based on values entered in the related input fields. Simply selecting a row from the list will close the list and populate the appropriate input fields with relevant values from the selected list entry.

If a value is entered (with or without a wild card character) that identifies a unique entry in the generated selection list, then display of the selection list window is bypassed and the input field(s) populated accordingly.

Input Fields Initial Values

When a panel is opened, input fields may contain values that have been determined using the following search sequence:

1. A field input value as specified by the user via a panel invocation primary command. The DB2 primary command opens a DB2 panel hierarchy and optionally populates panel input fields. e.g.
   ```
   DB2 7.4 CREATOR=ZZS
   ```
2. A default field input value as specified in the panel definition source. A default field input value is often defined to match default values in underlying syntax. e.g. DB2 default precision value for a DECIMAL field is 5.
3. If the field is configured to do so, the value displayed is that entered in the input field the last time the panel was opened. The value of each panel field is saved as a SELCOPYi INI variable when focus is removed from the panel window, and in the SELCOPYi User INI file when the SELCOPYi session is ended normally.

Output Fields

Output fields display variable data that is not part of the panel's static text and is included in the panel for information purposes only. These types of field are non-enterable and usually contain values that are generated as a result of previously supplied user options or input field entries.

Like input fields, output fields may be scrollable and expanded.

Option Check Boxes

An option check box is a single character input field which is either set on (checked) or off (unchecked). Any non-blank character may be entered in a check box to select that item, however, when the display is refreshed, the check box character will displauy as "/" (slash).

Comment text that accompanies a check box, describes the action taken when the box is checked. Further information may also be found in the panel help.

Radio Buttons

A radio button group is a collection of 2 or more mutually exclusive check boxes, one of which is always checked. When a radio button is checked, all other buttons in the radio button group become unchecked.

In the event that the user checks more than one radio button in the group before hitting <Enter> (or a PFKey), then the newly checked radio button that is positioned closest to the bottom right hand corner of the 3270 display, will be selected.

Options Menus

An option menu defines a list of numbered items from which the user may select a single item.

If other input fields exist in the panel view, an option menu usually has an accompanying option menu entry field in which the user enters a menu selection.

An item is selected using either of the following methods:

- ◊ Position the cursor on the required item and select <Enter> or, if configured, double-click the left mouse button on the item.
- ◊ If present, enter the menu item number in the option menu entry field provided, otherwise, enter the item number at the command line prompt.

Embedded Tables

IPO windows may include independently scrollable, embedded tables that have similar characteristics to the display of structured data in an SDE edit view. e.g. Prefix commands and PFKeys assignments that zoom, insert, delete and replicate table entries.

Embedded tables provide a method of supporting multiple, repeating groups of input/output fields. Unlike embedded lists, embedded tables may contain editable fields.

Embedded Lists

An IPO window may contain embedded lists based upon input parameters entered by the user. The list entries are not

generated until the user has completed input of, or accepted existing values for the parameter fields and hit <Enter>.

IPO windows that include embedded lists adopt characteristics (menu items, commands, etc.) as provided by list windows.

Panel Window Size & Location

Panel windows may be resized and repositioned using standard SELCOPYi window manipulation techniques. The last customised position and size of an IPO panel window is stored in the SELCOPYi User INI file and is restored to this position and size the next time the panel is opened.

The stored window size and position persists across subsequent SELCOPYi sessions.

Panel Window Hierarchy

An IPO panel window may be opened as a dependent window which is owned by another IPO panel. Furthermore, the process may be recursive so that an owned IPO panel window may itself be the owner of another IPO panel window. In the case of an expanded input field, the resulting text edit window view is owned by the IPO panel containing the input field.

Although it may still be in view, window focus cannot be placed on an owner IPO panel until the panel it owns is closed.

In addition to being owned, IPO panel windows containing embedded lists for selection of input field values are also modal.

Panel Scrollable Display

Unlike standard dialog windows, panel windows support up/down scrolling of data displayed in the window. Left/right scrolling is supported for scrollable Input/Output Fields, embedded lists and embedded tables only.

Up/down scrolling is particularly necessary where a window display area is not sufficiently large enough to display the entire contents of the panel (e.g. following a window resize or for 3270 terminals with a low number of displayed rows.)

For panel views that do not include an embedded table or list, the lines of the panel currently on display in the window display area, is reported in a non-scrollable line at the top of the panel. e.g. "Lines 11-36 of 36"

The "Scroll>" input field is used to control scrolling type as described by commands UP and DOWN. Scroll UP and DOWN are assigned to <PF7> and <PF8> respectively.

Unless positioned Scrolling performed on panel views that include an embedded table or list, will scroll the contents

Panel Window Views

A function associated with an IPO panel window may require a large amount of user input. If so, panel text and user input may either be displayed in a single, scrollable panel view, or split over a number of separate views within the same panel window, where each view contains related information.

Each panel window view may be considered an extension to user options and input fields presented in the first (primary) panel view.

Secondary panel window views are often displayed as a result of selections made in the primary panel view. The **BACK** command (assigned to <F3> by default) may be used to return focus to the previous view. If the focus view is the the primary panel view, then BACK will exit the panel.

Scrollable Input/Output Fields

Scrollable, expandable text input/output fields are supported allowing for input of more text than can be displayed in the field area.

These type of input/output fields are suffixed with a plus (+) and/or minus (-) symbol. The display of these symbols also provide an indication as to whether the field text in view is at the start ("+" only) or end ("-" only) of the field, or somewhere in between ("-/+").

To scroll the contents of the field, position the cursor within the field data and hit <F10> to scroll left and <F11> to scroll right.

The entire contents of the field may be expanded into a CBLe text edit view and so edited using the full functionality of the text editor, before being placed back in the input field when the text edit view is closed. To do this, position the cursor in the scrollable field and hit <F14> which, by default, is assigned to command EXPAND.

The expanded field text edit view displays 50 character portions of the text string on each line of the edit area (i.e. text wraps at column 50 to column 1 of the next line.) For input fields, any text entered beyond column 50 will be ignored.

The panel and field name associated with the input/output field and the maximum length of the text string is displayed before the edit display area. Text entered beyond the maximum length of the string will be truncated.

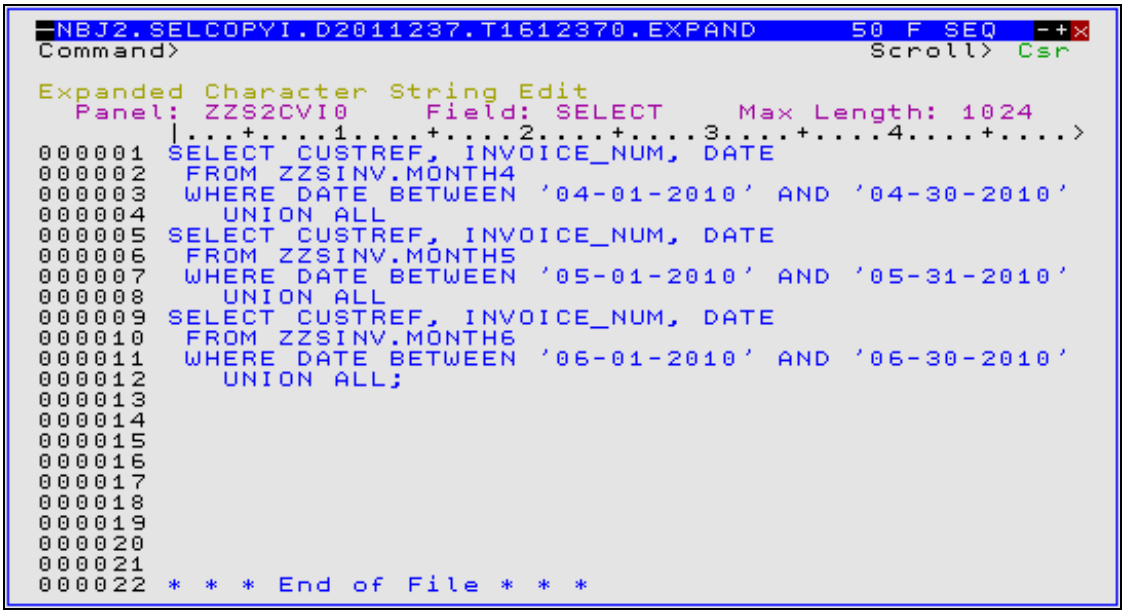


Figure 9. Expanded input field text edit view.

Input Field Data Recall

The last entry entered in a panel input field is stored in the user's SELCOPYi INI file when the panel is closed.

The next time that panel is opened, the contents of any input field may be restored to its last saved value by positioning the cursor within the input field and hitting <F16> which, by default, is assigned to command, REMIND.

Embedded Tables

Overview

IPO window embedded tables provide a method by which multiple, repeating groups of input fields may be inserted, updated, replicated or deleted. An embedded table may also contain non-editable (output) fields which are included in the table display for information only.

The contents of a table may be populated from other sources governed by panel input fields (e.g. SQL queries, SELCOPYi standard lists or other embedded tables.) Where a table contains rows of values to be used in generating the panel's function parameters, then populated table rows that are not required may be deselected simply by excluding them from the display.

Entries in a column of an embedded table may each represent rows of another embedded table. If so, each column field entry is coloured, by default, in yellow underscore with a ">" (greater than) prefix with a number of embedded table rows specification (e.g. `> 2 specified`.) Placing the cursor on a field entry of this type and pressing <Enter> will display an IPO sub-panel containing the embedded table represented by the parent table column field. On ending the sub-panel (<PF3>) the parent column entry will be updated to reflect the number of table rows specified in its child table.

Users should be aware that the order in which the table rows occur are the order in which parameters are generated for the panel's underlying command syntax. Although always syntactically correct, performance benefits may exist if parameters generated for table rows are specified in a particular order. e.g. Record key fields provided for Compare Files key synchronisation.

Table row entries may be edited using a set of command line (primary) commands, prefix (line) commands and pre-defined PFKey assignments that are analagous with structured data edit (SDE) commands and PFKey assignments.

Table Edit CLI (Primary) Commands

The following table contains command line interface (CLI) commands that may be executed by the user to edit tables embedded in IPO panels. The syntax specification for each command is as documented for the SDE command equivalents.

ALL	Synonym for WHERE.
BOTTOM	Display the last page of table rows. (Equivalent to DOWN MAX)
DELETE	Delete table rows. (Default is focus row)

DOWN	Scroll down through the display of table rows.
FLIP	Flip disply of excluded and non-excluded table rows.
INSERT	Insert table rows.
LEFT	Scroll left through the display of table columns.
LESS	Filter (exclude) additional table rows.
LOCATE	Locate table row based on *record_num* or *expression*.
MORE	Filter (include) additional table rows.
QUERY	Query current table edit options.
REDO	Redo changes to the table that have been undone.
REFRESH	Refresh the table contents.
RIGHT	Scroll right through the display of table columns.
SELECT	Select table columns and their order of display.
SELECTALL	Select/Deselect all table entries.
SET	Set table edit options.
SORT	Sort table column entries.
TOP	Display the first page of table rows. (Equivalent to UP MAX)
UNDO	Undo changes made to the table.
UP	Scroll up through the display of table rows.
WHERE	Filter table rows.
ZOOM	Zoom a table row to display it in single record view.

Table Edit Options

The following table contains available table edit SET and QUERY options.

COLHEADER	Display descriptive or internal format column header names. (See use of COLHEADER in Table Editing Techniques below).
COLOUR, COLOR	Colour specification for individual areas of the display.
COLWIDTH	Display width assignment for individual table columns.
MSGMODE	Controls display of messages.
MULTIPOINT	Controls support of >1 line label name assignment for any row.
PFKEY	Function Key assignment.
POINT	Line label name assignment.
PREFIX	Prefix area display settings.
REFERENCE	Table column field reference header line display.
SCALE	Table column scale header line display.
SHADOW	Display of shadow lines representing a group of excluded rows.

Table Edit Prefix (Line) Commands

.name	Set a line pointer (line label name).
A	Make this row the target for a move or copy (move or copy rows After this row).
B	Make this row the target for a move or copy (move or copy rows Before this row).
C(n) CC	Mark a row or a block of rows for copying. Rows may be copied to another position within the same table using prefix commands, A or B.
D(n) DD	Delete a row or a block of rows.
F(n) F*	Include the first *n* rows of an excluded row group. F* include all excluded rows.
I(n)	Insert a new row or a block of *n* new rows.
L(n) L*	Include the last *n* rows of an excluded row group. L* include all excluded rows.
M(n) MM	Mark a line or a block of lines for move. Lines may be moved to another position within the same table using prefix commands, A or B.
R(n) RR(n) "(n) ""(n)	Replicate (duplicate) a row or a block of rows *n* times.
X(n) X* XX	Exclude a row or a block of rows from the display. X* exclude all rows from the current row to the last row of the table.
Z	Switch to a zoomed (single record view) display of the row.

Table Edit PFKeys

The following PFKeys are assigned by default when the cursor is within the embedded table display area.

F1	HELP.
F2	SPLIT.
F3	BACK. (GO back to previous panel view.)
F4	WINDOW. (Navigate open windows.)
F5	SELECTALL. (Selects all visible rows.)
F6	SELECTALL DESELECT. (Deselects all visible rows.)
F7	UP. (Scroll up.)
F8	DOWN. (Scroll down)
F9	SWAP.
F10	LEFT. (Scroll left)
F11	RIGHT. (Scroll right)
F12	CRETRIEV.
F13	REMIND. (Reinstate previous value for focus entry field.)
F14	EXPAND. (Display focus entry field in a Text-Edit window.)
F15	END
F16	SHOWPOPUPMENU
F17	ZOOM. (Display a row in single view format.)
F21	SWAP LIST. (Display the ISPF swap list menu.)
F22	UNDO. (Undo a level of table edit changes.)
F23	REDO. (Redo a level of table edit changes undone by the last UNDO operation.)

Table Editing Techniques

The following hints and tips relate to tasks commonly performed on IPO panel embedded table fields

Single Row Display

Default display of an IPO embedded table is multi-row (table) format where **UP** (<PF7>) and **DOWN** (<PF8>) scrolls the table rows and **LEFT** (<PF10>) and **RIGHT** (<PF11>) scrolls the table columns.

All column names and values belonging to a single table row, may be displayed in the panle view simply by placing the cursor on a table row and pressing <PF2> or entering prefix command "Z" to execute **ZOOM** and so display that row in single row format. <PF2> again will return to the table display format. Column field values that are ediatble in multi-row format may also be edited in single row format.

Whilst in single row display format, use <PF10> and <PF11> to scroll backwards and forwards (i.e. LEFT and RIGHT) respectively through the table rows.

In some instances, the panel table will have been defined so that the zoomed display of the table fields is a formatted entry form containing explanitory text. This zoomed view of the table row is displayed in another panel view, in which case the END command (<PF3>) should be executed in order to return to the table view.

Filtering the Table Rows

A number of rows (n) may be manually excluded from display using the prefix (line) command "Xn" or, to mark a block of rows for exclusion, "XX".

WHERE, **MORE**, and **LESS** CLI (primary) commands may also be used to include/exclude multiple table row entries mechanically, based on the contents of any of the displayed fields. (**ALL** is a synonym for WHERE.)

These commands operate on structured data (SDE) expressions which support operators, character strings, numeric values, built-in functions, sub-expressions and *field values*.

Embedded table field values may be referenced in an expression via the column (field) reference number (e.g. #1, #4) or via the internally defined column (field) name.

By default, the column reference numbers and internally defined column names are not displayed. This is to maximise the amount of table data in view and to display more meaningful column headers provided by the column titles.

To display and then remove from display the field reference numbers, use command **SET REFERENCE ON/OFF** (abbreviated to **REF ON/OFF**). Similarly, to alternate the column header display between the column title an its internally defined name, use command **SET COLHEADER NAMES/TITLES** (abbreviation COLH N/T).

Examples:

```
LESS SelectTyp='AN'
```
Additionally exclude all entries where the value of table field name "SelectTyp" is "AN". Entries that were already excluded will be unaffected.

```
WHERE SelectFld >> 'ABC-' or #1 << 'DEF-'
```
Exclude all entries except those where the value of table field name "SelectFld" begins with literal "ABC-", or the value of field reference number 1 contains literal "DEF-".

Re-Ordering the Table Columns

By default, table columns are presented to the user in a logical order.

However, the user may suppress and/or change the order of columns displayed using the command **SELECT**, which specifies table columns by column (field) reference number or via the internally defined column (field) name. (See filtering techniques above for display of these column field attributes.

To redisplay columns in their default order, enter "SELECT *".

Adding/Inserting Table Rows

Table rows may be inserted using the INSERT (primary) command, "I(n)" prefix (line) command or by positioning the cursor on th erow befored the inserted row and pressing <PF1>.

Table rows may be replicated using the "R(n)" or "RR" prefix (line) commands or copied using prefix (line) commands "C(n)" or "CC" combined with "A" (after) or "B" (before).

The order in which table rows occur may be important. To re-order the table rows, use prefix (line) commands "M(n)" or "MM" combined with "A" (after) or "B" (before).

Undo/Redo Changes

Edited changes to tables (including field value updates, row inserts, deletions, etc.) may be undone and, if required, re-applied using the commands **UNDO** and **REDO** respectively.

Alternatively, the cursor may be positioned within the table area and then <PF22> or <PF23> pressed to perform UNDO or REDO respectively.

Undo levels are maintained for the table even if the panel view is changed and re-visited. However, on closing the panel the table rows are dropped from storage so that restarting the panel will reinitialise the table rows within the IPO panel view.

REFRESH Command

Syntax:

```
>>-- REFRESH ------------------------------------------------------------><
```

Description:

The REFRESH primary command will destroy then recreate the contents of table.

REFRESH will discard any table editing and row selections that have been performed. Furthermore, if the table is based on list information, SQL tables or other panel tables that have been updated since the table was created, then the refreshed table data will reflect these changes. e.g. A panel table based on the contents of a PDS library will reflect the members at the point the table is refreshed. Members that have been deleted will not appear in the refreshed table whereas members that have been renamed or added since the table was first created will be displayed accordingly.

SELECTALL Command

Syntax:

```
>>-- SELECTAll ---+--------------+-----------------------------------------><
                  |              |
                  +-- Deselect --+
```

Description:

For tables that contain a selection column, SELECTALL may be used to select or deselect **all** the table row entries.

Tables that include a selection column require a row to be selected to include it in the function performed by the panel to which it belongs. SELECTALL is a convenient method of selecting all rows or deselecting all previously selected rows.

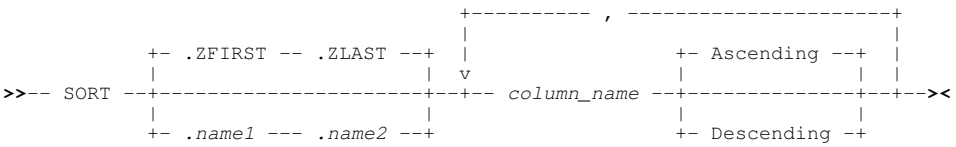
For panels that contain embedded tables, SELECTALL is usually assigned to <F5> and SELECTALL DESELECT assigned to <F6> by default.

Parameters:

`DESELECT`
: Indicates that all rows are to be deselected. Default is to select all rows.

SORT Command

Syntax:



Description:

The SORT primary command may be used to sort table rows by data contained in one or more named columns.

Sorting may be restricted to operate on a range of rows that fall between two labelled rows. If no range of rows is specified, an implicit range of .ZFIRST .ZLAST is used (i.e. all rows of text in the edit view.) If only the first line of a range of rows is specified (*.name1*) then the default last line is .ZLAST.

Table rows may also be sorted using the cursor simply by placing the cursor on the heading of the column on which you want to sort and then pressing the <Enter> key. Alternatively, if configured, simply double-click the left mouse button on the list column header.

Subsequent sorting on a column header using this method will reverse the order in which data in that column was last sorted.

Parameters:

`.name1`
: A label name identifying the first row in the range table rows to be included in the sort operation. The preceding "." (dot) in *.name1* is mandatory.

`.name2`
: A label name identifying the last row in the range table rows to be included in the sort operation. The preceding "." (dot) in *.name1* is mandatory.

  If *.name2* references a table row number which is lower than that referenced by *.name1*, then the order is reversed to define a positive number of rows. Default for *.name2* is .ZLAST.

`column_name`
: Name of a table column on which to sort.

  Note that the column title is not acceptable. To temporarily display column names, use the SET COLHEADER NAME option.

`ASCENDING | DESCENDING`
: The sort order is specified using a list of column sort specifications which consist of a column name followed by a sort direction. The sort direction is given as A for ascending or D for descending.

  If the sort direction is not provided, it defaults to ascending.

  Commas must be used to separate multiple column sort specifications whereupon intervening blanks are permitted.

VLMAX Command

Syntax:

Description:

The VLMAX primary command sets the display width of all variable length columns to their maximum widths.

VLMIN Command

Syntax:

```
>>-- VLMIN ------------------------------------------------------------------><
```

Description:

The VLMIN primary command sets the display width of all variable length columns to the minimum width required to view the current table data without truncation. i.e. a width equal to the longest entry in the column.

CCOLOUR - SET/QUERY Option

Syntax:

```
>>--+-------+----- CCOlour --- column_name --+--------------+----------------->
    |       |                                |              |
    +- SET -+                                +- ccolour_id -+
                         +- NONe ------+
                         |             |
 >--+--+- Blue ------+--+-------------+--+- WHen --+- expression -------+---><
    |  |             |  |             |  |         |                    |
    |  +- Red -------+  +- BLInk -----+  +- WHere -+                    |
    |  |             |  |             |                                 |
    |  +- Pink ------+  +- REVvideo --+                                 |
    |  |             |  |             |                                 |
    |  +- Green -----+  +- Uscore ----+                                 |
    |  |             |                                                 |
    |  +- Turquoise -+                                                 |
    |  |             |                                                 |
    |  +- Yellow ----+                                                 |
    |  |             |                                                 |
    |  +- White -----+                                                 |
    |  |             |                                                 |
    |  +- Default ---+                                                 |
    |                                                                  |
    +-- OFF -----------------------------------------------------------+

>>--- Query ------ CColour -+---------------+--------------------------------><
                            |               |
                            +- column_name -+
```

Description:

This option defines a column colouring definition. Colour highlighting of individual values within a particular column may be applied based on a true result from a supplied SDE expression.

Multiple column colouring definitions may be specified for each column within the table.

SET Value:

column_name
: Name of the table column on which the values will be coloured. See COLHEADER option to display table column names instead of column titles.

ccolour_id
: An arbitrary alpha-numeric identifier to be assigned to the current column colouring definition.

  If *ccolour_id* specifies the colour id of an existing column colouring definition, the definition will be replaced.

BLUE | RED | PINK | GREEN | TURQUOISE | YELLOW | WHITE | DEFAULT
: Supported colours. If DEFAULT is specified, the default colour for column data is set.

BLINK | REVERSE | USCORE | NONE
: Extended highlighting options. The coloured data may blink, be displayed in reverse video or be underlined. Default is NONE for no extended highlighting.

WHERE | WHEN *expression*
: Any valid SDE expression. If the result of this expression is non-zero, the specified colour highlighting will be applied to the column entry.

OFF
: Deletes the column colouring definition identified by *ccolour_id*.

QUERY Response:

For each column colour definition, QUERY CCOLOUR displays a message containing the *ccolour_id*, colour, extended highlighting option and SDE expression that comprise the definition. If *column_name* is specified, only the colour attributes for that table column are displayed.

COLHEADER - SET/QUERY Option

Syntax:

```
>>-+------------+-- COLHeader --------+-- Name ---+-----------------------------><
   |            |                     |           |
   +- SET ------+                     +-- Title --+


>>--- Query ------ COLHeader ------------------------------------------------><
```

Description:

This option controls the display format of the table column names header line.

By default, column are displayed with their descriptive column titles. However, when referencing columns on a SELECT command or in an expression for WHERE, MORE or LESS row filtering operations, then the column's internal format name or its column field reference id must be used. For this purpose, SET COLH NAME may be used to display the columns with their internal name format.

SET Value:

`NAME | TITLE`
: Display all table columns with their internal column name format (NAME) or with their descriptive column title (TITLE).

QUERY Response:

The column header names format (NAME or TITLE).

Selection Lists

Selection lists provide facility to select required input field values from a list of available options.

A list of selectable entries is displayed as an embedded list within a new, modal panel window. The entries themselves are generated from appropriate system objects (e.g. Catalog/VTOC data sets) and are filtered so that only those entries that satisfy a provided mask string that includes wild card characters.

Since the panel contains an embedded list, the panel adopts characteristics as provided by list windows . i.e. Display of list window menu bar items, the ability to select and sort columns, filter list rows and locate entries.

To select an entry, position the cursor on the required list row entry and press the <Enter> key or, if configured, double-click the left mouse button on the required row entry.

The selection list panel window title includes a short description usually identifying the parent utility panel and the input field to which the selected entry will apply. e.g. (Create Filter) Structure File. By default, the panel field entries are populated with values that were entered in the associated input fields belonging to the parent panel.

The following selection lists relate specifically to input fields that reference file objects and may be invoked from many SELCOPYi utility panels. (e.g. Compare Files, Create Filter.)

Select Dataset Name (Catalog)

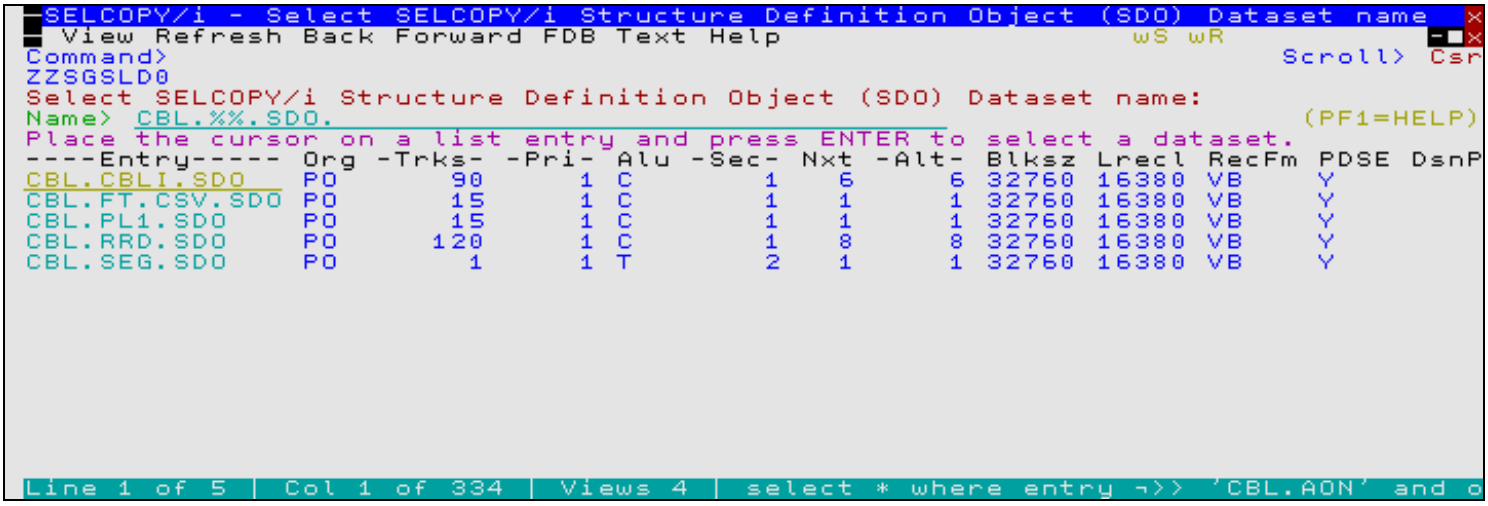


Figure 10. SELCOPYi - Select Dataset Name (Catalog List).

The Select (Cataloged) Dataset Name panel selection list is displayed when a wild card '%' (percent) or '*' (asterisk), both representing zero or more characters, is specified in a panel DSN input field and no volume id has been specified in the related file Volume input field.

Name>
Input field in which a new DSN mask may be entered in order to regenerate the selection list.
The DSN mask string may contain one or more of the wild card characters "%" (percent) or "*" (asterisk), each representing zero or more characters.

Extended Details
Enter / to display extended details from the VTOC, in addition to catalog information for each entry.

Select Dataset Name (VTOC)

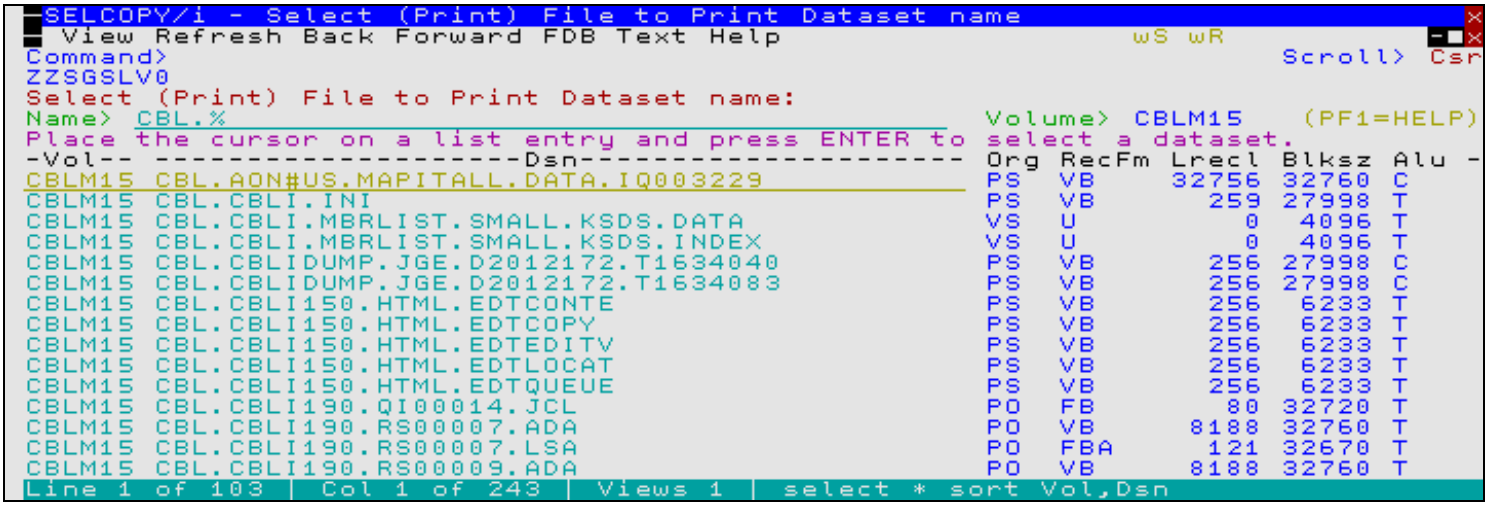


Figure 11. SELCOPYi - Select Dataset Name (VTOC List).

The Select Dataset Name panel selection list which contains only entries for a particular volume (VTOC) is displayed when a wild card '%' (percent) or '*' (asterisk), both representing zero or more characters, is specified in a panel DSN input field and a volume id is also specified in the related file Volume input field.

Name>
Input field in which a new DSN mask may be entered in order to regenerate the selection list.
The DSN mask string may contain one or more of the wild card characters "%" (percent) or "*" (asterisk), each representing zero or more characters.

Volume>
Output field displaying the volume id of the DASD volume VTOC used to display entries that match the DSN mask. This field is non-enterable.

Select HFS Path Name

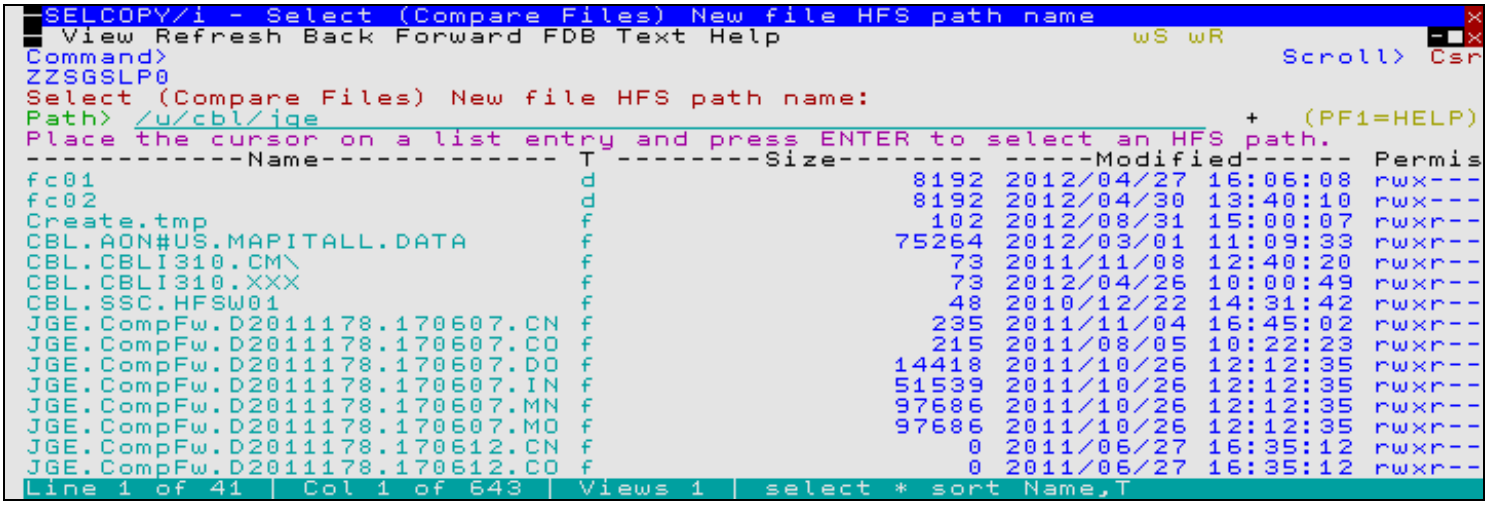


Figure 12. SELCOPYi - Select HFS Path Name (List).

The Select HFS Path Name panel selection list is displayed when a wild card '%' (percent) or '*' (asterisk), both representing zero or more characters, is specified in a panel DSN/Path input field and other elements of the input string value invalidate it as being a DSN (e.g. presence of invalid DSN characters and/or qualifiers of length > 8 characters.)

Path>

Input field in which a new HFS file path mask may be entered in order to regenerate the selection list.
The HFS file path mask string may contain one or more of the wild card characters "%" (percent) or "*" (asterisk), each representing zero or more characters.

Select Member Name

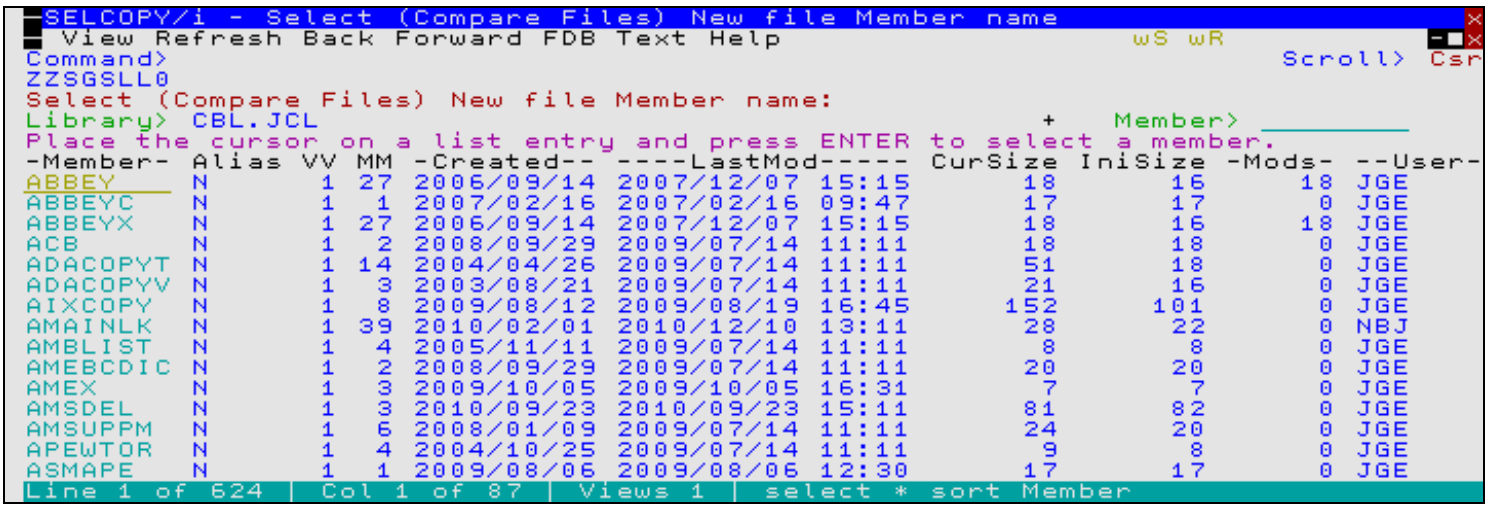


Figure 13. SELCOPYi - Select Member Name (List).

The Select Member Name panel selection list is displayed when a wild card '%' (percent) or '*' (asterisk), both representing zero or more characters, is specified in a panel Member input field or a PDS/PDSE library DSN is specified in the related file DSN input field and the panel Member input field is blank.

Library>

Output field displaying the PDS/PDSE library DSN from which the member list has been generated. This field is non-enterable.

Member>

Input field in which a new member mask may be entered in order to regenerate the selection list.
The member mask string may contain one or more of the wild card characters "%" (percent), representing a single character, or "*" (asterisk), representing zero or more characters.

Primary Option Menu (=)

Primary Option Menu Panel

The Primary Option Menu panel (ZZSGPRIM) is an interactive panel window providing an entry point to all SELCOPYi panels and functions.

This panel may be started by enetering "=" (equals) at any command line. A fast path may be specified immediately following the "=" symbol to directly open sub-panels of the Primary Option menu. (e.g. =0.4.1 for "COBOL Compiler options.")

Menu Bar Items

`File`
The File drop-down menu contains the single item, Exit, to close the panel.

`SwapList`
If SELCOPYi is operating within an ISPF split screen, opens the ISPF task list of active ISPF logical sessions.

`Window`
Opens the Window List window containing a selectable list of all open windows in the SELCOPYi session.

`Help`
Open the general help for the Primary Option menu panel.

`QuickRef`
Open the SELCOPYi Quick Reference.

Options

0 Settings	Set SELCOPYi options
1 Text Edit	Edit/View small text-type files
2 Data Edit	Edit/Browse potentially large data files
3 List	List Volumes,VTOCs,Datasets,Members etc
4 Home	Edit and execute point-and-shoot commands
5 Copy/Reformat	File Copy with optional copybook reformat
6 Search/Update	File Search/Update/Copy/Reformat
7 Compare	File/Library Compare Utilities
8 Utilities	General utilities
9 Structure	Create structure from copybooks etc
10 Filter	Create record selection filter
11 Print	Print Dataset (Batch)
12 DB2	Work with DB2, browse/edit tables etc
13 SMF	Work with MVS System Managed Facilities data
T **Training**	Setup SELCOPYi Training Material
WL Window List	Display active windows, select to switch focus
X **Exit**	Exit SELCOPYi

Panel Output Fields

`User:`
An output field displaying the user's logon id.

`Version:`
An output field displaying the version of SELCOPYi.

`Date:`
An output field displaying the current date.

`Time:`
An output field displaying the current time.

`OpSys:`
An output field displaying the operating system release.

`System:`
An output field displaying the z/OS system name as defined by the SYSDEF statement in active parmlib member IEASYMxx.

`VM User:`
For VM guest operating systems or CMS users, this field displays the VM userid of the guest machine or CMS system.

Settings (=0)

Settings Panel

The Settings panel (ZZSGSET0) is an interactive panel window, opened on selection of option 0. in the SELCOPYi Primary option menu.

This panel, and its sub-panels, establish default options and values applicable to the individual user's SELCOPYi environment. These values are assigned to variables set in the user's own SELCOPYi User INI file.

Options

Individual option entries relate to specific functions of the SELCOPYi environment.

1 Startup	Startup options
2 System	System options
3 Text Edit	Text Editor (CBLe) options
4 Data Edit	Structured Data Editor (SDE) options
5 List	List window options
6 Batch	JCL Information for generated Batch Jobs
7 DB2	DB2 options
8 Function Keys	Maintain Personal and Installation-wide PF Key Settings
9 Search/Update	Set Search/Update (FSU) utility report options
10 SMF	Set MVS System Managed Facilities options

Panel Input/Output Fields

`REXX Macro Library Path:`
Fields that together establish the library search chain used to locate a SELCOPYi REXX macro.

`User Library>`
Normally set to the default *userid*.**SELCOPYi.CBLE,** this input field allows the user to enter the fully qualified name of one or more PDS/PDSE SELCOPYi REXX macro libraries to appear first in the macro path search chain.

This field may be expanded (using <F2>) in order to enter any number of blank delimited library data set names in the order in which they are to appear in the search chain. Close the expanded display (<F3>) and press <ENTER> to refresh the count of user libraries (#1 of n).

Note that, in any single update of the User Library field, if a non-existant library name is specified, then message ZZSE062E "Invalid macro path" is returned, the update is not applied and the original User Library field value is reset to its state prior to attempting the update.

`Site Library>`
A non-enterable (output) field identifying the DSN of the penultimate library in the macro search path. This library contains macros that have been developed at the client's installation and made available to all SELCOPYi users.

`CBL Supplied Library>`
A non-enterable (output) field identifying the DSN of the last library in the macro search path. This library (hlq.SZZSDIST.CBLE) contains macros that have been distributed by CBL as part of the SELCOPY Product Suite and contains macros available to all SELCOPYi users.

Startup Settings (=0.1)

The Startup Settings panel (ZZSGSET1) is an interactive panel window, opened on selection of option 1. in the SELCOPY/i Settings panel.

This panel specifies which application windows are to be opened at startup of SELCOPY/i.

Panel Input Fields

`Primary Option Menu`
Select this option to automatically open the SELCOPY/i Primary Option menu at startup.

`Menu Exit`
Select this option to fully exit SELCOPY/i (thereby bypassing the SELCOPY/i "logo" screen and saving valuable keystrokes) if PF3 is pressed at the Primary Option Menu. This option is obeyed only if the option to automatically open the Primary Option Menu at startup is also selected.

Home File
Select this option to automatically open the the user's HOME command centre (CMX) file at startup.

System Settings (=0.2)

The System Settings panel (ZZSGSET2) is an interactive panel window, opened on selection of option 2. in the SELCOPYi Settings panel.

This panel specifies options relating to SELCOPYi general operation.

Panel Input Fields

Command Line>
Specifies the location of the command line in all SELCOPYi windows to the TOP or BOTTOM of the display.

Command Delim>
Identifies the single character interpreted as the command separator used to enter multiple commands from a single command prompt.

Act-bar Tab>
Specifies whether the keyboard Tab key will position the cursor at the menu bar items displayed on the 3270 screen.

Popup Tab>
Specifies whether the keyboard Tab key will position the cursor at popup or dropdown menu items displayed in a menu window on the 3270 screen.

Abend Trap>
Specifies whether the SELCOPYi abend trap is activated to trap any SELCOPYi internal system abends and, if possible, recover from the abend.

If set ON, the abend will be trapped and a formatted dummp written to a SELCOPYi dump data set. This dump data set may be requested by CBL for diagnostic purposes.

Dump Prefix>
Specifies the data set name prefix (maximum length 26) to be used for a generated SELCOPYi formatted dump data set.

Qualifiers of the form '.Dyyyyddd.Thhmmssx', representing the current date and local time, are appended to the dump data set name prefix qualifiers.

Text Edit Settings (=0.3)

The Text Edit Settings panel (ZZSGSET3) is an interactive panel window, opened on selection of option 3. in the SELCOPYi Settings panel.

This panel specifies options relating to the SELCOPYi text editor.

Panel Input Fields

Interface>
Specifies the default edit interface.

The SELCOPYi text editor supports edit commands supported by both the ISPF editor and the CMS XEDIT/Windows KEDIT editors. Some command verbs exist for both editors but can have very different effects (e.g. CHANGE). The prevailing SELCOPYi text edit interface dictates the precedence by which common command verbs are interpreted and also influences the screen display and scrolling.

Size Warning>
Specifies the file size threshold at which the SELCOPYi text editor will warn the user that it is about to load all records of a large file into storage. The message also prompts the user to either continue with the load or switch to using structured data edit which supports edit without all records being loaded in storage.

This value may be specified as a number of bytes (nnn), kilobytes (nnnK) or megabytes (nnnM).

Load Warning>
Specifies the file load warning threshold. During load of a file for text edit, when the number of bytes loaded reaches a factor of this load warning threshold, then a message is displayed prompting the user to continue or cancel the file load.

This value may be specified as a number of bytes (nnn), kilobytes (nnnK) or megabytes (nnnM).

ISR Macros>
Specifies whether ISPF Rexx Edit macros support is activated for SELCOPYi text edit. Setting this option to "YES" has the following effect:

◊ ISREDIT (ISPF Edit) macro commands and assignment statements are intepreted by the SELCOPYi text editer.

◊ Following search of the SELCOPYi macro path libraries, libraries in the SYSUEXEC and SYSEXEC concatenations are searched for a member name that matches any unrecognised command passed to the SELCOPYi text editer.

See ISPF Edit Macros for details.

Action Comments>
Specifies the 1-4 character string that identifies the start of comment data in a line of text processed by the ACTION facility. If set to a non-blank string, that string is excluded from the command and is treated as a comment delimiter.

Action Multiple>
Controls the significance of the '|' (OR symbol) as a special character in a line of text processed by the ACTION facility. If set on, '|' is excluded from the command string and is treated as a command string delimiter.

Action Place Cursor>
Controls the significance of the first '_' (underscore) as a special character in a line of text processed by the ACTION facility. If set on, the first '_' is excluded from the command string and identifies the location at which the cursor is to be positioned if the command string is placed at the command prompt.

Structured Data Edit (SDE) Settings (=0.4)

The Structured Data Edit (SDE) Settings panel (ZZSGSET4) is an interactive panel window, opened on selection of option 4. in the SELCOPYi Settings panel.

This panel specifies options relating to the SELCOPYi Structured Data (SDE) editor.

Options

1 COBOL	COBOL Compiler and Replacing Options
2 PL/1	PL/1 Compiler Options
3 HLASM	HLASM Assembler Options
4 Aux	Auxiliary Dataset Options
5 Work-File	Compiler Work-file Allocation Settings
6 Associations	Manage the association of copybook mappings with data files

Panel Input Fields

Load Warning>
Specifies the number of records loaded warning threshold. During load of a file for structured data edit, when the number of records loaded reaches a factor of this load warning threshold, then a message is displayed prompting the user to continue or stop the file load. If the load is stopped, then Update-in-place Edit is used.

This value may is specified as a number of records (nnn).

Max Storage>
Specifies the maximum storage available for SDE edit of a single data set.

An SDE edited data set is limited by the lesser of the prevailing MAXSTOR value and the amount of free private area storage above the 16MB line available within the region at the time of open. This limit is used to determine the SDE edit technique and data record management used to edit the data set.

This value may be specified as a number of bytes (nnn), number of kilobytes (nnnK) or a number of megabytes (nnnM). A value of 0 (zero) implies no maximum storage is to be applied.

Max Window Size> rows x columns
On initial display of an SDE data edit window, its size will be restricted to the specified number of display rows and columns.

Once the window is displayed, it may be resized to encompass more rows and/or columns if required.

Auto Rec-Types>
With this option set on, then if there is only one level one data element and all the level 2 data elements are groups that form a union (i.e. the second and subsequent level 2 data elements all redefine the first), then the level 2 data elements will be treated as defining the record types.

Data Names Case>
Set this option as "UPPER" to automatically upper case any data names (record-types, field names) obtained from a COBOL or PL1 copybook in order to define a record layout.

Set this option as "MIXED" to maintain the case as defined by the copybook.

This option affect the appearance of these data names on screen while in browse/edit of a data file using the layout in question. Commands that refer to these data names (e.g. SELECT, WHERE, FIND etc) are case insensitive irrespective of this setting.

Exclusion Level>
This option affects browse/edit using a segmented record layout only, and modifies the handling of each segment's "exclusion" status.

Segments may be excluded/re-included during your browse/edit session using

- ◊ the "FILTER" dialog,
- ◊ line-commands such as "X", "F" and "L",
- ◊ primary commands such as "EXCLUDE" (X), "FIND" (F), "ALL" (WHERE), "MORE", "LESS" and "RESET".

Set this option as "LOGICAL" to control exclusions at the individual segment level.

Set this option as "PHYSICAL" to force exclusion of any individual segment to affect all segments (the primary and all secondaries) within that physical record.

Hex Punctuation>
This option controls whether or not data in formatted record fields displayed as printable hexadecimal digits contain "," (comma) and blank character punctuation.

The separate HEXView primary command may be used to force a field to be displayed as printable hex, overriding its default format (as dictated by the field's defined data type).

Note that 2 printable hexadecimal digits are displayed for each byte of data within a field displayed as printable hex. If this option is set to DEFAULT, the hex display will have a "," (comma) inserted at each half-word (2-byte) boundary and a blank character at each full-word (4-byte) boundary.

Printable hex display with option set to DEFAULT:

```
<---,---- ----,---- ----,---- ----,---- 1---,---- ----,---- ----,---->
D9A4,9496 A499,40C8 81A2,40C9 A340,4040 4040,4040 4040,4040 4040,4040
E3A4,9995 8995,8740 E381,8293 85A2,4040 4040,4040 4040,4040 4040,4040
C496,957D A340,E896 A440,D985 9485,9482 8599,4040 4040,4040 4040,4040
```

Printable hex display with option set to NONE:

```
<---+----1----+----2----+----3----+----4----+----5----+----6--->
D9A49496A49940C881A240C9A3404040404040404040404040404040404040
E3A4999589958740E381829385A2404040404040404040404040404040404040
C496957DA340E896A440D9859485948285994040404040404040404040404040
```

See SET HEXPUNCTUATION for more info.

Find/Change/eXclude Scope>
Data-Edit's FIND, CHANGE, EXCLUDE and ONLY commands each support specification of **ANY** or **FOCUS** to control the scope of lines affected.

This option controls the default used if ANY/FOCUS are not explicitly stated.

If **ANY** is selected then all record-types are processed (included in the search) provided they are "visible". See VIEW , VBASE , and V/V+/V- line-commands for more information.

If **FOCUS** is selected then only records/segments assigned the default record type (visible and EXCLUDED records) are processed (included in the search).

Earlier versions of SELCOPYi did not support the **ANY** option (which is now the supplied default) at all. User's wishing to reinstate the previous default should select **FOCUS** for this panel option.

COBOL Compiler Options (=0.4.1)

The COBOL Compiler Options panel (ZZSGSETC) is an interactive panel window, opened on selection of option 1. in the Structured Data Edit (SDE) Settings panel.

This panel specifies options relating to the SELCOPYi Structured Data (SDE) COBOL copybook support.

Panel Input Fields

Copybook Interpreter>
A value of "INTERNAL" indicates that SELCOPYi's own internal processing will interpret any COBOL copybooks in order to generate an internal SDE structure (SDO).

A value of "EXTENAL" indicates that the COBOL compiler (specified below) will interpret any COBOL copybooks in order to generate an internal SDE structure (SDO).

SELCOPYi's internal COBOL compiler will normally run much quicker and require much less storage than an external compiler. e.g. Your external COBOL compiler may not run in a standard 32MB TSO region.

EXTERNAL Compiler>
Specifies the full DSN and member name of the external COBOL Compiler module (e.g. IGY330.SIGYCOMP(IGYCRCTL).)

If "Copybook Interpreter" is set to "EXTERNAL", then SELCOPYi will invoke the COBOL compiler when generating an internal SDE structure (SDO) from a COBOL Copybook.

Specification of a COBOL Compiler is necessary only if your COBOL compiler program module is not named IGYCRCTL and is not found in the library search chain.

COBOL Max RC>
Specifies the maximum acceptable COBOL compiler return code for which SELCOPYi will continue to generate an SDE structure (SDO).

Where the COBOL compiler return code is greater than this value, the SDE create structure operation fails with an error message.

COBOL Replacing Options:
References 12 pairs of ('From:' and 'to:') fields that together generate a COBOL REPLACE statement which is inserted in the temporary source mamber used by SELCOPYi as input to the COBOL compiler. This REPLACE statement is applied to **all** copy books selected for compilation.

The 'From:' field specifies a *pseudo-text* source string to be replaced. The corresponding 'to:' field specifies a *pseudo-text* replacement string.

PL/1 Compiler Options (=0.4.2)

The PL/1 Compiler Options panel (ZZSGSETP) is an interactive panel window, opened on selection of option 2. in the Structured Data Edit (SDE) Settings panel.

This panel specifies options relating to the SELCOPY/i Structured Data (SDE) PL/1 copybook support.

Panel Input Fields

Copybook Interpreter>
A value of "INTERNAL" indicates that SELCOPYi's own internal processing will interpret any PL1 copybooks in order to generate an internal SDE structure (SDO).

A value of "EXTENAL" indicates that the PL1 compiler (specified below) will interpret any PL1 copybooks in order to generate an internal SDE structure (SDO).

SELCOPYi's internal PL1 compiler will normally run much quicker and require much less storage than an external compiler. The external PL1 compiler will not run in a standard 32MB TSO region.

EXTERNAL Compiler>
Specifies the full DSN and member name of the PL1 Compiler module (e.g. IEL330.SIBMZCMP(IBMZPLI).) SELCOPY/i will invoke the PL1 compiler when generating an internal SDE structure (SDO) from a PL1 Copybook.

Specification of a PL1 Compiler is necessary only if your PL1 compiler program module is not named IBMZPLI and is not found in the library search chain.

PL/1 Max RC>
Specifies the maximum acceptable PL1 compiler return code for which SELCOPY/i will continue to generate an SDE structure (SDO).

Where the PL1 compiler return code is greater than this value, the SDE create structure operation fails with an error message.

PL/1 Compiler Additional Options:
Additional compiler options to be added using a "*PROCESS" statement.

HLASM Assembly Options (=0.4.3)

The HLASM Assembly Options panel (ZZSGSETA) is an interactive panel window, opened on selection of option 3. in the Structured Data Edit (SDE) Settings panel.

This panel specifies options relating to the SELCOPY/i Structured Data (SDE) HLASM DSECT support.

Panel Input Fields

HLASM Assembler>
Specifies the full DSN and member name of the HLASM assembly module (e.g. HLA.SASMMOD1(ASMA90).) SELCOPY/i will invoke the HLASM assembly module when generating an internal SDE structure (SDO) from an HLASM DSECT.

Specification of an HLASM assembly module is necessary only if your HLASM assembler program module is not named ASMA90 and is not found in the library search chain.

HLASM Max RC>
Specifies the maximum acceptable HLASM assembly return code for which SELCOPY/i will continue to generate an SDE structure (SDO).

Where the HLASM assembly return code is greater than this value, the SDE create structure operation fails with an error message.

Auxiliary Dataset Options (=0.4.4)

The Auxiliary Dataset Allocation panel (ZZSGSETX) is an interactive panel window, opened on selection of option 4. in the Structured Data Edit (SDE) Settings panel.

This panel allows the user to configure data set options that are subsequently used to allocate a temporary, auxiliary data set when the Auxiliary Edit technique is required. Auxiliary edit occurs when editing a non-KSDS data set that is larger than the maximum storage value or the calculated amount of free private area storage above the 16MB line. Note that an auxiliary data set is a RECFM=VB physical sequential data set.

Options in this panel should be customised so that auxiliary data sets comply with your system standards.

Panel Input Fields

Aux Dataset HLQ>
Specifies the dataset name prefix (maximum length 26 characters) to be used by SELCOPYi SDE when allocating an Auxiliary Edit data set.

Qualifiers of the form '.Dyyyyddd.Thhmmssx', representing the current date and local time, are appended to these data set name prefix qualifiers.

Default value is %USER%.SELCOPYI.SDEAUX.

Aux Dataset Unit>
Specifies the UNIT device number, device type or esoteric group name.
Note that no UNIT parameter is required if the auxiliary data set is SMS managed. Specify a STORCLAS or let an automated class selection (ACS) routine select a storage class for the data set.

SMS Classes:
Fields relating to SMS data set management.

Data Class>
SMS Data Class to be used. Specify a Data Class if one is not automatically selected via an ACS routine.

Storage Class>
SMS Storage Class to be used. Specify a Storage Class if one is not automatically selected via an ACS routine.

Management Class>
SMS Management Class to be used. Specify a Management Class if one is not automatically selected via an ACS routine.

Compiler work file allocation settings (=0.4.5)

The Compiler work-file allocation settings panel (ZZSGSETW) is an interactive panel window, opened on selection of option 5. in the Structured Data Edit (SDE) Settings panel.

This panel allows the user to adjust the space allocated for the work files used by the High Level Assembler (HLASM), COBOL or PL1 compilers when invoked by SELCOPYi in order to generate a native SDE structure (SDO) from a copybook.

See also COMPILERDDSIZE - SET/QUERY/EXTRACT Option.

Manage Copybook Associations (=0.4.6)

The Structure Associations Settings panel (ZZSASSC) is an interactive panel window, which may opened by any of the following:

1. Selecting option 6 from the Structured Data Edit (SDE) Settings panel.
2. Selecting option 4 from the Structure item from the Primary Options menu.
3. Typing primary command STRUCTURE (STRUCT) at any command line.

This panel controls options relating to the automatic association of structures (copybook mappings) with data files when the edit or browse features are used and no structure is specified.

The relationships are maintained in a table which may be manually edited. In this case generic associations may be defined by specifying wildcards in a data file name mask.

Panel Input Fields

`Apply`
: Select this option to automatically apply an associated structure when using the Data Editor to edit or browse a dataset without specifying an explicit structure.

  For instance if the **B** or **EU** line-commands are issued from a dataset/library list, or the **GO** primary command is issued to switch to the Data Editor from an existing browse/edit view.

`Save`
: Select this option to automatically add or update an entry in the associations table whenever the Data Editor is used to edit or browse a dataset with an explicit structure.

  For instance if the Data Edit panel is used to edit a data file with a copybook mapping, or a **USING** parameter is spcified on an 'SD EDIT/BROWSE' command executed from your HOME file using the ACTION key.

Primary Commands

The following primary commands are supported.

EDIT (E)

Open the Data file to Copybook Associations table edit view and modify the defined associations.

Generic associations may be defined by speciying standard dataset/member name wildcard characters in the data file mask (DSN) column entry.

List Window Settings (=0.5)

The List Window Settings panel (ZZSGSET5) is an interactive panel window, opened on selection of option 5. in the SELCOPYi Settings panel.

This panel specifies options relating to the SELCOPYi list windows.

Panel Input Fields

ENTER Key Action>
For file list windows only (window class LISTFILE), identifies the default action on pressing the <Enter> key on a list entry.

Possible actions are:

Edit	Open a CBLe text edit window to edit the file.
View	Open a CBLe text edit window to view the file (read-only).
Browse	Open a structure data edit (SDE) window to browse the file.
SDE	Open a structure data edit (SDE) window to perform full function edit of the file.
SDEU	Open a structure data edit (SDE) window to perform Update-in-place edit of the file.
None	Disable all actions on the <Enter> key.

Batch Settings (=0.6)

The Batch Settings panel (ZZSGSET6) is an interactive panel window, opened on selection of option 6. in the SELCOPYi Settings panel.

This panel specifies options relating to SELCOPYi batch job generation.

Panel Input Fields

JOB Statement>
Specifies a //**JOB** statement that will be automatically inserted at the start of any z/OS batch job generated using SELCOPYi, or whenever the *JOBCARD* command is issued from a text-edit session. A maximum of 4 lines may be specified.

Substitution will automatically occur for any embedded environment variables, which are specified using standard notation (normally by enclosing the name in per-cent signs) , e.g.

```
//%fn%  JOB ,,CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID
```

In the example above, the edited library member name is used for the job name.

SYSOUT Class >
Specifies the SYSOUT class that will be specified for //SYSPRINT or //SDEPRINT DD statements in any z/OS batch job generated using SELCOPYi.

SDSF FastPath>
Specifies the fast-path that may be entered at the ISPF Primary menu panel in order to start SDSF (or an equivalent product) in an ISPF environment.

This option is used by SELCOPYi's *OQ* and *OP* commands in order to display JES2/JES3 job queues and the operator console respectively.

Note that whenever the *SUB* command is used to submit a batch job from the SELCOPYi text-editor, *OQ* is automatically invoked to display the output from that particular job.

SDE Print File>
Specifies the print output fileid which is allocated to DDname SDEPRINT when the primary command PRINT is invoked interactively without parameter SYSOUT=*outputclass*.

DB2 Settings (=0.7)

The DB2 Settings panel (ZZSGSET7) is an interactive panel window, opened on selection of option 7. from the SELCOPYi Settings panel.

Options

1 Audit	Audit Log Dataset Options
2 ExecSQL	ExecSQL Output Options
3 SubSys	Set DB2 Subsystem specific options (Load lib paths etc)

Execute SQL Settings (=0.7.2)

The Execute SQL Settings panel (ZZSGSET1) is an interactive panel window, opened on selection of option 7.2 in the SELCOPYi Settings panel.

This panel specifies the out put options for the ExecSQL primary command.

Panel Input Fields

`Output Dataset`
: Specify the name of the output dataset written by ExecSQL. If left blank a dataset name will be generated in the format ***prefix*.ZZS2ZSQL.D*yyyyddd*.T*hhmmssx*.LST**.

`Output Structure`
: Specify the name of the output structure (SDO) dataset written by ExecSQL. If left blank a dataset name will be generated in the format ***prefix*.ZZS2ZSQL.D*yyyyddd*.T*hhmmssx*.SDO**.

DB2 Subsystem Options Settings (=0.8)

The DB2 Subsystem Options panel (ZZS2SET3) is an interactive panel window, opened on selection of option 3. from the SELCOPYi DB2 Settings panel.

The menu panel provides an option for your site's system administrator to edit a table in which options may be supplied for each DB2 subsystem that is to be accessed using SELCOPYi.

The table consists of three columns (SID, Option and Value).

The **SID** column names the DB2 subsystem for which an option should apply.

The **Option** columns indicates the particular option for which a value is supplied. At present only one option (**LLIB**) is supported. The LLIB option allows a DB2 load library path to be specified for each SID overcoming a previous restriction that subsytems defined on incompatible levels of DB2 (e.g. DB2 v9 versus DB2 v12) could not be accessed in the same SELCOPYi session.

The **Value** column supplies a value for the option. e.g. a single load library dataset name.

Multiple rows should be supplied where the option refers to a library concatenation e.g. **LLIB**

Options

1 Private	Edit your personal DB2 Subsystems options table
2 Shared	Edit the site-wide DB2 Subsystems options table
3 Product	View the product supplied skeleton table

`1. Private`
: Edits the user's own personal DB2 Subsystems options table as stored on disk.

  If this table does not already exist then a copy of the site (or product) table will be made.

`2. Shared`
: Edits the installation specific SHARED DB2 Subsystems options table as stored on disk. This table should be maintained by your systems programmer in order to define site specific global settings specific to each DB2 subsystem. e.g. the load library path.

  If this table does not already exist then a copy of the product supplied skeleton will be made.

`3. Product`
: Browses the SELCOPYi PRODUCT supplied skeleton DB2 Subsystems options table.

Search/Update Settings (=0.9)

The Search/Update (FSU) Settings panel (ZZSGSET9) is an interactive panel window, opened on selection of option 9. from the SELCOPY/i Settings panel.

This panel specifies options relating to the default display of the formatted report output from the File Search/Update/Copy/Remap (FSU) utility.

Following completion of a foreground execution of the FSU utility, a standard set of Hit record report columns are displayed based on the the width of the 3270 terminal display.

Where the terminal display is greater than 80 columns wide, all Hit record columns are displayed by default. This settings panel allows selection of the report columns to be displayed at the start of the Hit report line following the zDSN and zMember columns but before the zRecord column (or column group). It also determines whether these columns are to be held when scrolling left or right.

Columns that are deselected in this settings panel will be displayed following the zRecord column (or column group).

Panel Input Fields

`Show selected fields only.`
Enter "/" in this input field to activate the FSU report column selections that follow.

`zRecNo`
Enter "/" in this input field to include display of the zRecNo column. This column displays the record number within the data set, HFS file or library member for which a match has been found.

`zHitNo`
Enter "/" in this input field to include display of the zHitNo column. This column displays the occurrence number of the match within the data set, HFS file or library member.

`zLrecl`
Enter "/" in this input field to include display of the zLrecl column. This column displays the length of the record in which the match was found.

`zHits`
Enter "/" in this input field to include display of the zHits column. This column displays the number of matches found within the record.

`Hold`
Enter "/" in this input field to hold display of the zDSN and zMember columns, and any of the columns selected above, when scrolling left and right.

Text Edit (=1)

Text Edit Panel

The Text Edit Entry panel (ZZSGEDIT) is an interactive panel window, opened on selection of option 1. in the SELCOPYi Primary option menu.

This panel is used to open an existing cataloged or uncataloged data set PDS/PDSE library member or HFS file for CBLe text edit. For full documentation on text edit sessions, please refer to publication *"SELCOPYi Text Editor (CBLe)"*.

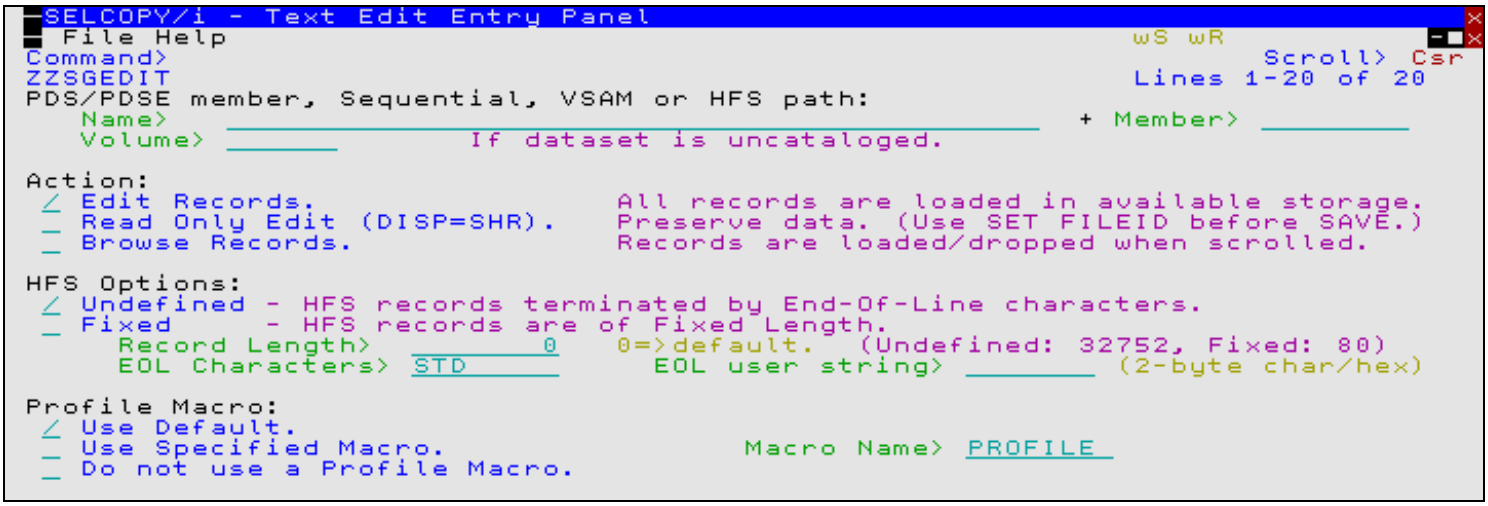


Figure 14. SELCOPY/i - Text Edit Entry Panel.

Panel Input Fields

`PDS/PDSE member, Sequential, VSAM or HFS path:`
Fields that identify the existing sequential or VSAM data set, HFS file or PDS/PDSE library member to be edited.

`Name>`
`Member>`
An absolute or relative HFS Path name or the fully qualified name of a sequential data set or PDS/PDSE library member. Note that the Member field is ignored if Name field is not a library or contains wildcard characters. Dataset names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards '%' (percent) or "*" (asterisk) are entered, or dataset is a PDS/PDSE library and member is left blank.

`Volume>`
Specifies a volume serial id mask for an uncataloged data set. (Not applicable to HFS files.)

If an entry exists in this field, then the selectable list of files will be generated from a list of VTOC entries for the specified volume.

`Action:`
Identifies the action taken by the panel on pressing the <Enter> key. Mutually exclusive options are as follow:

◊ **Edit Records**
Open the file with exclusive SPFEDIT ENQ and load all records into available storage for full text edit capabilities.

◊ **Read Only Edit (DISP=SHR)**
Open the file with shared SPFEDIT ENQ and load all records into available storage before releasing the ENQ. View supports full text edit capabilities.

Attempts to save changes will fail with message ZZSE045E. However, use of File menu bar option "**Save As**" will allow save of the in-storage data as a different fileid. Alternatively, SET FILEID or any other text edit SET commands that manipulate the fileid assigned to in-storage records, may be used prior to SAVE.

If the new fileid is an as yet unallocated data set, then the relevant Allocate Non-VSAM or Define VSAM KSDS/ESDS/RRDS/LDS dialog panel will be opened as appropriate.

◊ **Edit Records**
Open the file with shared SYSDSN ENQ and load only enough records to fill the SDE edit view display area. Records are loaded and dropped from storage when scrolling occurs. Browse does not allow record editing.

HFS Options:

Identifies options used specifically for editing HFS files. Mutually exclusive options are as follow:

◊ **Undefined**
End-of-Line (EOL) characters are used to delimit the end of records.

For Undefined HFS files, the Record Length field defines the **maximum** length of a record within which to identify the EOL character combination. If the EOL combination is not found, the record is chopped at this length. Default maximum record length is 32752.

◊ **Fixed**
HFS file is to be chopped into a number of records of fixed length.

For Fixed HFS files, the EOL Characters field is ignored and Record Length field defines the **actual** length of each record. Also, message ZZSE178E is displayed if the file size is not a multiple of the record length. Default record length is 80.

◊ **Record Length>**
The maximum record length used for Undefined HFS files or the actual record length used for Fixed HFS files. Zero (0) implies the default value of 32752 for Undefined and 80 for Fixed.

◊ **EOL Characters>**
One of the permitted EOL values that represent a 1 or 2 byte end-of-line character combination. Entering a blank or invalid value in this field will display a selectable list of valid entries.

EOL value elements are as follow:

STD	-	Any of the standard EOL combinations.
NL	X'15'	New Line.
CR	X'0D'	Carriage Return.
LF	X'0A'	Line Feed.
user	-	A user specified EOL combination.

STD is default so that input data is scanned for any of the standard EOL combinations, stopping when one is found. This EOL combination is then used for the remainder of the file data.

◊ **EOL user string>**
If **user** has been selected from the EOL Characters> field, then this field is used to identify a user defined 1 or 2 byte character or hex string. e.g. '##', X'FF'.

Profile Macro:

Indicates whether or not a text edit profile macro is to be executed when the Edit or View text edit window view is opened and, if so, the name of the profile macro to be executed.

Mutually exclusive options are as follow:

◊ **Use Default.**
Use the default text edit profile macro (PROFILE).

◊ **Use Specified Macro.**
Use an edit profile macro with macro name specified by the Macro Name> field.

◊ **Do not use a Profile Macro.**
No edit profile macro is to be executed - all edit options are default.

Macro Name>

Applicable only if option Use Specified Macro is selected. This field names the text edit profile macro to be executed when Edit or View action is used.

Data Edit (=2)

Structured Data Browse/Edit Panel

The Structured Data Browse/Edit Panel panel (ZZSGSDE0) is an interactive panel window, used to open an existing file for Structured Data (SDE) edit.

For full documentation on the Structured Data Editor, please refer to publication *"SELCOPY/i Structured Data Editor (SDE)"*.

The panel may be opened via the following:

- Select option 2. 'Data Edit' in the SELCOPY/i Primary option menu.
- Select 'Structured edit' from the File menu in the main window menu bar.
- Execute the SDE command EDITDIALOG on the command line of an existing SDE window view, or from a SELCOPY/i text edit view if preceded by SDATA.
- Execute the command SDE with no parameters.
- Execute the prefix command "SD" from an Execute CBLVCAT or file list window.

The SELCOPY/i Structured Browse/Edit panel generates an SDE BROWSE or EDIT command to open an SDE window view in the current frame window.

Depending on whether the **Additional Options** option has been set, pressing the <Enter> key or, if configured, double-clicking the left mouse button, will will either action the data edit or display the Structured Data Browse/Edit - Options panel view.

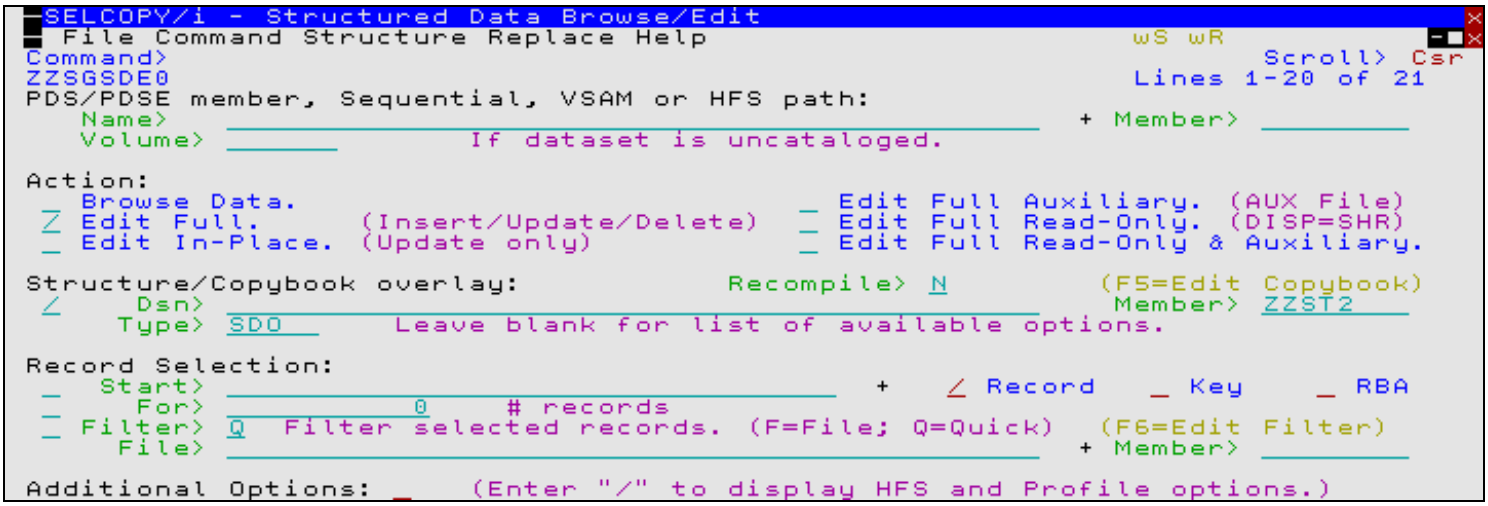


Figure 15. Structured Data Browse/Edit Panel.

Menu Bar Items

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values entered in the panel so that they may be redisplayed the next time the panel is opened.

Command
: Generate the BROWSE or EDIT command line syntax for specified field entries and display it in a temporary CMX file text edit view. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

  The user has the opportunity to edit the command prior to its execution and/or copying it to the home (CMX) command centre for future reference and re-execution.

Structure
: Opens the Create Structure (SDO) Menu to generate a SELCOPY/i SDO from a source Assembler, COBOL or PL1 Copybook or an XREF file.

Replace
: Opens the COBOL Compiler Options panel to review and, if necessary, add COBOL REPLACE "From" and "To" pseudo-text values to be used in compiling a COBOL copybook.

  Values entered in this panel apply only to the current user. System wide COBOL REPLACE values may also have been entered in the SELCOPY/i Site INI file. (See the *"SELCOPY Product Suite Customisation Guide"* for details.)

Help
: Display help for this panel view.

Panel Input Fields

By default, field entries are populated with arguments and options that were entered the last time the Structured Edit dialog window was used.

Many field entries are optional and need to be activated by entering "/" in the preceding field. This provides easy deactivation/reactivation of a field value without having to clear the field.

PDS/PDSE member, Sequential, VSAM or HFS path:
Input fields which together identify a single existing, sequential or VSAM data set, GDG file generation, HFS file or PDS/PDSE library member to be edited.

Name>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Volume>
Specifies the name of the input data set volume. This is required only if output is to an uncataloged data set.

Format: TABL | SNGL | DUMP
TABL
Multi-record table format.
SNGL
Single-record format.

Recommended for edit/browse of spanned or HFS files that may have very long records (>32K) since TABL mode would require storage to hold at least a screen-full of records. Note that SELCOPYi will handle spanned (RECFM=VBS) records with a length of up to 16MB each.

DUMP
Single-record Hex Dump format.

Action:
Select one of the mutually exclusive options that identify the edit technique to be performed.

- **Browse Data**
  Browse the data only.
- **Edit Full**
  Edit the data with full editing (record insert/delete) capability. Editing techniques employed are KSDS edit for KSDS data sets otherwise, Auxiliary Edit if the file is too large to be loaded into available storage or standard in-storage edit.
- **Edit In-Place**
  Edit the data with only update capability. Record length may not be changed and records may not be inserted or deleted. Editing techniques are the same as for Edit Full but that Auxiliary Edit is never employed as, for large files, records will be loaded and dropped from storage as appropriate.
- **Edit Full Auxiliary**
  As for Edit Full but that Auxiliary Edit will be used for edit of non-KSDS files regardless of their size. As such, no attempt will be made to load all the file's records into storage. This is of benefit when many large files, that would otherwise be succesfully loaded into available storage, are to be edited concurrently.
- **Edit Full Read-Only**
  As for Edit Full but that changes to the data may not be saved using its original DSN or HFS fileid. SAVEAS must be used instead to save the data to a different (potentially new) fileid.
- **Edit Full Read-Only & Auxiliary**
  As for Edit Full Read-Only but with the enforced use of Auxiliary Edit as described by Edit Full Auxiliary.

Structure/Copybook overlay:
If activated this option defines fields which together specify a cataloged structure file (Assembler, COBOL or PL1 Copybook, ADATA file or a SELCOPY/i SDO) used to the browsed/edited records. The structure may be a sequential data set or a PDS/PDSE library member.

If no structure is specified, each data set record will be of the default record type "Unmapped", i.e. a single character field of length equal to that of the record.

Dsn>
Identifies the fully qualified data set name of a sequential data set or PDS/PDSE library containing an Assembler, COBOL or PL1 Copybook, ADATA file or SDO structure.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Type:
Indicate the type of structure (ASM, COBOL, PL1, ADATA or SDO).

Recompile>
If *Structure/Copybook overlay* refers to an Assembler, COBOL, PL1 or ADATA source file (not an SDO) then a compile step must be performed in order to turn the source copybook into SELCOPYi's own internal structure (SDO) format. A temporary SDO will be created, lasting the duration of the SELCOPYi session, making subsequent reference to the same copybook during the session much faster.

If, however, the copybook (or any of its included components) is modified during the session, then a recompile of the source will normally be required.

For performance reasons **N** (meaning no recompile should occur) is the default.

Specify **Y** to force a copybook recompile each time the process is run. Alternatively type the primary command:
SD DROP *<copybook_name>*

Record Selection:
Fields which together identify criteria by which a subset of records from the file are selected for browse/edit. If any record selection field is activated for data edit, then Edit In-Place is performed regardless of the selected edit technique.

Start>
If activated, the **Start>** field specifies the first record in the file to be included in the browse/edit. Records occurring sequentially before the start record will be excluded. If this field is not activated, records are selected beginning at record 1.

This input field may contain a record number, an RBA number (for ESDS input only), or a key string (for KSDS input only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

Record | Key | RBA
Identifies the type of start value specified in the **Start>** field. Enter "/" in the appropriate, mutually exclusive parameter field.

For>
If activated, the **For>** field specifies the maximum number of records within the file to be browsed/edited. If this field is not activated, records are selected beginning at start record and ending at the last record in the file.

Filter>
If activated, the **Filter>** field specifies options to either generate a new record filter or use an existing record filter file. A record filter will perform further subsetting on input file records selected for processing by the Start> and/or For> input fields.

Filter options are as follow:

Q On pressing <PF6>, the Quick Filter dialog panel will be opened in order to generate a temporary filter on the formatted or unformatted record data, depending on whether a structure/copybook is activated.

If you require a structure to view you data, but wish to specify your filter on the unformatted data (i.e without referring to field names) then temporarily deactivate your structure/copybook (by removing the slash on the left hand side of the screen) while your filter is being defined.

F

Use a permanent filter identified by the sequential data set or member name in the File> field.

On pressing <PF6>, the Create File Filter dialog panel will be opened in order to display the contents of an existing filter file or create and save a new filter file.

If option "F" is selected, then specifiaction of a filter fileid is mandatory.

File>
Identifies the fully qualified data set name of a sequential data set or PDS/PDSE library containing a record filter. Quotes are unnecessary but permitted.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the File> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Additional Options:
Select this option only if HFS file record length determination is not a standard End-of-Line character combination. Similarly if no edit profile macro is to be executed or an edit macro which is not the default (SDEPROF) is to be executed when the file is edited.

If selected, the Structured Data Browse/Edit - Options panel view will open on pressing <Enter>.

Structured Data Browse/Edit - Options Panel

The Structured Data Browse/Edit - Options panel is opened when Additional Options is selected in the Structured Data Browse/Edit panel.

This panel is used to specify additional data edit options for initial edit profile macro execution and HFS file record determination.

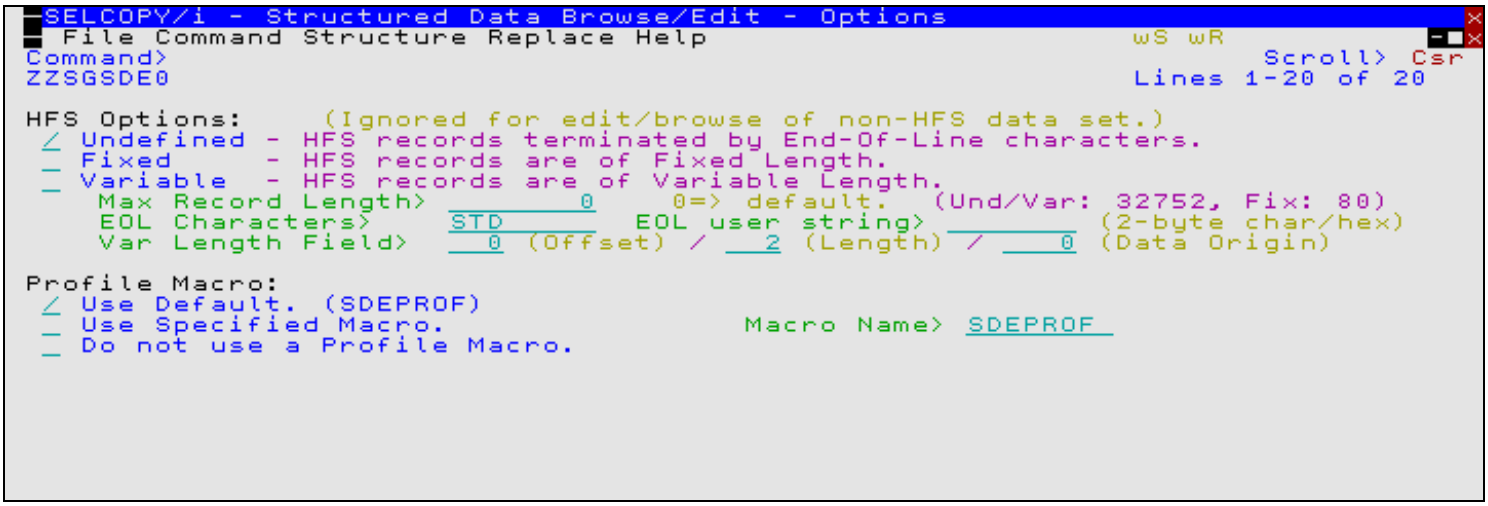


Figure 16. Structured Data Browse/Edit - Options panel view.

Menu Bar Items

Menu bar items are as described for Structured Data Browse/Edit .

Panel Input Fields

HFS Options:
Options and values that apply to edit of HFS files only.

Undefined | Fixed | Variable
Identify the format of input HFS records.
Undefined indicates that records are terminated by an End-of-Line (EOL) string.
Fixed indicates that all records are of a fixed length as defined by a specified LRECL.
Variable indicates that all records are of variable length as defined by a length field within the data.

Max Record Length>
Applicable to each of the record formats, this value defines the LRECL (maximum length) of input records. A record longer than this value will be chopped into multiple records.
A 0 (zero) value implies the default which is 32752 for Undefined and Variable record formats and 80 for Fixed record format.

EOL Characters>
Applicable to Undefined record format only, choose from one of the following EOL character combinations:

STD	-	Any standard line-end.
CR	X'0D'	Carriage Return.
LF	X'0A'	Line Feed.
NL	X'15'	New Line.
CRLF	X'0D0A'	Carriage Return + Line Feed.

LFCR	X'0A0D'	Line Feed + Carriage Return.
CRNL	X'0D15'	Carriage Return + New Line.
user	-	A 2-byte user string specified in **EOL user string>**

EOL user string>
Applicable only if **EOL Characters>** is set to **user**, this field specifies the user supplied 2-byte EOL string. It may be specified in character or hexadecimal notation. (e.g. '##', X'FFFF')

Var Length Field>
Applicable to Variable record format only, these fields identify the location of the record length fields within the data.

(Offset)
Offset of the record length field from the start of the record. Default is 0. (i.e. the length field is at the start of the record.)

(Length)
Length (number of bytes) of the record length field. Default is 2.

(Data Origin)
Offset into the record data at which the value in the record length field is to be applied. Default is 0. (i.e. the record length include the length field.)

Profile Macro:
Options that identify the data edit macro to be executed when the file is edited.

- ◊ **Use Default.**
  Use the default supplied profile data edit macro SDEPROF
- ◊ **Use Specified Macro.**
  Use the edit macro named in the Macro Name> input field.
- ◊ **Do not use a Profile Macro.**
  Suppress use of a profile edit macro when the file is edited.

Macro Name>
Identies the edit macro to be executed if option **Use Specified Macro** is selected. This edit macro must be a member found in the macro library search path.

List File Windows (=3)

List Menu Panel

The List Menu panel (ZZSGLIST) is an interactive panel window opened on selection of option 3. in the SELCOPY/i Primary option menu.

SELCOPY/i file lists provide detailed information for DASD files and related system resources. (e.g. ENQs, DASD and Associated Cataloged objects.)

All list file windows are of window class, LISTFILE, and have common characteristics defined for list window classes.

Options

1 Volumes	**LVOL**	List DASD Volumes
2 VTOC	**LV**	List VTOC files
3 Extents	**LX**	List VTOC Extents
4 Dslist		Data Set List Utilities
5 Catalog	**LC**	List Cataloged datasets (catalog detail)
6 Dataset	**LD**	List Dataset details (catalog & VTOC detail)
7 Library	**LL**	List PDS/PDSE Library members
8 Allocated	**LA**	List Allocated files (DD names)
9 Enqueues	**LQ**	List Resource Enqueues
10 Job Enqueues	**LJQ**	List Job Enqueues
11 Associations	**LAS**	List File Assocations
12 HFS	**LP**	List HFS Paths
13 DB2	**LD**	List DB2 objects

List DASD Volumes (=3.1)

See List Windows for general features and commands common to all list windows.

The DASD Volumes List window may be opened via the following:

- Select option 1. 'Volumes' from the List Menu.
- Select 'DASD Volumes' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LVOL on the command line of any window.

The DASD Volumes window displays the attributes of DASD volumes defined to your system.

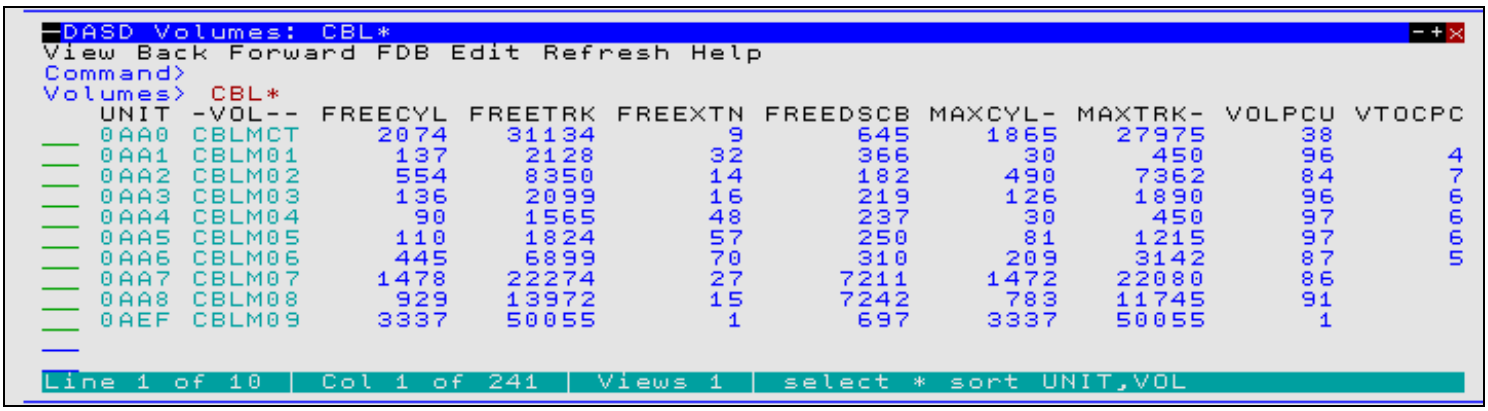


Figure 17. DASD Volumes window.

Panel Input Fields

`Volume>`

Specify a volume id mask. The mask supports the following wild cards:

`*` An asterisk indicates that one or more characters within the volume id can occupy that position. An asterisk can precede or follow a set of characters.

`%` A single percent sign indicates that exactly one character can occupy that position. (Up to 6 percent signs can be specified.)

By default, a volume id mask that is less than 6 characters in length and does not contain an * (asterisk) wild card will be treated as having an implied trailing * wild card.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix Line command T.
I	Open the volume statistics window for the volume containing the file.
T	Open the VTOC files list window for the volume.
VC	Open an Execute CBLVCAT window and issue a LISTVTOC operation for the entry.
X	Open the VTOC Extents list window for the volume.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

The data displayed for z/OS is:

Name	Type	Description
UNIT	Hex	Unit address
VOL	Char	Volume serial number
FREECYL	UInt	Free cylinders
FREETRK	UInt	Free tracks
FREEXTN	UInt	Free extents
FREEDSCB	UInt	Free DSCBs
MAXCYL	UInt	Largest free extent (CYLs)
MAXTRK	UInt	Largest free extent (TRKs)
VOLPCU	UInt	Volume percent used
VTOCPCU	UInt	VTOC percent used
TOTALCYL	UInt	Total cylinders
TRKCYL	UInt	Tracks per cylinder
TRKLEN	UInt	Track length
UCBTYPE	Hex	Unit type
SMS	Char	SMS managed indicator
VTOCIX	Char	Indexed VTOC
VTOCXTN	UInt	Number of VTOC extents
VTOCTRK	UInt	Number of VTOC tracks
LOWCCHH	Hex	VTOC start CCHH
HIGHCCHH	Hex	VTOC end CCHH
DSCBTRK	UInt	DSCBs per track
FREEVIR	UInt	Number of free VTOC index recs
FRAGINDX	UInt	Fragmentation index
ALTCCHH	Hex	Next available alt track CCHH
ALTREM	UInt	Remaining alternate tracks
MOUNT	Char	Mount usage status
DEV	Char	Device type
MODEL	Char	Device model
MODELX	Hex	Device model (hex)
CACHE	Char	Cached device
SHARE	Char	Shareable device
CYLMS	Char	Cylinder managed storage
EXATTR	Char	Extended attribute DSCBs
STARTCYL	UInt	Cylinder managed space start
MINCYL	UInt	Minimum cylinder allocation unit
ALTCYL	UInt	Number of alternate cylinders
STORGRP	VChar	SMS storage group

TMFREEXTN	UInt	Track managed total free extents
TMFREECYL	UInt	Track managed total free cylinders
TMFREETRK	UInt	Track managed total free tracks
TMMAXCYL	UInt	Track managed largest free extent (cylinders)
TMMAXTRK	UInt	Track managed largest free extent (tracks)
TMFRAGINDX	UInt	Track managed fragmentation index
VOLTRACKS	UInt	Total tracks on volume
TMTRACKS	UInt	Track managed total tracks on volume

The data displayed for z/VSE is:

Name	Type	Description
UNIT	Hex	Unit address
VOL	Char	Volume serial number
TYPE	Char	External device type code
FORMAT	Hex	Device format
AVRVTOC	Hex	VTOC address
PUBC	Hex	PUB device type code
DTFC	Hex	DTF device type code
UCBC	Hex	Unit code
DCTPCYL	UInt	Primary cylinders
DCTACYL	UInt	Alternate cylinders
DCTTCYL	UInt	Tracks per cylinder
DCTBTRK	UInt	Bytes per track
DCTTFIX	UInt	Cylinders under fixed head
DCTMAXR	UInt	Maximum physical record size
DCTDEVC	Hex	Device constants

List VTOC Files (=3.2)

See List Windows for general features and commands common to all list windows.

The VTOC File List window may be opened via the following:

- Select option 2. 'VTOC' from the List Menu.
- Select 'VTOC Files' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LV on the command line of any window.

The VTOC File List window displays data set entry information in a DASD volume's Volume Table of Contents (VTOC).

Note: List VTOC Files is not supported for CMS.

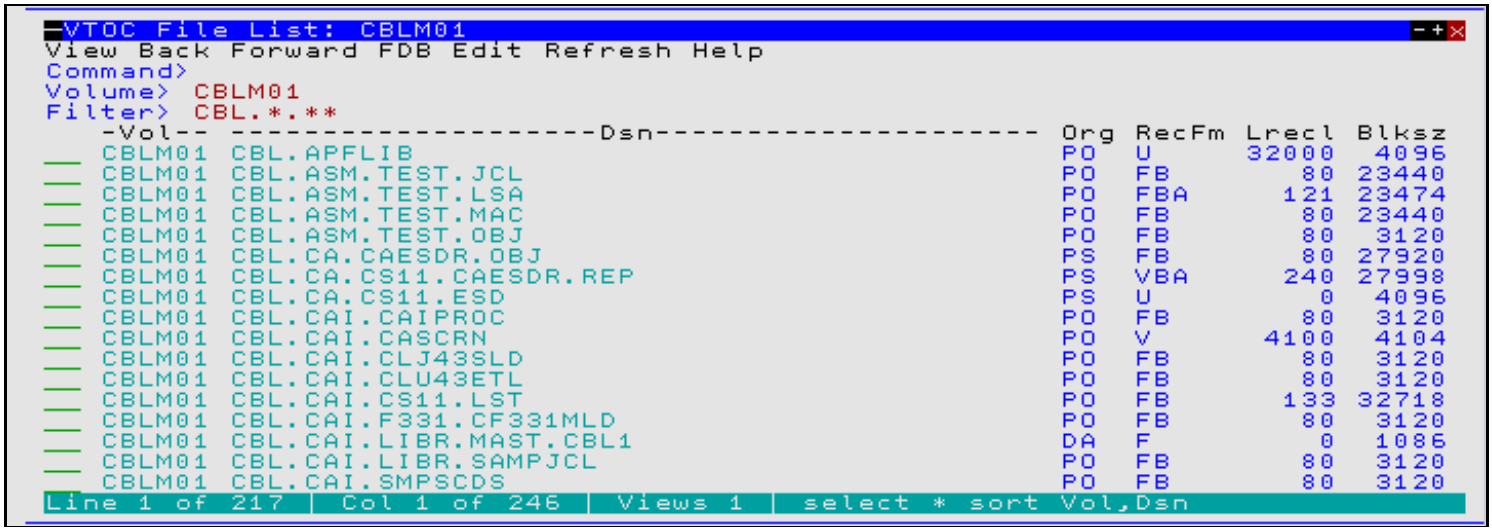


Figure 18. VTOC File List window displaying all entries beginning 'CBL.' on volume 'CBLM01.'

Panel Input Fields

Volume>
The 1-6 character volume id containing the required VTOC.

DSN mask>
Note: This parameter is not supported for z/VSE.

Select only data sets that match the specified filter mask. The filter mask supports the following wild cards:

*	A single asterisk represents a DSN qualifier, or zero or more characters within a DSN qualifier.
**	A double asterisk represents zero or more qualifiers within a DSN. Double asterisk must be preceded or followed by either a "." (dot/period) or a blank. It cannot precede or follow an alphanumeric character.
%	A single percent sign represents exactly one character, other than "." (dot/period), within a DSN qualifier. Up to 8 percent signs can be specified in each qualifier.

A filter field that contains **neither** "*" (asterisk) nor "**" (double asterisk) wild cards will have a wildcard string of "*.**" automatically appended and so list all those data sets whose names begin with the filter string.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command M if entry is a PDS/PDSE library, prefix line command E otherwise.
A	Open the Define Catalog Alias panel for this entry.
AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.
AS	Open an Associations list window to list associated objects for this entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	For z/OS only, open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
G	For z/OS only, open the Library Member Generations List window.
FO	Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
FS	If the entry is a PDS(E), open the File Search window for the entry.
I	Open a Data Set Information panel display for the entry.
IC	Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.
ID	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
K	Delete (Kill) the entry without prompting for verification.
M	If the entry is a PDS/PDSE, open a Library List window. (Default)
Q	For z/OS only, list dataset enqueues (major name SYSDSN) for this entry.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
T	Open an Execute CBLVCAT window to issue a LISTVCAT TUNE operation for the entry.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
Z	Perform a compress of a z/OS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
?	Open the volume statistics window for the volume containing the entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

Name	Type	Description
Dsn	Char	Dataset name
Org	Enum	Data set organisation
Alu	Char	Allocation unit
Pri	UInt	Primary space allocation
Sec	UInt	Secondary space allocation
Alt	UInt	Allocation total
Nxt	UInt	Number of extents
Trks	UInt	Tracks allocated
DsnPcu	UInt	Dataset percent used
DsKb	UInt	Dataset space Kilobytes
Blksz	UInt	Block Size
Lrecl	UInt	Logical record length
RecFm	Enum	Record format
Created	VTOCDate	Creation date
Referenced	VTOCDate	Last referenced date
Expires	VTOCDate	Expiry Date
SMi	Hex	SMS indicators (DS1SMSFG)
VS	Hex	VSAM indicators (DS1OPTCD)
DSInd	Hex	Dataset indicators (DS1DSIND)
KyL	UInt	Dataset key length
RKP	UInt	Relative key position
TBal	UInt	Bytes remaining on last track
BlkTrk	UInt	Blocks per track
Vol	Char	Volume serial number
F1Vol	Char	Format 1 DSCB volume serial
VSeq	UInt	Volume sequence number
Flag1	Hex	DS - Dataset flags (DS1FLAG1) (hex)
LastTrack	UInt	DS - Last used track
LastBlock	UInt	DS - Last used block on last used track
LastFree	UInt	DS - Space remaining on last used track

List VTOC Extents (=3.3)

See List Windows for general features and commands common to all list windows.

The VTOC Extent List window may be opened via the following:

- Select option 3. 'Extents' from the List Menu.
- Select 'VTOC Extents' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LX on the command line of any window.

The VTOC Extent List window displays all information in a DASD volume's Volume Table of Contents (VTOC) by physical extent. This includes free extents and volume control areas such as the VTOC and the label area.

Note: Not supported for z/VSE.

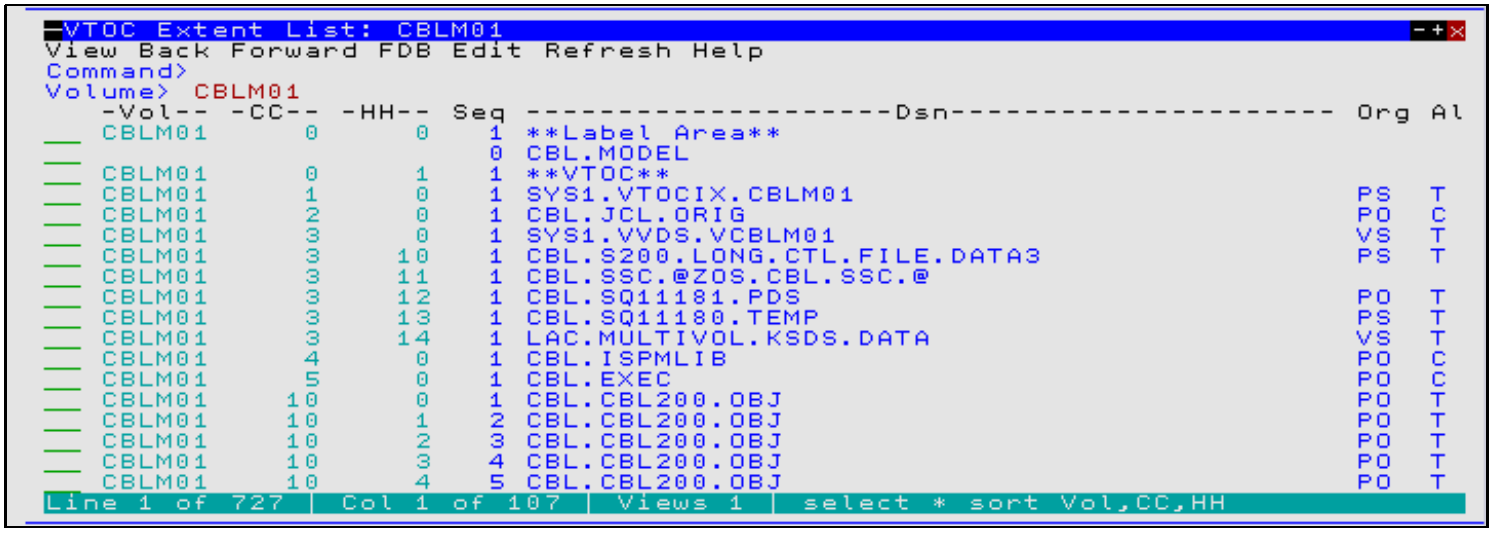


Figure 19. VTOC Extent List window displaying all extents on volume 'CBLM01.'

Panel Input Fields

Volume>
The 1-6 character volume id containing the required VTOC.

DSN mask>
Select only data sets that match the specified filter mask. The filter mask supports the following wild cards:

*	A single asterisk represents a DSN qualifier, or zero or more characters within a DSN qualifier.
**	A double asterisk represents zero or more qualifiers within a DSN. Double asterisk must be preceded or followed by either a "." (dot/period) or a blank. It cannot precede or follow an alphanumeric character.
%	A single percent sign represents exactly one character, other than "." (dot/period), within a DSN qualifier. Up to 8 percent signs can be specified in each qualifier.

A filter field that contains **neither** "*" (asterisk) nor "**" (double asterisk) wild cards will have a wildcard string of "*.**" automatically appended and so list all those data sets whose names begin with the filter string.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command M if entry is a PDS/PDSE library, prefix line command E otherwise.
A	Open the Define Catalog Alias panel for this entry.
AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.
AS	Open an Associations list window to list associated objects for this entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FO	Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
FS	If the entry is a PDS(E), open the File Search window for the entry.
G	For z/OS only, open the Library Member Generations List window.
I	Open a Data Set Information panel display for the entry.
IC	Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.
ID	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
K	Delete (Kill) the entry without prompting for verification.
M	If the entry is a PDS/PDSE, open a Library List window. (Default)

Q	List dataset enqueues (major name SYSDSN) for this entry.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
T	Open an Execute CBLVCAT window to issue a LISTVCAT TUNE operation for the entry.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
Z	Perform a compress of an z/OS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
?	Open the volume statistics window for the volume containing the entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

Name	Type	Description
Vol	Char	Volume serial number
CC	UInt	Cylinder number (decimal)
HH	UInt	Head number (decimal)
Seq	UInt	Extent sequence
Dsn	Char	Dataset name
Org	Enum	Data set organisation
Alu	Char	Allocation unit
Trks	UInt	Tracks allocated
Nxt	UInt	Number of extents
LoCCHH	Hex	Extent Low CCHH
HiCCHH	Hex	Extent High CCHH

Data Set List Utility (=3.4)

The Data Set List Utility panel (ZZSGOPEN) is an interactive panel window, opened on selection of option 4. 'Dslist' from the List Menu.

This panel is used to perform tasks on an existing cataloged or uncataloged data set PDS/PDSE library member or HFS file for CBLe text edit.

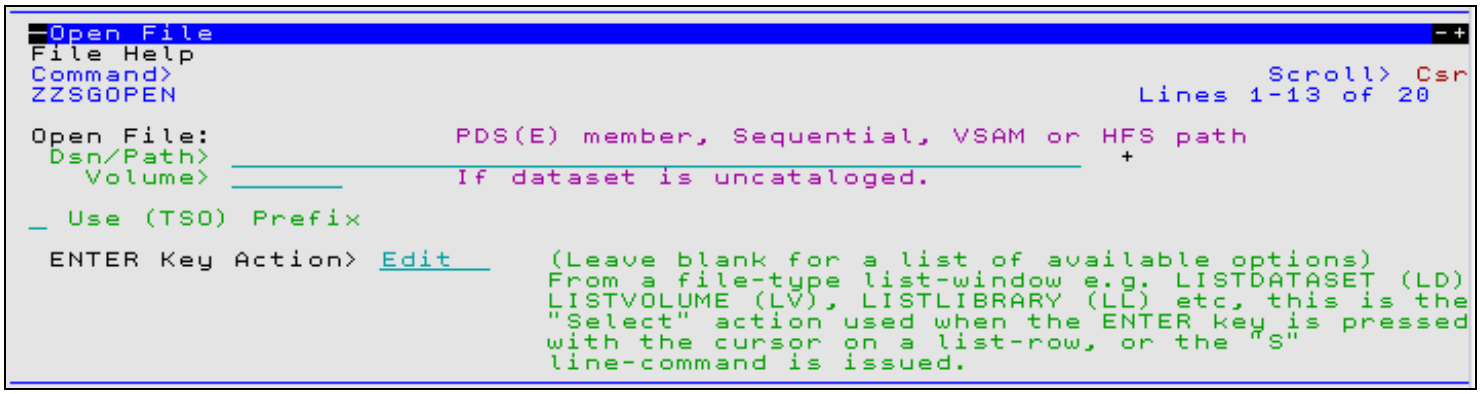


Figure 20. SELCOPY/i - Data Set List Utility.

Panel Input Fields

Fileid Mask:
Fields that identify the existing sequential or VSAM data set, HFS file or PDS/PDSE library member to be edited.

Dsn/Path>
An absolute or relative HFS Path name or the fully qualified name of a sequential data set or PDS/PDSE library. Enclosing apostrophes (') may be used to ignore use of the TSO prefix if selected.

A selectable list of data set names or HFS files will be displayed as appropriate if either wild card character "%" (percent), representing a single character, or "*" (asterisk), representing zero or more characters, is specified. If a volume id exists in the Volume field, then a list of selectable data sets will be restricted to those contained in that volume's VTOC.

Volume>
Specifies a volume serial id mask for an uncataloged data set. (Not applicable to HFS files.)

Use (TSO) Prefix
Option field that controls whether the defined TSO prefix is used as the high level qualifier of the data set to be text edited.

Note that the TSO prefix will not be used if the data set name is enclosed in apostrophes (') or the fileid is an HFS file path.

ENTER Key Action>
Option field that controls the action on pressing the <Enter> key or, if configured, double-clicking the left mouse button on an entry in the resulting file list.

Acceptable values are as follow:

Edit	Edit using the CBLe Text editor with Read/Write authority.
View	Edit using the CBLe Text editor with Read Only authority.
Browse	Browse data using the SDE Structured Data editor.
SDE	Full Edit of data using the SDE Structured Data editor.
SDEU	Update-in-place Edit of data using the SDE Structured Data editor.
None	No action to be taken. A list prefix command must be used instead.

List Catalog Entries (=3.5)

See List Windows for general features and commands common to all list windows.

The Catalog List window may be opened via the following:

- Select option 5. 'Catalog' from the List Menu.
- Select 'Cataloged Files' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LC on the command line of any window.

For z/VM CMS, the File List window is opened in place of the Catalog List or Dataset List window and displays information about files residing on accessed mini-disks. Refer to documentation on List CMS Files.

For z/VSE, the Catalog List window is supported only where the CBL software product CBLVCAT is installed and active. The Catalog List window uses CBLVCAT to read the specified VSAM catalog records to obtain information about the cataloged files.

For z/OS, the Catalog List window displays the basic catalog entry information for ICF cataloged data sets. The Dataset List window should be used to display more detailed information on cataloged data sets.

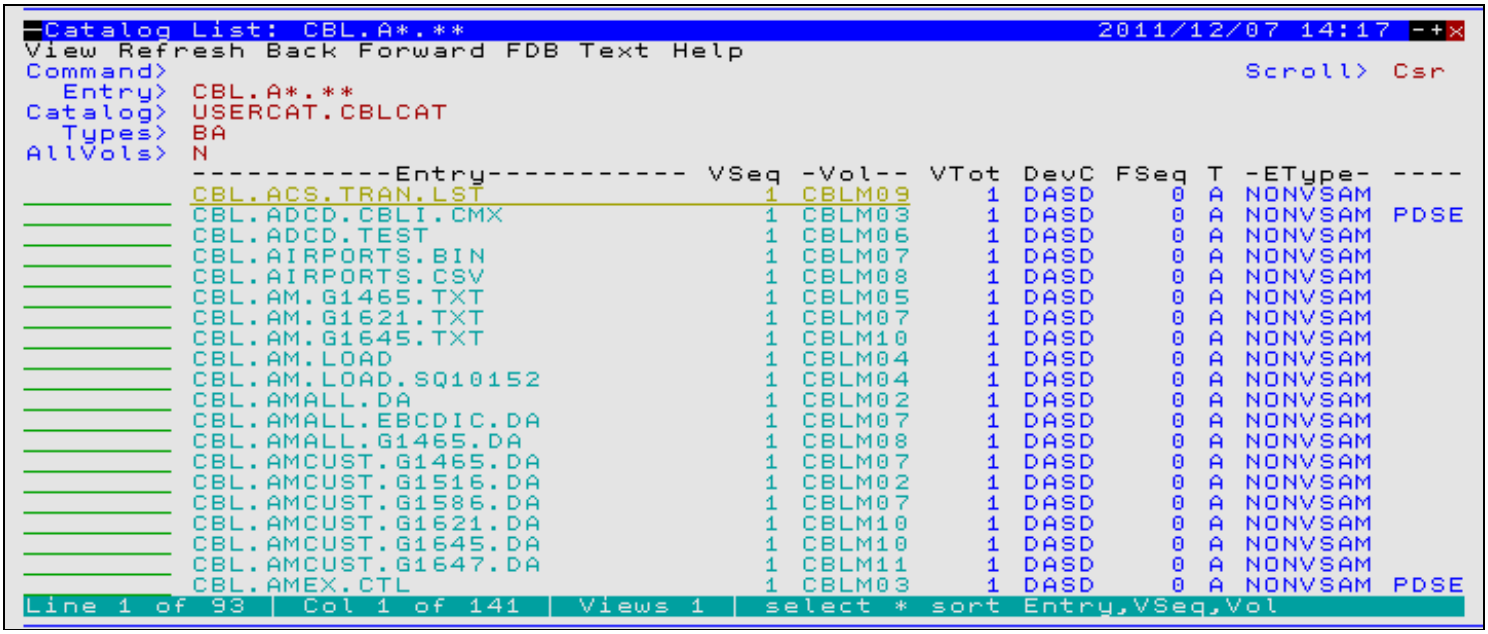


Figure 21. z/OS Catalog List Window.

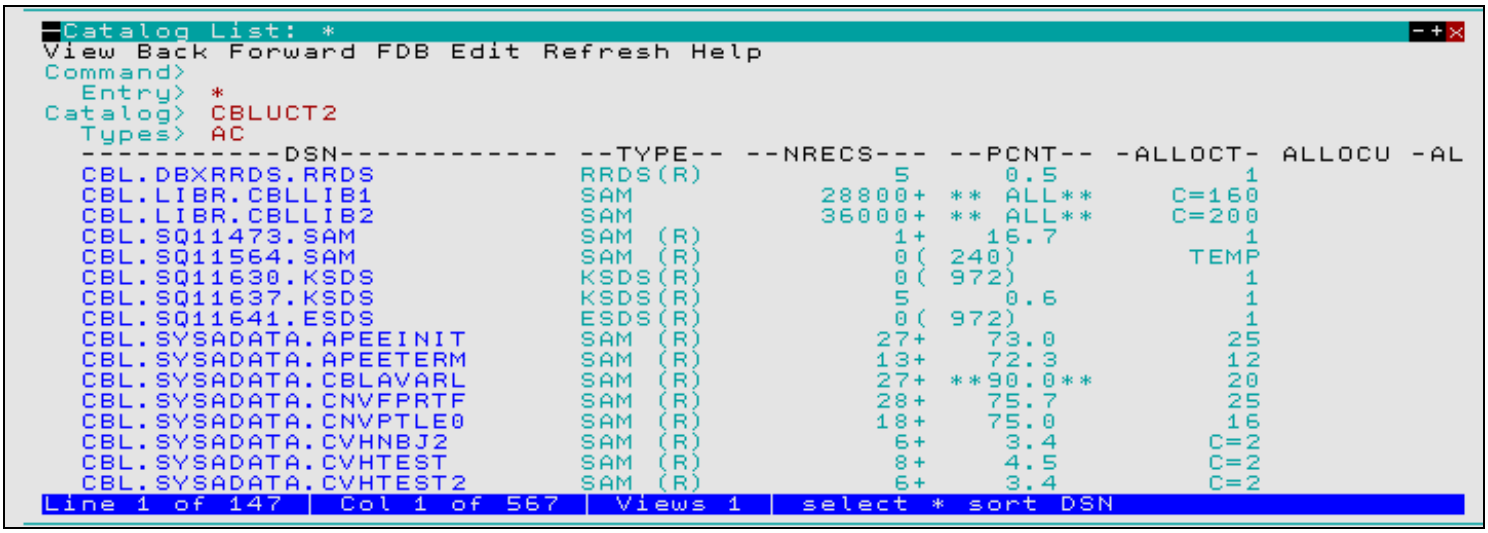


Figure 22. z/VSE Catalog List Window.

Panel Input Fields

DSN mask>

Specify the dataset name mask.

z/OS Systems:

On z/OS systems, the mask supports the following wild cards:

*	A single asterisk represents a DSN qualifier, or zero or more characters within a DSN qualifier.
**	A double asterisk represents zero or more qualifiers within a DSN. Double asterisk must be preceded or followed by either a "." (dot/period) or a blank. It cannot precede or follow an alphanumeric character.
%	A single percent sign represents exactly one character, other than "." (dot/period), within a DSN qualifier. Up to 8 percent signs can be specified in each qualifier.

If the last character of the DSN mask is "." (dot/period), then this marks the end of the low level DSN qualifier within the DSN mask. The trailing "." is stripped and no wildcard string is appended to the DSN mask. e.g.

```
DEV*.                 becomes:   DEV*
DEV.OEM.TRSPAN*.      becomes:   DEV.OEM.TRSPAN*
DEV.*.*SAMP%%.        becomes:   DEV.*.*SAMP%%
```

If the last character of the DSN mask is **not** "." (dot/period), then a default trailing wild card string is automatically appended to the DSN mask as follows:

1. If the DSN mask is a single qualifier or the last character of the DSN mask is "*" (asterisk), then a wildcard string of ".**" is appended. e.g.

```
DEV                  becomes:   DEV.**
DEV*                 becomes:   DEV*.**
DEV.OEM.TRSPAN*      becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA*          becomes:   DEV.*.*SPA*.**
```

2. Otherwise a wildcard string of "*.**" is appended. e.g.

```
DEV.OEM.TRSPAN       becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA%          becomes:   DEV.*.*SPA%*.**
SYS1.*.Z19           becomes:   SYS1.*.Z19*.**
```

Note that a warning message is displayed if the high level qualifier of the DSN mask is "*" (asterisk) or "**" (double asterisk). A DSN mask of this type would result in all catalogs being searched which would take some time to execute and would use a large amount of system resources.

z/VSE Systems:

On z/VSE systems, the mask is a valid CBLVCAT LISTCAT KEY parameter string. i.e. entries with file name **beginning** with the specified string or, if prefixed by "/" (slash), entries with file name **containing** the specified string. (See the CBLVCAT User Manual.)

If no DSN mask is specified, all entries will be selected.
Note that wild cards are not supported within the z/VSE DSN mask, however, "*" (asterisk) is tolerated if placed at the end of the DSN mask.

`Catalog>`

Nominate a specific catalog in which to search for the requested entry.

For **z/OS** systems, this is a catalog DSN. Specifying a catalog DSN is unneccessary if an alias exists for the DSN mask high level qualifier (HLQ) in the master catalog. In this case, the appropriate catalog DSN will automatically be inserted in this field. If the HLQ contains a wild card, then all matching aliases are interrogated, the required catalogs are searched and the last catalog searched placed in the Catalog> field.

For **z/VSE** systems, this is a disk label assigned to the VSAM catalog for which entries are to be listed.
If no catalog file label is specified, the Catalog List window displays all user catalogs cataloged in the master catalog.

Default is the master catalog.

`Types>`

Specify the catalog entry types required. Default is all types. One or more of the following types may be specified with no intervening blanks:

A	non-VSAM (or VSAM SAM) data set.
B	z/OS - Generation data group.
C	Cluster.
G	Alternate Index.
H	z/OS - Generation data set.
R	VSAM PATH.
X	Alias.
U	User catalog connector entry.
L	z/OS - Tape volume catalog library entry.
W	z/OS - Tape volume catalog volume entry.

`AllVols>`

Specify "Y" or "N" to control whether repeated display of the same entry occurs for multi-volume data sets. If "N" is specified, then only the primary volume entry is displayed.

`VSAM Data+Ix>`

Specify "Y" or "N" to control whether or not VSAM DATA and INDEX components are to be included in the list. Note that the VSAM CLUSTER entry is always displayed.

Prefix Line Commands

For **z/OS** systems, the following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command M if entry is a PDS/PDSE library, prefix line command E otherwise.
A	Open the Define Catalog Alias panel for this entry.
AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.

AS	Open an Associations list window to list associated objects for this entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FO	Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
FS	Open the File Search window for the entry.
G	For z/OS only, open the Library Member Generations List window.
I	Open a Data Set Information panel display for the entry.
IC	Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.
ID	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
K	Delete (Kill) the entry without prompting for verification.
M	If the entry is a PDS/PDSE, open a Library List window. (Default)
Q	List dataset enqueues (major name SYSDSN) for this entry.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
T	Open an Execute CBLVCAT window and issue a LISTVCAT TUNE DEFINE operation for the entry.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
Z	Perform a compress of a z/OS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
?	Open the volume statistics window for the volume containing the entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

For **z/VSE** systems, the following prefix line commands are available:

Command	Description
D	Delete the entry. User will be prompted to verify the deletion.
K	Delete (Kill) the entry without prompting for verification.
R	Rename the entry.
T	Open an Execute CBLVCAT window and issue a LISTVCAT TUNE DEFINE operation for the entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

For **z/OS** systems, the data displayed is:

Name	Type	Description
Entry	Char	Vol - Entry Name
VolX	Char	Vol - Volid, Volid+, MIGRAT1/2, *ALIAS, *VSAM or *PATH
DevC	Char	Alc - Device Class DASD|TAPE|etc
FSeq	UInt	Alc - Tape File Sequence number
T	Char	CAT - Entry Type Code
EType	Char	CAT - Entry Type blank=NONVSAM
DSType	Char	DS - Dataset type PDSE|KSDS|ESDS|RRDS|etc
UnitType	Hex	Alc - Device Unit type

UnitName	Char	Alc - Device Unit Name
DataClas	Char	SMS - Data Class
MgmtClas	Char	SMS - Management Class
StorClas	Char	SMS - Storage Class
Stripes	UInt	SMS - Stripe Count for striped datasets
Vol	Char	Vol - Volid (direct from catalog)
VSeq	UInt	Vol - Sequence number (Cat) 1 = First
VTot	UInt	Vol - Total no of volumes
VolPrime	BitFlag	Vol - Primary volume
VolCandi	BitFlag	Vol - Candidate volume
VolOFlow	BitFlag	Vol - Overflow keyrange volume
VolConv	BitFlag	Vol - Converted VSAM dataset volume
VolNVSAM	BitFlag	Vol - NonVSAM
VolKyRng	BitFlag	Vol - Keyrange qualifier exists.
VolVPCI	BitFlag	Vol - Primary VVR
VolSSQWD	BitFlag	Vol - Sequence set with data
KRQual	Char	VS - VSAM Key range qualifier

For **z/VSE** systems, the data displayed is:

Name	**Type**	**Description**
ALLOCP	Char	Defined Primary Allocation
ALLOCS	Char	Defined Secondary Allocation
ALLOCT	Char	Current Total Space Allocation
ALLOCU	Char	Unused Allocated Space
AVRL	Char	Defined Average Record Length
BLKSIZE	Char	Defined VSAM SAM Block Size
BUFSP	Char	Defined Buffer Space (BUFSP)
BUFSP/IXL	Char	Defined BUFSP or INDEX levels
CATALOG	Char	Catalog File Name
CI/CA	Char	Number of Control Intervals/Control Area
CISIZE	Char	Defined Control Interval Size
COMPONENT	Char	VSAM Object Component Name
DEFINED	Char	Date the File was Defined
DSN	Char	Fileid
ENTRY	Char	VSAM Component Entry Name
EXCPS	Char	Number of Executed Channel Programs
EXPIRES	Char	Expiry Date
FREEBYTES	Char	Free Space Bytes Value
FRSP	Char	Defined Freespace (Bytes/CI and CI/CA)
HIALLRBA	Char	Current High Allocated RBA
HIUSERBA	Char	Current High Used RBA
IMB/REP	Char	Defined Index Attributes (IMBED and/or REPLICATE)
IXL	Char	Number of INDEX Levels
KL	Char	Defined KEY Length
KL/BLK/IMB	Char	Merge KL, BLK and IMB/REP Values
LMAX	Char	Defined Maximum Record Length
NRECS	Char	Current Number of Records
NSEC	Char	Number of Allocated Extents Minus One
PCNT	Char	Calculated Amount of Used Space
PHYREC	Char	Physical Record Size allocated by VSAM
RECSTATS	Char	Number of Records Deleted, Inserted, Updated and Read
RKP	Char	Defined Relative KEY Position
S/C	Char	Defined Local Share option and the Primary Space Class
SEVL	Char	CBLVCAT's Highest Severity Level Massage Reference for the File
SHR	Char	Defined Local (Cross Region) and Cross System Share Options
SPLITCA	Char	Number of CA Splits to Date.
SPLITCI	Char	Number of CI Splits to Date.
TIMESTMP	Char	Time Stamp of VSAM object (Last Closed)

TYPE	Char	File Type
VOLUME	Char	Defined Primary Volume Serial Number

List CMS Files

See List Windows for general features and commands common to all list windows.

Supported for z/VM CMS systems only, the File List window may be opened via the following:

- Enter command FL (synonym for LC) or LD on the command line of any window.

For z/VM CMS, the **File List** window is opened in place of the z/OS or z/VSE Catalog List or Dataset List windows and displays information about files residing on accessed mini-disks.

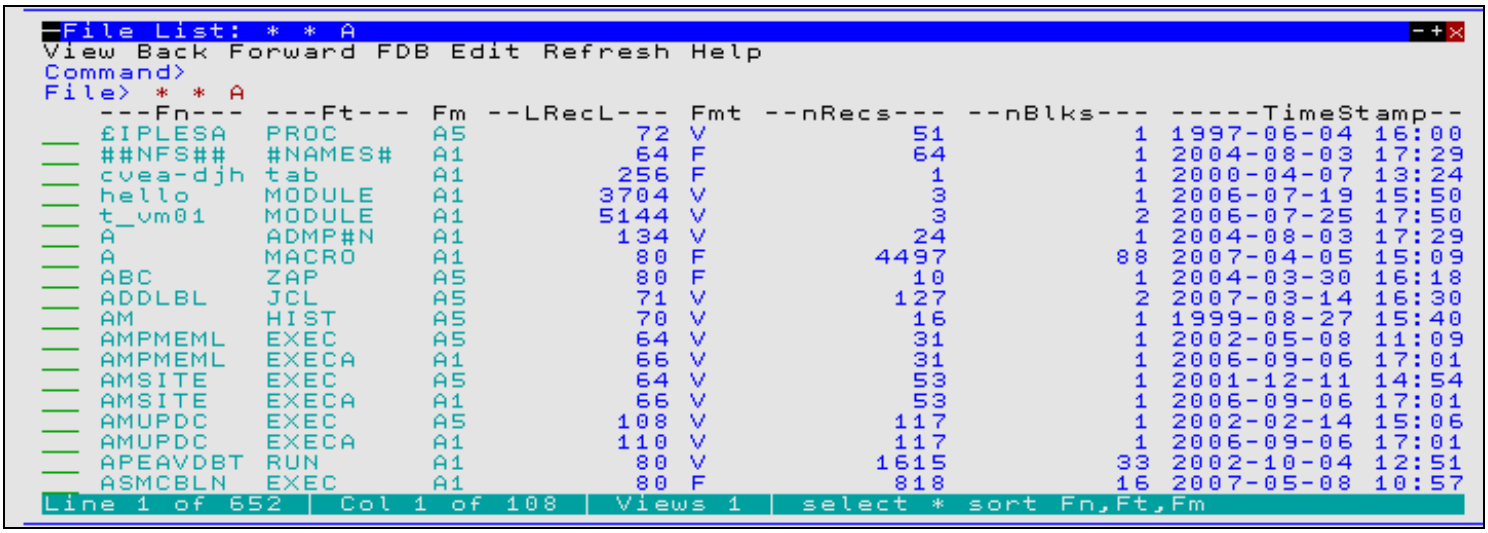


Figure 23. CMS File List window displaying all files on mini-disk A.

Panel Input Fields

File>

Specify the CMS fileid mask.

The fileid mask may consist of up to 3 qualifiers representing a filename filetype filemode combination where qualifiers are separated by one or more blanks or a "." (dot/period).

A single "*" (asterisk) wild card may be used to represent an entire qualifier or zero or more characters at a particular position within the qualifier. Wild card "*" may be specified more that once, anywhere within a qualifier.

Default filemode qualifier is "A", default filetype qualifier is "*".

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command E.
C	Copy the entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
F	Open the file search window for the PDS.
K	Delete (Kill) the entry without prompting for verification.
R	Rename the entry.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.

>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

Name	Type	Description
nBlks	UInt	Number of blocks.
nRecs	UInt	Number of records.
Entry	Char	File id.
Fm	Char	File mode.
Fmt	Char	Record format.
Fn	Char	File name.
Ft	Char	File type
Label	Char	Disk label.
LRecL	UInt	Record length.
TimeStamp	Char	Last update date and time.

List Dataset Details (=3.6)

See List Windows for general features and commands common to all list windows.

The Dataset List window may be opened via the following:

- Select option 6. 'Dataset' from the List Menu.
- Select 'Dataset Details' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LD on the command line of any window.

For CMS, the File List window is opened in place of the Catalog List or Dataset List window and displays information about files residing on accessed mini-disks. Refer to documentation on List CMS Files.

The Dataset List window is not supported on **VSE** systems.

The Dataset List window displays the basic catalog entry information together with the details of their geometry obtained either from the catalog or the VTOC for cataloged data sets.

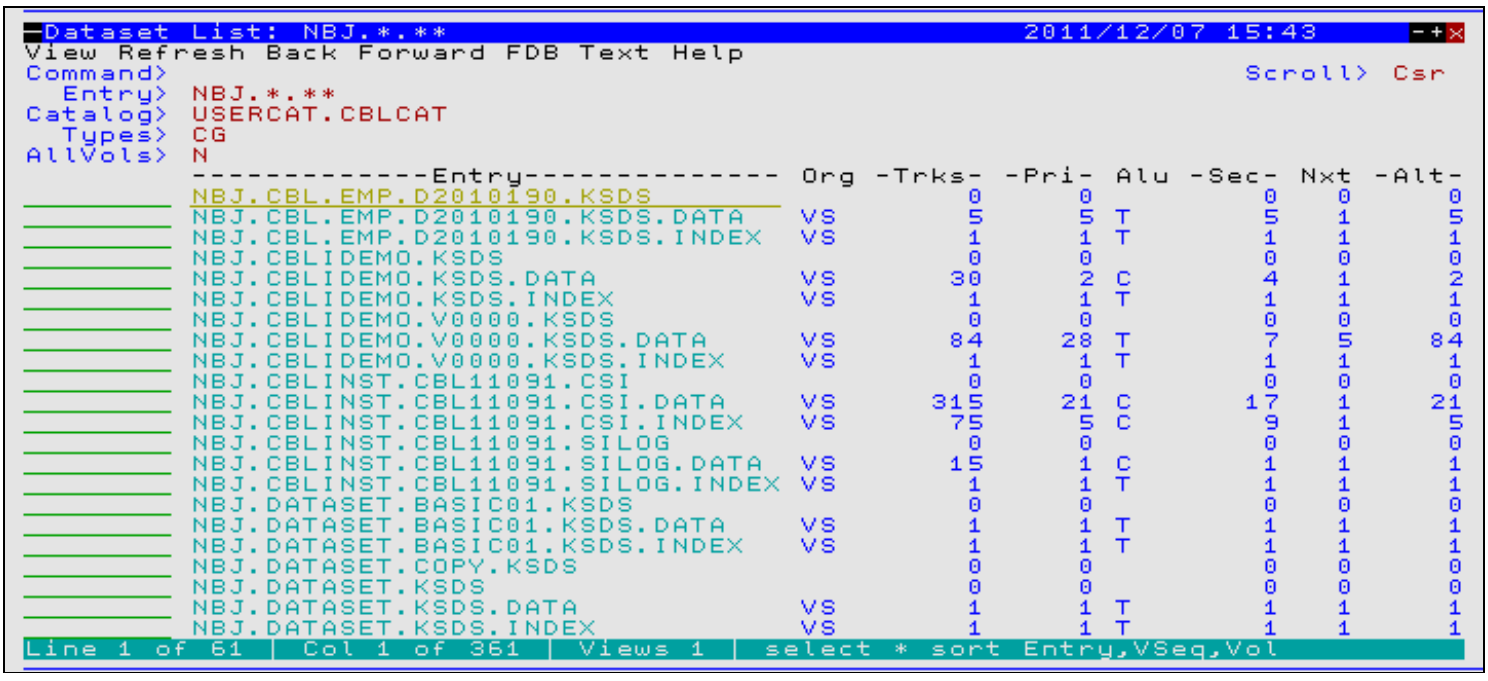


Figure 24. Dataset List window displaying all Cluster and AIX entries beginning 'CBL.'

Panel Input Fields

DSN mask>

Specify the dataset name mask which supports the following wild cards:

*	A single asterisk indicates that either a qualifier or one or more characters within a qualifier can occupy that position. An asterisk can precede or follow a set of characters.
**	A double asterisk indicates that zero or more qualifiers can occupy that position. A double asterisk cannot precede or follow any characters; it must be preceded or followed by either a dot or a blank.
%	A single percent sign indicates that exactly one character can occupy that position. (Up to 8 percent signs can be specified in each qualifier.)

If the last character of the DSN mask is "." (dot/period), then this marks the end of the low level DSN qualifier within the DSN mask. The trailing "." is stripped and no wildcard string is appended to the DSN mask. e.g.

```
DEV*.                 becomes:   DEV*
DEV.OEM.TRSPAN*.      becomes:   DEV.OEM.TRSPAN*
DEV.*.*SAMP%%.        becomes:   DEV.*.*SAMP%%
```

If the last character of the DSN mask is **not** "." (dot/period), then a default trailing wild card string is automatically appended to the DSN mask as follows:

1. If the DSN mask is a single qualifier or the last character of the DSN mask is "*" (asterisk), then a wildcard string of ".**" is appended. e.g.

```
DEV                   becomes:   DEV.**
DEV*                  becomes:   DEV*.**
DEV.OEM.TRSPAN*       becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA*           becomes:   DEV.*.*SPA*.**
```

2. Otherwise a wildcard string of "*.**" is appended. e.g.

```
DEV.OEM.TRSPAN        becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA%           becomes:   DEV.*.*SPA%*.**
SYS1.*.Z19            becomes:   SYS1.*.Z19*.**
```

Note that a warning message is displayed if the high level qualifier of the DSN mask is "*" (asterisk) or "**" (double asterisk). A DSN mask of this type would result in all catalogs being searched which would take some time to execute and would use a large amount of system resources.

Catalog>

Nominate a specific catalog in which to search for the requested DSN mask.

This is a catalog DSN. Specifying a catalog DSN is unneccessary if an alias exists for the DSN mask high level qualifier (HLQ) in the master catalog. In this case, the appropriate catalog DSN will automatically be inserted in this field. If the HLQ contains a wild card, then all matching aliases are interrogated, the required catalogs are searched and the last catalog searched placed in the Catalog> fields.

Default is the master catalog.

Types>

Specify the catalog entry types required. Default is all types. One or more of the following types may be specified with no intervening blanks:

A	non-VSAM (or VSAM SAM) data set.
B	MVS - Generation data group.
C	Cluster.
G	Alternate Index.
H	MVS - Generation data set.
R	VSAM PATH.
X	Alias.
U	User catalog connector entry.
L	MVS - Tape volume catalog library entry.
W	MVS - Tape volume catalog volume entry.

AllVols>

Specify "Y" or "N" to control whether repeated display of the same entry occurs for multi-volume data sets. If "N" is specified, then only the primary volume entry is displayed.

VSAM Data+Ix>

Specify "Y" or "N" to control whether or not VSAM DATA and INDEX components are to be included in the list. Note that the VSAM CLUSTER entry is always displayed.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command M if entry is a PDS/PDSE library, prefix line command E otherwise.
A	Open the Define Catalog Alias panel for this entry.
AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.
AS	Open an Associations list window to list associated objects for this entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FO	Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
FS	Open the File Search window for the entry.
G	For z/OS only, open the Library Member Generations List window.
I	Open a Data Set Information panel display for the entry.
IC	Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.
ID	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
K	Delete (Kill) the entry without prompting for verification.
M	If the entry is a PDS/PDSE, open a Library List window. (Default)
Q	List dataset enqueues (major name SYSDSN) for this entry.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
T	Open an Execute CBLVCAT window and issue a LISTVCAT TUNE DEFINE operation for the entry.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
Z	Perform a compress of an MVS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
?	Open the volume statistics window for the volume containing the entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

Name	Type	Description
Entry	Char	CAT - Catalog Entry Name (usually a DSN)
VolX	Char	Vol - Volid, Volid+, MIGRAT1/2, *ALIAS, *VSAM or *PATH
Org	Enum	DS - Data Set Organiz'n PS|PO|DA|VS|etc
RecFm	Enum	DS - Record format
Lrecl	UInt	DS - Logical record length
Blksz	UInt	DS - Block Size
DsKb	UInt	Alc - Data space used in Kilobytes
Trks	UInt	Alc - Total Tracks
Pri	UInt	Alc - Primary units
Alu	Char	Alc - Allocation units: C|T|B=Cyl|Trk|Blk
Sec	UInt	Alc - Secondary units
Nxt	UInt	Alc - Number of extents
Alt	UInt	Alc - Total units

PDSE	BitFlag	SMS - Partitioned dataset Extended Y|N
DsnPcu	UInt	Alc - Percent of allocated space used
Referenced	VTOCDate	Date - Last Referenced
Created	VTOCDate	Date - Created
Expires	VTOCDate	Date - Of Expiry
BlkTrk	UInt	Alc - Blocks per track
CKDKeyL	UInt	Alc - CKD Physical Key Length
RKP	UInt	Alc - CKD Relative Key Position
DevC	Char	Alc - Device Class DASD|TAPE|etc
UnitName	Char	Alc - Device Unit Name
UnitType	Hex	Alc - Device Unit type
FSeq	UInt	Alc - Tape File Sequence number
EType	Char	CAT - Entry Type blank=NONVSAM
T	Char	CAT - Entry Type Code
DSInd	Hex	DS - Dataset flags (DS1DSIND) (hex)
RACF	BitFlag	DS - Dataset is RACF defined Y|N
DSType	Char	DS - Dataset type PDSE|KSDS|ESDS|RRDS|etc
LastVol	BitFlag	DS - Last vol holding data for dset Y|N
TBal	UInt	DS - TrackBalance: Bytes free on last trk
DCOB	BitFlag	SMS - DASDM CREATE originated blksize Y|N
DataClas	Char	SMS - Data Class
XATTR	BitFlag	SMS - Extended attributes exist Y|N
XFD	BitFlag	SMS - Extended format dataset Y|N
HFS	BitFlag	SMS - HierarchicalFileSystem Y|N
MgmtClas	Char	SMS - Management Class
ReBlk	BitFlag	SMS - May be reblocked Y|N
SMi	Hex	SMS - SMS indicators (DS1SMSFG) (hex)
StorClas	Char	SMS - Storage Class
Stripes	UInt	SMS - Stripe Count for striped datasets
SMS	BitFlag	SMS - System managed dataset Y|N
InICF	BitFlag	VS - Dataset cataloged in ICF catlg Y|N
IsICF	BitFlag	VS - Dataset is an ICF catalog Y|N
VS	Hex	VS - VSAM indicators (DS1OPTCD) (hex)
KRQual	Char	VS - VSAM Key range qualifier
Vol	Char	Vol - Volid (direct from catalog)
VSeq	UInt	Vol - Sequence number (Cat) 1 = First
VTot	UInt	Vol - Total no of volumes
F1Vol	Char	Vol - Volid of 1st volume (Format 1 DSCB)
VOLSq	UInt	Vol - Sequence number (VTOC) 1 = First
VolPrime	BitFlag	Vol - Primary volume
VolCandi	BitFlag	Vol - Candidate volume
VolOFlow	BitFlag	Vol - Overflow keyrange volume
VolConv	BitFlag	Vol - Converted VSAM dataset volume
VolNVSAM	BitFlag	Vol - NonVSAM
VolKyRng	BitFlag	Vol - Keyrange qualifier exists.
VolVPCI	BitFlag	Vol - Primary VVR
VolSSQWD	BitFlag	Vol - Sequence set with data
Flag1	Hex	DS - Dataset flags (DS1FLAG1) (hex)
LastTrack	UInt	DS - Last used track
LastBlock	UInt	DS - Last used block on last used track
LastFree	UInt	DS - Space remaining on last used track
Compressable	BitFlag	DS - Compressable extended format
Checkpointed	BitFlag	DS - Checkpointed
Large	BitFlag	DS - More than 64K tracks
EAttrOpt	BitFlag	DS - Optional extended attributes specified
EAttrNo	BitFlag	DS - No extended attributes specified

List Library Members (=3.7)

See List Windows for general features and commands common to all list windows.

The Library List window may be opened via the following:

- Select option 7. 'Library' from the List Menu.
- Select 'Library Members' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LL on the command line of any window.

The Library List window displays members of a PDS/PDSE (z/OS) or LIBR (z/VSE) library.

Note that, for z/OS PDSE version 2 libraries that support member generations, a library member list will include only the base (generation 0) member entries. To display previous generations of library members, the Library Member Generations list window maust be used.

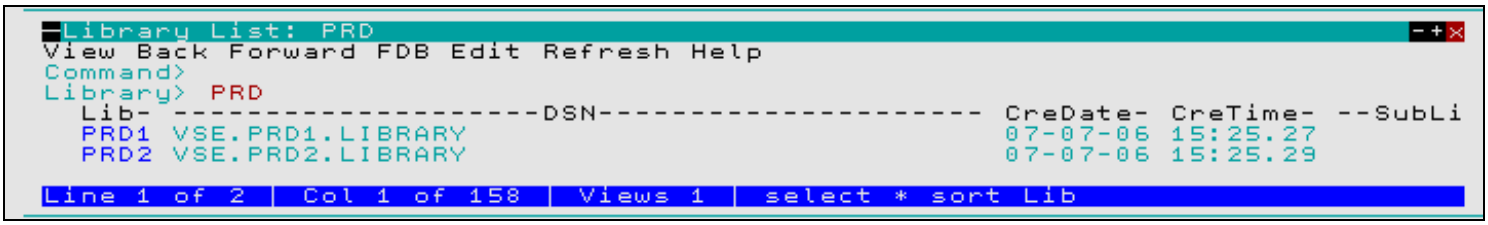


Figure 25. Library List window displaying z/VSE libraries beginning 'PRD'.

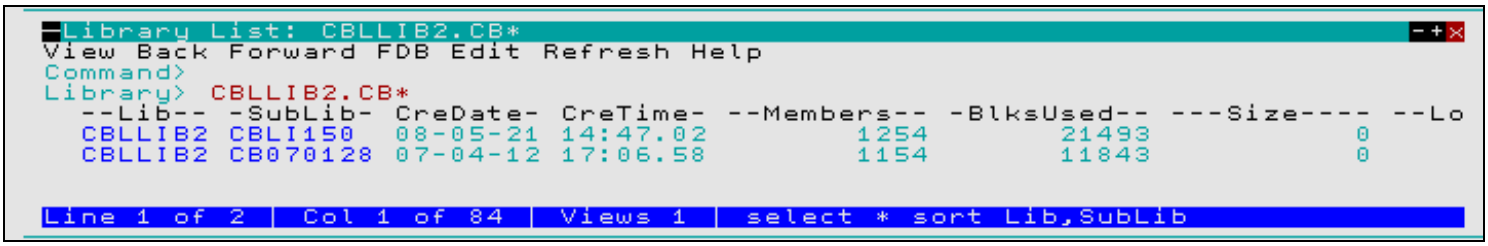


Figure 26. Library List window displaying z/VSE library 'CBLLIB2' sub-libraries beginning 'CB'.

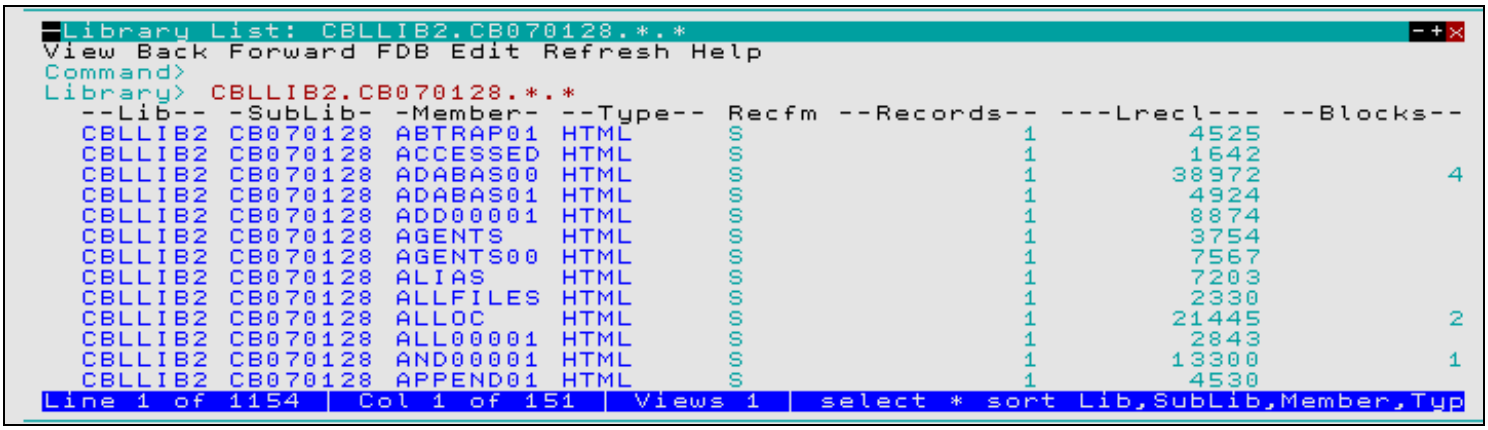


Figure 27. Library List window displaying all members of z/VSE sub-library 'CBLLIB2.CB070128'.

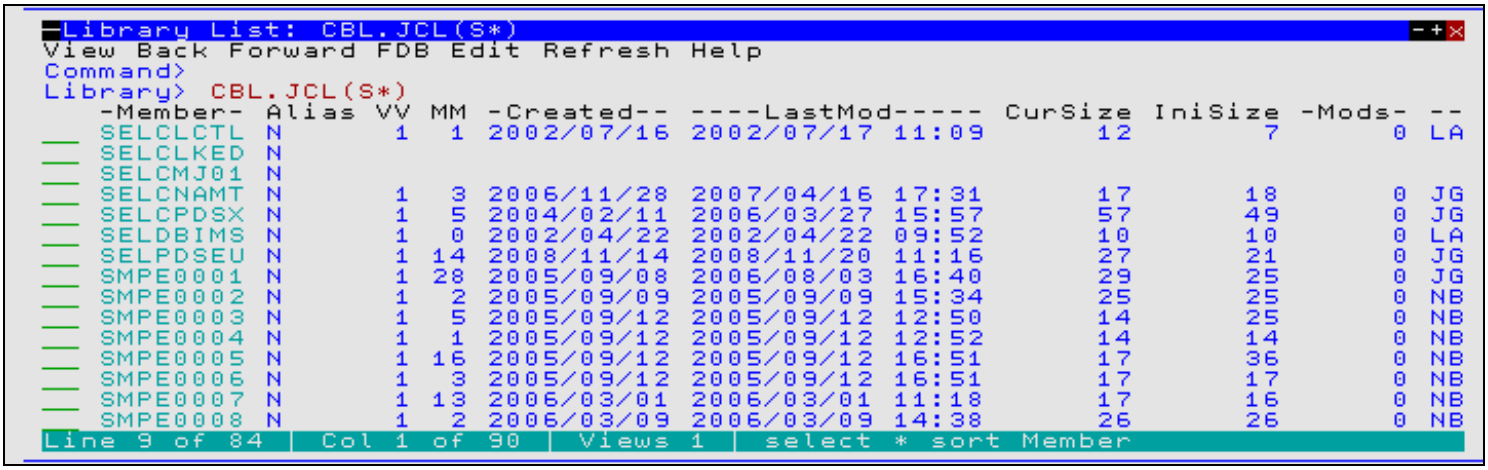


Figure 28. Library List window displaying members of z/OS PDSE 'CBL.JCL' whose names begin with "S".

Panel Input Fields

Library>
The name of the library for which the contents are to be listed.

z/OS Systems:
On z/OS systems, the library parameter is a PDS (or PDSE) dataset name and optionally one or more member name masks. A member name mask supports the following wild cards:

*	A single asterisk represents an entire member name or zero or more characters within a member name mask.
%	A single percent sign represents exactly one character within a member name mask. Up to 8 percent signs can be specified in each member name mask.

If specified, the member name mask must immediately follow the PDS(E) DSN and be enclosed in "( )" (parentheses). Multiple member name masks, all specified within the single set of parentheses, must be separated by one or more blanks and/or a "," (comma). e.g.

```
LL  DEV.OEM.CBL202.CBLI.HELP.HTML(S*AN%  WIN*, *R)
```

z/VSE Systems:
On z/VSE systems, the library parameter may be:

1. A library name. In this case the statistics for the library are listed. e.g.

   ```
   LL  CBLLIB
   ```

2. A library name and sublibrary name. In this case the sublibrary name may be a mask containing "*" (asterisk) wild cards as supported by z/VSE Librarian. The statistics for all sublibraries which fit the sublibrary name mask are listed. e.g.

   ```
   LL  CBLLIB.TEST*
   ```

3. A library name, sublibrary name and member name and type. In this case the member name and type may be a mask containing "*" (asterisk) wild cards. The statistics for all members which fit the mask are listed. e.g.

   ```
   LL  CBLLIB.TEST01.*.Z
   ```

Prefix Line Commands

For **z/OS** systems, the following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command E.
A	Open the Create Alias dialog window.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.

E	Open the CBLe text editor to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
EX	Execute the entry. (Invokes the TSO command, EXECUTE, using the entry name as input.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FC	Open the File Copy Panel to perform an advanced file copy operation (record selection, reformat records using copybooks.)
FS	Open the File Search window for the entry.
G	For z/OS only, open the Library Member Generations List window.
IC	Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.
J	Submit the entry to batch. Executes the CBLe CLI SUBMIT command using the entry name as input. (A CBLe frame window must be active for this operation to suceed.)
K	Delete (Kill) the entry without prompting for verification.
M	Move the entry.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

For **z/VSE** systems, the following prefix line commands are available when a list of z/VSE libraries or sublibraries is displayed:

Command	Description
<Dflt>	Prefix line command M.
M	Opens another Library List window containing the library/sub-library contents.
L	Lock the z/VSE LIBR member. (z/VSE Only)
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

For **z/VSE** systems, the following prefix line commands are available when a list of z/VSE sublibrary members is displayed:

Command	Description
<Dflt>	Prefix line command E.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the text editor to edit this entry.
J	Submit the entry to batch. Executes the CBLe CLI SUBMIT command using the entry name as input. (A CBLe frame window must be active for this operation to suceed.)
K	Delete (Kill) the entry without prompting for verification.
L	LOCK the member.
R	Rename the entry.
U	UNLOCK the member. A member may only be unlocked by the user that locked it.
V	Open the CBLe text editor to View (edit read/only) this entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

The data displayed for z/OS non-LOAD libraries is:

Name	Type	Description
Member	Char	Member name
Alias	BitFlag	Alias indicator
VV	Int	Version number
MM	Int	Modification level
Created	TimeDec	Creation date
LastMod	TimeDec	Last modified date and time
CurSize	Int	Current size
IniSize	Int	Initial size
Mods	Int	Modified records
User	Char	User id

The data displayed for z/OS LOAD libraries is:

Name	Type	Description
Member	Char	Member name
TTR	Hex	TTR
Rent	BitFlag	Renterable
Reus	BitFlag	Reusable
Test	BitFlag	Test module
Refr	BitFlag	Refreshable
Exec	BitFlag	Executable
Big	BitFlag	More than 16M load module
SizeHex	Hex	Contiguous storage required
EPA	Hex	Entry point address
AC	Hex	APF code
RMode	Char	Residence mode
AMode	Char	Main entry point address mode
AAmode	Char	Alias entry point address mode
AliasOf	Char	Name of aliased member
AOEPA	Hex	Entry point of aliased member
SSILvl	Hex	SSI change level
SSIFlg	Hex	SSI flag
SSISer	Hex	SSI member serial number
LMSize	Int	Large module size
LMEPA	Hex	Large module main entry point
LMAEPA	Hex	Large module alias entry point
Page	BitFlag	Page alignment required
LFmt	BitFlag	Linear format
Ovly	BitFlag	Overlay structure
Load	BitFlag	Only loadable
Scat	BitFlag	Scatter format
1Blk	BitFlag	No rld items and 1 text block
Flvl	BitFlag	Ony linkage editor F
NRLD	BitFlag	Contains no RLD items
Nrep	BitFlag	Cannot be reprocessed
TstC	BitFlag	Contains TEST cards
LnkF	BitFlag	Created by linkage editor F
Alias	BitFlag	Alias indicator

The data displayed for z/VSE libraries is:

Name	Type	Description
Lib	Char	Library file name
SubLib	Char	Sublibrary name
CreDate	Char	Creation date
CreTime	Char	Creation time
Members	Int	Number of members
BlksUsed	Int	Number of library blocks used

Size	Int	Sublibrary size limit
Locked	Int	Number of locked members

The data displayed for z/VSE sublibraries is:

Name	Type	Description
Lib	Char	Library file name
SubLib	Char	Sublibrary name
Member	Char	Member name
Type	Char	Member type
Recfm	Char	Record format
Records	Int	Number of records or bytes
Lrecl	Int	Logical record length
Blocks	Int	Number of library blocks
UpdDate	Char	Last update date
UpdTime	Char	Last update time
CreDate	Char	Creation date
CreTime	Char	Creation time
SYSIPT	BitFlag	SYSIPT data in procedure
MSHP	BitFlag	Member is MSHP controlled
MSHPByP	BitFlag	MSHP control is bypassed
PrintCC	Enum	Printer control characters
MBSTLOCK	Char	Lock identifier

List Library Member Generations

See List Windows for general features and commands common to all list windows.

The Library Member Generations List window may be opened via the following:

- Select 'Library member generations' from the Utilities/List menu in the CBLe main window menu bar.
- Execute line command "G" against a PDSE dataset entry in a list of DSN entries (e.g. in a Catalog, Dataset or VTOC List) or against an individual member name in a Library member list.
- Enter command LL on the command line of any window together with a PDSE Library Member Generation name argument.

The Library Member Generations List window displays selected generations of member entries belonging to a z/OS PDSE version 2 libarary allocated with the MAXGENS attribute.

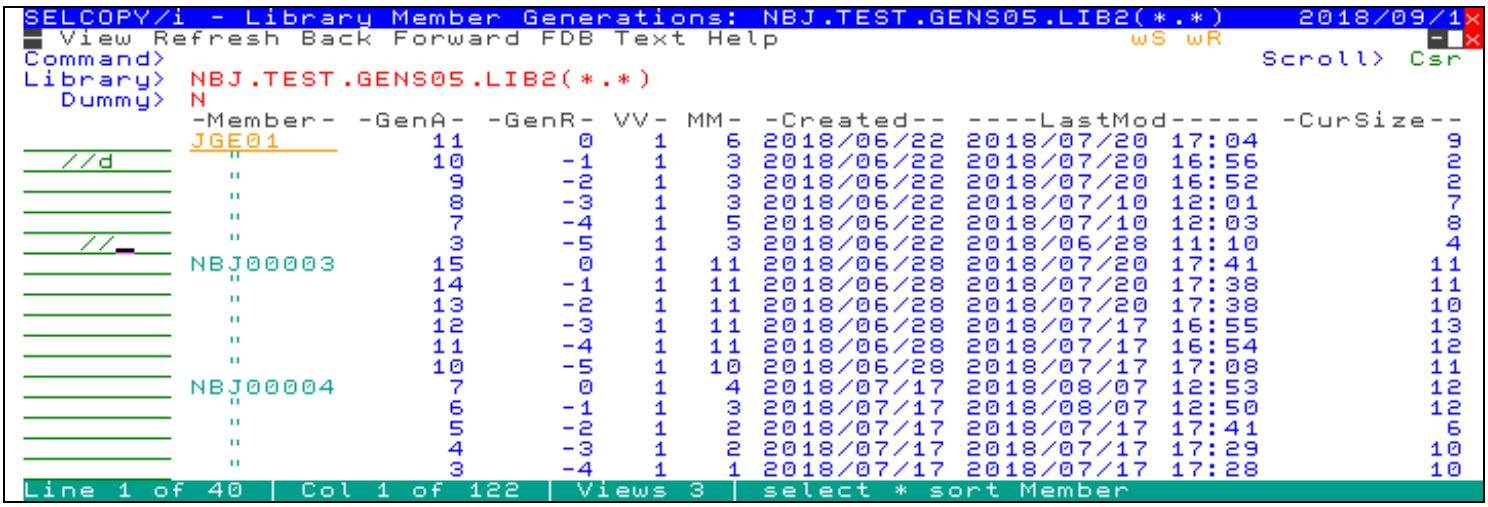


Figure 29. PDSE Library Member Generations List window.

Panel Input Fields

Library>
The name of the PDSE version 2 library for which the contents are to be listed. The library name may optionally be followed by one or more member name and generation masks in parentheses. See sections Generic File Object Names and Generic z/OS PDSE Library Member Generations for details.

Dummy>
Specify "Y" or "N" to control whether dummy entries are included in the list. Dummy entries identify the last generation number of a member for which all generations have been deleted.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command E.
B	Open the Data Editor to BROWSE the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the Text Editor to EDIT this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
J	Submit the entry to batch. Executes the CBLe CLI SUBMIT command using the entry name as input. (A CBLe frame window must be active for this operation to suceed.)
K	Delete (Kill) the entry without prompting for verification.
RC	Recover the entry, making it the prime (generation zero) member.
SD	Open the Data Editor SDE BROWSE/EDIT Dialog Window to browse or edit this entry.
V	Open the Text Editor to VIEW this entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

The data displayed for z/OS non-LOAD libraries is:

Name	Type	Description
Member	Char	Member name
GenA	Int	Absolute member generation number
GenR	Int	Relative member generation number
VV	Int	Version number
MM	Int	Modification level
Created	TimeDec	Creation date
LastMod	TimeDec	Last modified date and time
CurSize	Int	Current size
IniSize	Int	Initial size
Mods	Int	Modified records
User	Char	User id
TTR	Hex	Relative track and record number
Dummy	BitFlag	Dummy generation indicator

The data displayed for z/OS LOAD libraries is:

Name	Type	Description
Member	Char	Member name
GenA	Int	Absolute member generation number
GenR	Int	Relative member generation number
Created	STCK	Generation created UTC timestamp
LastMod	STCK	Generation UTC timestamp of last change
TTR	Hex	TTR
Rent	BitFlag	Renterable

Reus	BitFlag	Reusable
Test	BitFlag	Test module
Refr	BitFlag	Refreshable
Exec	BitFlag	Executable
Big	BitFlag	More than 16M load module
SizeHex	Hex	Contiguous storage required
EPA	Hex	Entry point address
AC	Hex	APF code
RMode	Char	Residence mode
AMode	Char	Main entry point address mode
AAmode	Char	Alias entry point address mode
AliasOf	Char	Name of aliased member
SSILvl	Hex	SSI change level
SSIFlg	Hex	SSI flag
SSISer	Hex	SSI member serial number
Page	BitFlag	Page alignment required
LFmt	BitFlag	Linear format
Ovly	BitFlag	Overlay structure
Load	BitFlag	Only loadable
Scat	BitFlag	Scatter format
1Blk	BitFlag	No rld items and 1 text block
Flvl	BitFlag	Ony linkage editor F
NRLD	BitFlag	Contains no RLD items
Nrep	BitFlag	Cannot be reprocessed
TstC	BitFlag	Contains TEST cards
LnkF	BitFlag	Created by linkage editor F
Alias	BitFlag	Alias indicator
Dummy	BitFlag	Dummy generation indicator

List MVS Allocated Files (=3.8)

See List Windows for general features and commands common to all list windows.

The Allocated Datasets may be opened via the following:

- Select option 8. 'Allocated' from the List Menu.
- Select 'Allocated Files' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LA on the command line of any window.

The resultant Allocated Datasets window displays all current DDname allocations defined to the environment running SELCOPYi.

- For **z/OS TSO**, DDnames allocated within the user's TSO region.
- For **z/OS VTAM**, DDnames allocated to the SELCOPYi VTAM application.

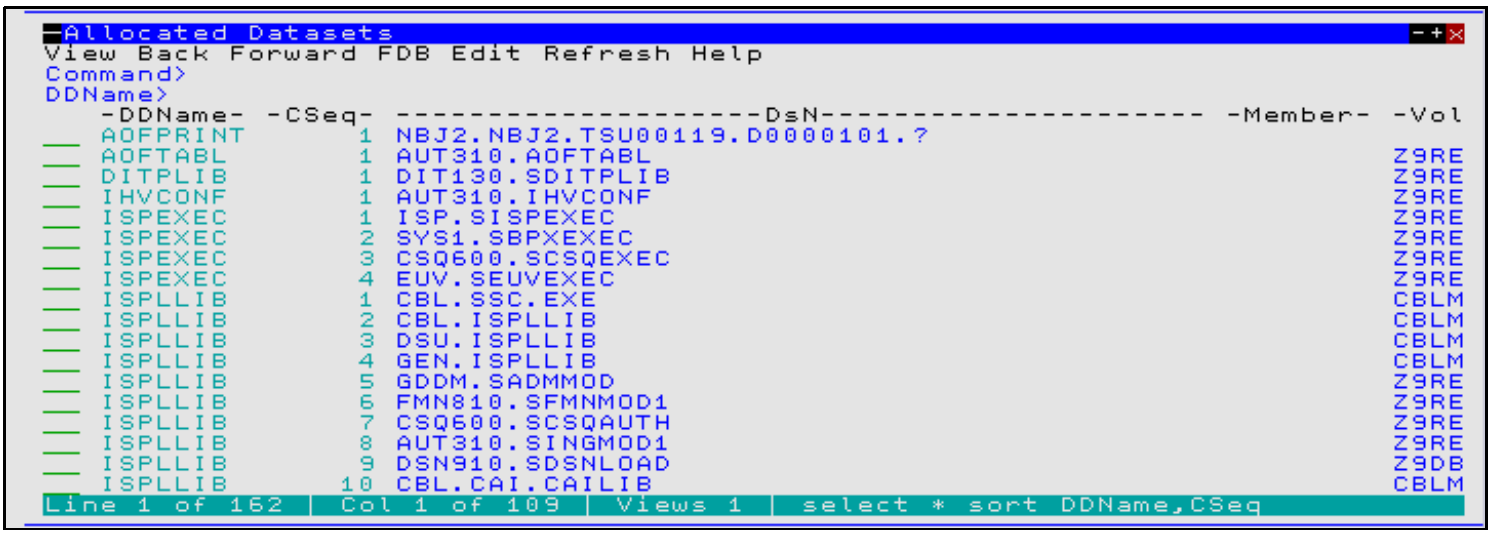


Figure 30. List MVS Allocated Datasets window.

Panel Input Fields

DDName>
Select only DDNames that match this ddname mask.

A DDName mask supports the following wild cards:

- `*` A single asterisk represents the complete z/OS MVS DDName or zero or more characters within the DDName mask.
- `%` A single percent sign represents exactly one character within the DDName mask. Up to 8 percent signs can be specified in each DDName mask.

If no wildcards are specified within the DDName mask then all MVS ddnames that **begin** with the specified DDName mask are selected.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command M if entry is a PDS/PDSE library, prefix line command E otherwise.
A	Open the Define Catalog Alias panel for this entry.
AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.
AS	Open an Associations list window to list associated objects for this entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FO	Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
FS	If the entry is a PDS(E), open the File Search window for the entry.
G	For z/OS only, open the Library Member Generations List window.
I	Open a Data Set Information panel display for the entry.
IC	Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.
ID	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
K	Delete (Kill) the entry without prompting for verification.
M	If the entry is a PDS/PDSE, open a Library List window. (Default)
Q	List dataset enqueues (major name SYSDSN) for this entry.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
T	Open an Execute CBLVCAT window and issue a LISTVCAT TUNE DEFINE operation for the entry.
U	Unallocate the DD name.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
Z	Perform a compress of an z/OS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
?	Open the volume statistics window for the volume containing the entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

Name	Type	Description
DDName	Char	DD name
CSeq	Int	Concatenmation sequence
DsN	Char	Dataset name
Mbr	Char	PDS member
Vol	Char	Volume serial number
Org	Char	Data set organisation
Recfm	Char	Record format
Lrecl	Int	Logical record length
BlkSize	Int	Block size
Disp	Char	Dataset disposition

List VSE Standard Labels

See List Windows for general features and commands common to all list windows.

The VSE Standard Label window may be opened via the following to display all permanent and temporary file labels:

- Select 'Allocated Files' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LA on the command line of any window.

Where information for individual fields are uninitialised, then the null indicator (-1) is displayed. e.g. If EXTNO field is null, then no extents have been associated with the label.

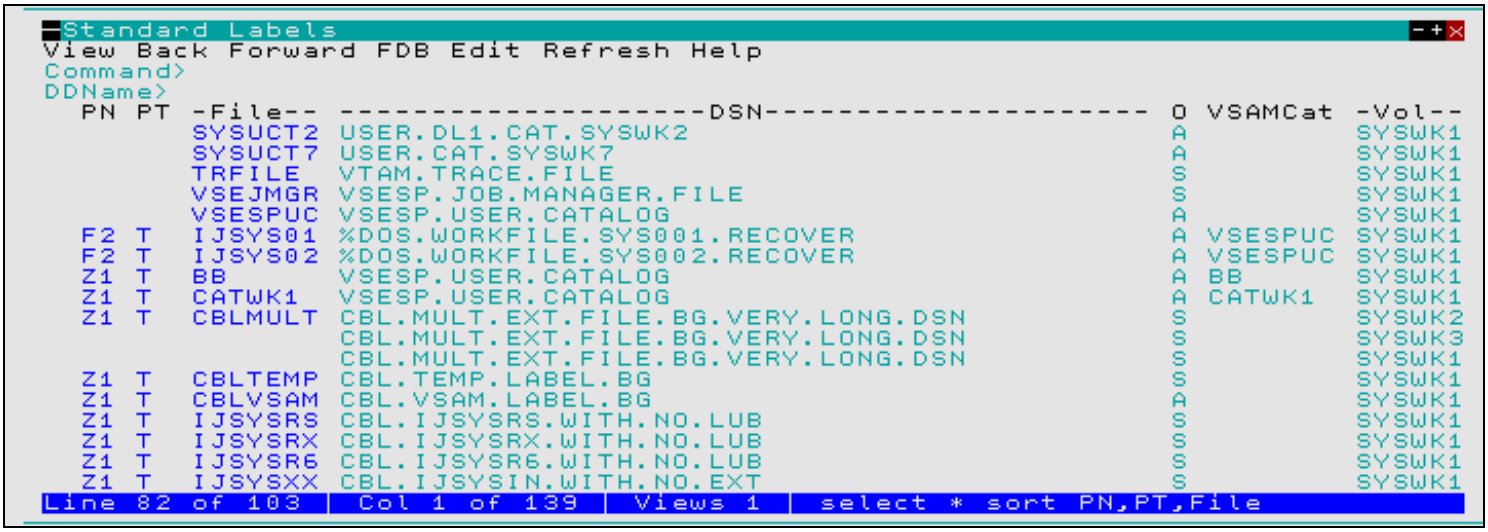


*Figure 31.*List Standard Labels Window for VSE.

Panel Input Fields

DDName>

Select only VSE label names that match this DDName mask.

A DDName mask supports the following wild cards:

*	A single asterisk represents the complete VSE label name or zero or more characters within the DDName mask.
%	A single percent sign represents exactly one character within the DDName mask. Up to 8 percent signs can be specified in each DDName mask.

If no wildcards are specified within the DDName mask then all VSE labels that **begin** with the specified DDName mask are selected.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
ID	For VSAM cataloged files only, open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
VC	For VSAM cataloged files only, open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

Name	Type	Description
ADisp	Char	Abend disposition
BLKSIZE	Int	SAM CKD DTFSD BLKSIZE override
BUFND	Int	VSAM ACB BUFND override. Number of Data buffers
BUFNI	Int	VSAM ACB BUFNI override. Number of Index buffers
BUFSP	Int	VSAM ACB and IDCAMS DEFINE BUFSP override
CISIZE	Int	SAM FBA DTFSD CISIZE override
DSN	Char	File dataset name
ExpDate	VTOCDate	Expiration date
ExtAlloc	Int	Number of allocated tracks/blocks
ExtNo	Int	Extent Sequence Number
ExtStart	Int	Start of extent (relative track/block number)
File	Char	File name
FBA	BitFlag	FBA Device Indicator for OPEN
LogUnit	Char	Assigned System or Programmer Logical Unit
O	Char	Open code for file type
ODisp	Char	Open disposition
PriAlloc	Int	VSAM/SAM RECORDS primary allocation
PN	Char	Partition name
PT	Char	Perm/Temp
RetPeriod	Int	Retention period in number of days. (Default 7)
RECSIZE	Int	VSAM/SAM Record size
SecAlloc	Int	VSAM/SAM RECORDS secondary allocation
TDisp	Char	Termination disposition
Vol	Char	Volume serial of this extent
VSAMCat	Char	VSAM catalog

List MVS Enqueues (=3.9)

See List Windows for general features and commands common to all list windows.

The Enqueue List window may be opened via the following:

- Select option 9. 'Enqueues' from the List Menu.
- Select 'Enqueues' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LQ on the command line of any window.

The Enqueue List window displays outstanding MVS enqueues by major name and minor name (queue name and resource name).

Note: Not implemented for VSE.

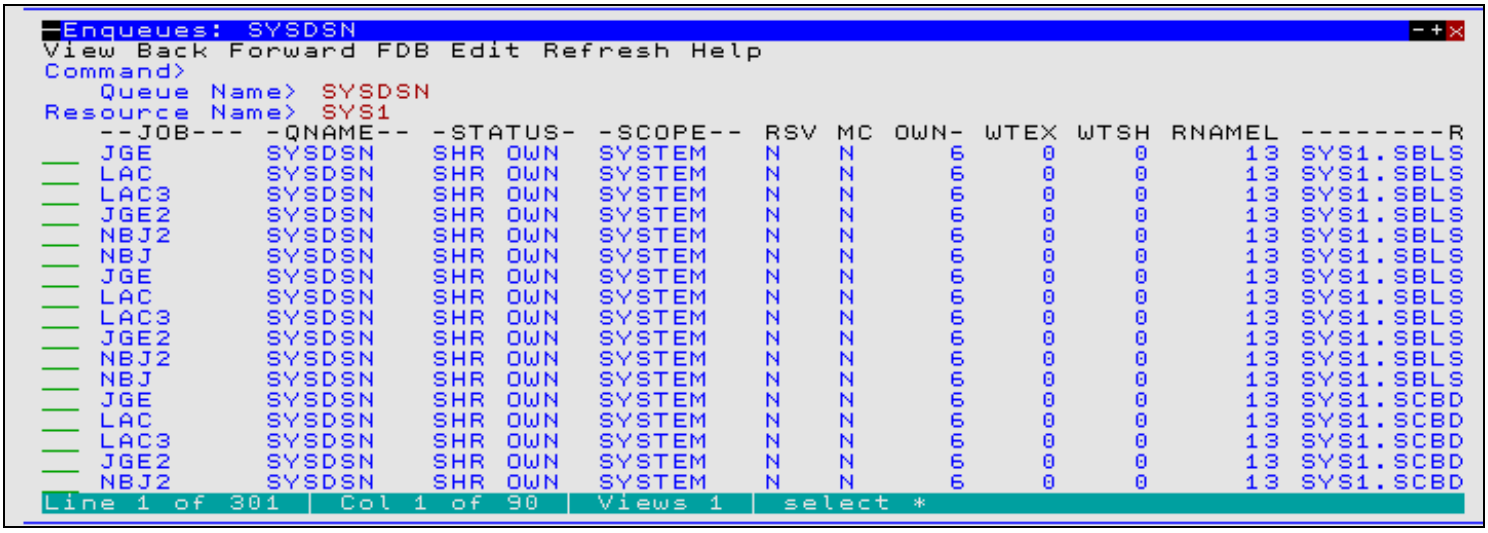


Figure 32. Enqueue List window displaying outstanding enqueues with resource name beginning 'SYS1' in queue SYSDSN.

Panel Input Fields

`Queue Name>`
: The major name (queue name) of the ENQ resource. This is a 1-8 character upper case name. For example, dataset allocations are ENQueued with resource name SYSDSN .

`Resource Name>`
: This is a 1-256 character, case sensitive minor name (resource name). You need only enter the prefix of the resources you are interested in. All resources for the given queue with resource beginning with this value are listed.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

Name	Type	Description
JOB	Char	Job name
QNAME	Char	Enqueue Major Name (Queue)
STATUS	Char	Status of Enqueue
SCOPE	Char	Scope of Enqueue
RSV	Char	Reserve
MC	Char	Must complete
OWN	Int	Number of owners
WTEX	Int	Number of waiters exclusive
WTSH	Int	Number of waiters shared
RNAMEL	Int	Rname length
RNAME	VChar	Enqueue Minor Name (Resource)

List MVS Job Enqueues (=3.10)

See List Windows for general features and commands common to all list windows.

The Job Enqueue List window may be opened via the following:

- Select option 10. 'Job Enqueues' from the List Menu.
- Select 'Job Enqueues' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LJQ on the command line of any window.

The Job Enqueue List window displays outstanding MVS enqueues held by a given job.

Note: Not implemented for VSE.

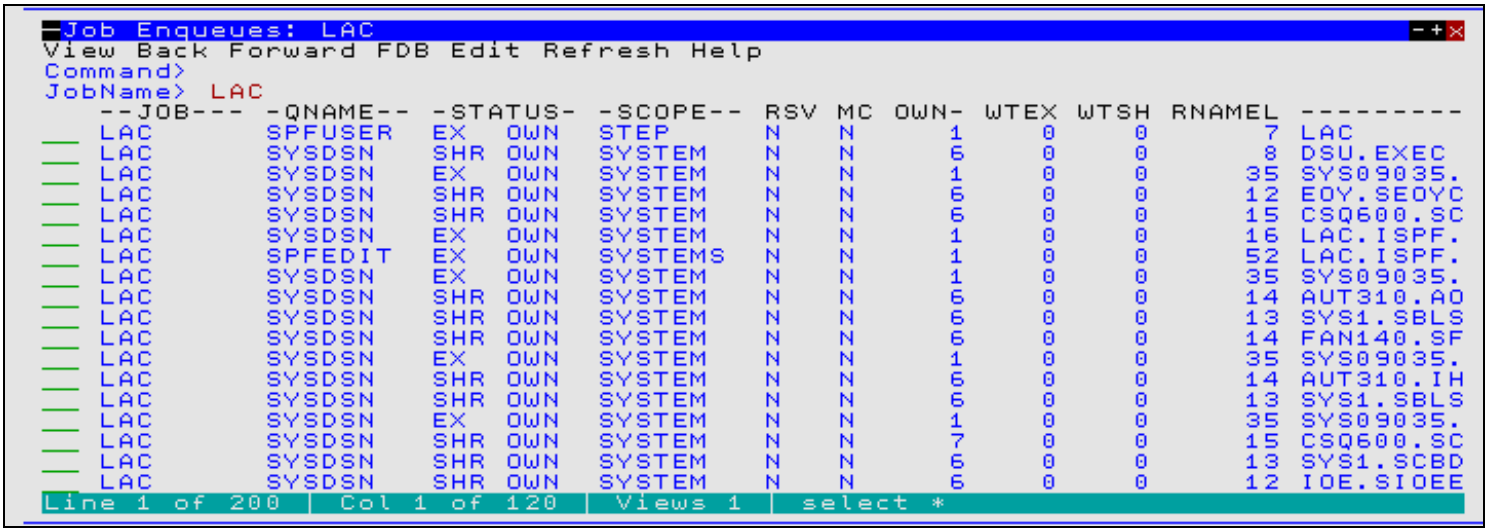


Figure 33. Job Enqueue List window displaying outstanding enqueues for Job 'LAC'.

Panel Input Fields

`JobName>`
: The name of the job for which the ENQueues are to be listed.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

Name	Type	Description
JOB	Char	Job name
QNAME	Char	Enqueue Major Name (Queue)
STATUS	Char	Status of Enqueue
SCOPE	Char	Scope of Enqueue
RSV	Char	Reserve
MC	Char	Must complete
OWN	Int	Number of owners
WTEX	Int	Number of waiters exclusive
WTSH	Int	Number of waiters shared
RNAMEL	Int	Rname length
RNAME	VChar	Enqueue Minor Name (Resource)

List Associations (=3.11)

See List Windows for general features and commands common to all list windows.

The Associations List window may be opened via the following:

- Select option 11. 'Associations' from the List Menu.
- Select 'Associations' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LAS on the command line of any window.
- Enter the prefix command AS against a data set item in a list type window.

The Associations List window displays all components associated with the selected cataloged entries. e.g. A VSAM Cluster entry may display its Data, Index and any Altenate Index objects with which it is associated.

Note: Not implemented for CMS and VSE.

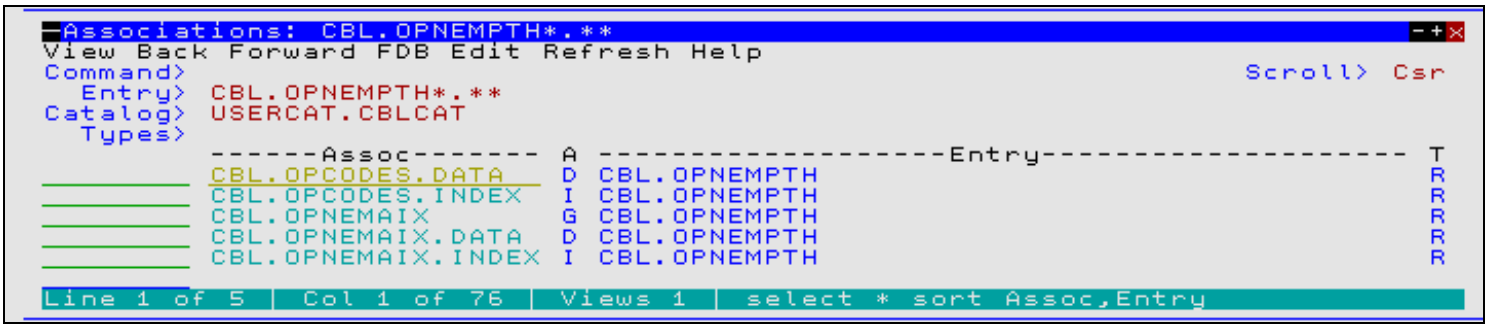


Figure 34. Associations List Window.

Panel Input Fields

Entry>

Specify the fileid mask that includes at least one cataloged object.

The fileid mask represents a DSN mask that supports the following wild cards:

*	A single asterisk represents a DSN qualifier, or zero or more characters within a DSN qualifier.
**	A double asterisk represents zero or more qualifiers within a DSN. Double asterisk must be preceded or followed by either a "." (dot/period) or a blank. It cannot precede or follow an alphanumeric character.
%	A single percent sign represents exactly one character, other than "." (dot/period), within a DSN qualifier. Up to 8 percent signs can be specified in each qualifier.

If the last character of the fileid mask is "." (dot/period), then this marks the end of the low level DSN qualifier within the fileid mask. The trailing "." is stripped and no wildcard string is appended to the fileid mask. e.g.

```
DEV*.                 becomes:   DEV*
DEV.OEM.TRSPAN*.      becomes:   DEV.OEM.TRSPAN*
DEV.*.*SAMP%%.        becomes:   DEV.*.*SAMP%%
```

If the last character of the fileid mask is **not** "." (dot/period), then a default trailing wild card string is automatically appended to the fileid mask as follows:

1. If the fileid mask is a single qualifier or the last character of the fileid mask is "*" (asterisk), then a wildcard string of ".**" is appended. e.g.

```
DEV                   becomes:   DEV.**
DEV*                  becomes:   DEV*.**
DEV.OEM.TRSPAN*       becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA*           becomes:   DEV.*.*SPA*.**
```

2. Otherwise a wildcard string of "*.**" is appended. e.g.

```
DEV.OEM.TRSPAN        becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA%           becomes:   DEV.*.*SPA%*.**
SYS1.*.Z19            becomes:   SYS1.*.Z19*.**
```

Note that a warning message is displayed if the high level qualifier of the fileid mask is "*" (asterisk) or "**" (double asterisk). A fileid mask of this type would result in all catalogs being searched which would take some time to execute and would use a large amount of system resources.

Catalog>

Nominate a specific catalog in which to search for the requested entry.

This is a catalog DSN. Specifying a catalog DSN is unneccessary if an alias exists for the fileid mask high level qualifier (HLQ) in the master catalog. In this case, the appropriate catalog DSN will automatically be inserted in this field. If the HLQ contains a wild card, then all matching aliases are interrogated, the required catalogs are searched and the DSN of the last catalog searched is placed in the Catalog> field.

Types>

Specify the catalog entry types for which associations will be reported. Default is all types. One or more of the following types may be specified with no intervening blanks:

A	non-VSAM (or VSAM SAM) data set.
B	MVS - Generation data group.
C	Cluster.
G	Alternate Index.
H	MVS - Generation data set.
R	VSAM PATH.
X	Alias.
U	User catalog connector entry.
L	MVS - Tape volume catalog library entry.
W	MVS - Tape volume catalog volume entry.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command AS.
A	Open the Define Catalog Alias panel for this entry.
AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.
AS	Open an Associations list window to list associated objects for this entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FO	Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
I	Open a Data Set Information panel display for the entry.
ID	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
K	Delete (Kill) the entry without prompting for verification.
M	If the entry is a PDS/PDSE, open a Library List window. (Default)
Q	List dataset enqueues (major name SYSDSN) for this entry.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
T	Open an Execute CBLVCAT window and issue a LISTVCAT TUNE DEFINE operation for the entry.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
Z	Perform a compress of an MVS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
?	Open the volume statistics window for the volume containing the entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

Name	Type	Description
Assoc	Char	Associated entry name
A	Char	Associated entry type code
Entry	Char	Entry name
T	Char	Entry type code

List HFS Path (=3.12)

See List Windows for general features and commands common to all list windows.

The HFS Path List window displays the contents of the specified HFS directory path and optionally its sub-directories. It may be opened via the following:

- Select option 12. 'HFS' from the List Menu.
- Select 'HFS Path Details' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LP on the command line of any window.
- Enter command LD with an HFS path argument on the command line of any window.

The HFS Path List window displays file, directory and link names contained in the specified HFS path, together with stored information for each directory entry.

Note: List HFS Path is not supported for CMS or VSE.

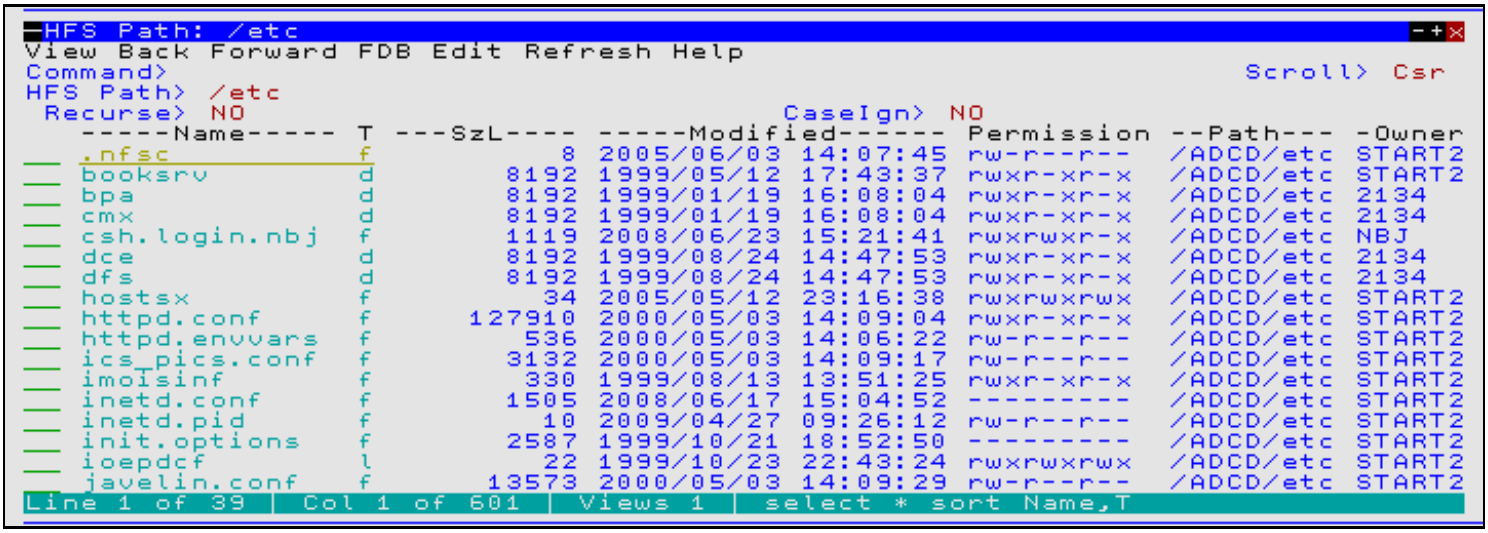


Figure 35. HFS Path List window.

Panel Input Fields

HFS Path>
Specify the absolute or relative HFS path name.

The name portion of the HFS path is the character string at the end of the path that follows the last "/" (slash) of the fileid, or is the entire path name if "/" is not specified.
The following wild cards may only be specified within the name portion of the HFS path.

*	A single asterisk represents zero or more characters.
%	A single percent sign represents a single character.

Recurse>
Enter "YES" to recursively list the contents of all sub-directories found within the HFS path specification.
Default is "NO".

CaseIgn>
Enter "YES" to bypass case sensitivity for the name portion of the specified HFS path.
Default is "NO".

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Default action depends on the list entry as follows: • For a directory entry or a symbolic link to a directory, open a new List HFS list window to display the the contents of the directory. • For all other entries, a Text Editor view is opened to edit the data (as for prefix command "E").
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
D	Delete the entry (file, link or directory). User will be prompted to verify the deletion.
E	Open a CBLe text editor view to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	Open the **FSU - File Search/Update Window** to perform an advanced search and optionally update the contents of the entry.
K	Delete (Kill) the entry without prompting for verification.
R	Rename the entry.
SD	Open the **SDE BROWSE/EDIT Dialog Window** to browse or edit the entry's data within a Structured Data Environment **window view**.
UT	Opens the general file utilities menu to generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

Name	Type	Description
Name	ALPair	Filename
T	Enum	Dir entry type
Size	UInt	Bytes Used
Modified	Date	Data Modified Timestamp
Permission	Char	Permissions as displayed by the LS shell command.
Path	ALPair	Path
Owner	Char	Owner name
Group	Char	Group name
Fmt	Enum	File Format
Links	Int	Number of links
Mode	Int	HFS Mode (nnn)
UidX	BitFlag	Set user ID on execution
GrpX	BitFlag	Set group ID on execution
Sticky	BitFlag	Sticky Bit
INode	Hex	File Serial Number (INode)
Dev	Hex	Device ID
DevMaj	Hex	Major Device number
DevMin	Hex	Minor Device number
Uid	Int	Owner ID
Gid	Int	Group ID
Changed	Date	File Stat Chg Timestamp
Accessed	Date	Last Accessed Timestamp
Created	Date	File Creation Timestamp
BlkSz	UInt	File Block Size
AuditId	Char	RACF File ID for auditing
AA1	Hex	Auditor audit byte 1
AA2	Hex	Auditor audit byte 2
AA3	Hex	Auditor audit byte 3
AA4	Hex	Auditor audit byte 4

UA1	Hex	User audit byte 1
UA2	Hex	User audit byte 2
UA3	Hex	User audit byte 3
UA4	Hex	User audit byte 4
rU	BitFlag	Read permission for User(Owner)
wU	BitFlag	Write permission for User(Owner)
xU	BitFlag	Exec permission for User(Owner)
rG	BitFlag	Read permission for Group
wG	BitFlag	Write permission for Group
xG	BitFlag	Exec permission for Group
rO	BitFlag	Read permission for Others
wO	BitFlag	Write permission for Others
xO	BitFlag	Exec permission for Others
NoDel	BitFlag	Files should not be deleted
ShrLib	BitFlag	Shared Library
NoShrs	BitFlag	No shareas flag
Auth	BitFlag	APF authorized flag
PgmC	BitFlag	Program controlled flag
ExtLink	BitFlag	External Symbolic Link
NoDelM	BitFlag	(Mask) Files should not be deleted
ShrLibM	BitFlag	(Mask) Shared Library
NoShrsM	BitFlag	(Mask) No shareas flag
AuthM	BitFlag	(Mask) APF authorized flag
PgmCM	BitFlag	(Mask) Program controlled flag
ExtLinkM	BitFlag	(Mask) External Symbolic Link
AclAccess	BitFlag	Access ACL exists
AclFModel	BitFlag	File Model ACL exists
AclDModel	BitFlag	Directory Model ACL exists
Set	Hex	Flag bytes 1-4
FTag	Char	File Tag
BlksH	UInt	Blocks Allocated (High Order)
BlksL	UInt	Blocks Allocated (Low Order)
Opq	Hex	Opaque attribute flags
OpqM	Hex	(Mask) Opaque attribute flags
M1	Hex	HFS Mode byte 1
M2	Hex	HFS Mode byte 2
M3	Hex	HFS Mode byte 3
RefT	UInt	Reference Time
Id	Hex	File Identifier
CTime	UInt	Ctime Micro_Seconds
SecLabel	Char	Security Label
Res	Char	Reserved
Res1	UInt	Reserved
Res2	Char	Reserved
Res3	Char	Reserved

List Storage Groups

See List Windows for general features and commands common to all list windows.

The Storage Groups List window displays the defined System Managed Storage (SMS) Storage Groups. It may be opened via the following:

- Select option 14. 'StorGrps' from the List Menu.
- Select 'SMS Storage Groups' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LSG on the command line of any window.

The Storage Groups List window displays the name, description and other information relating to the Storage Group definition.

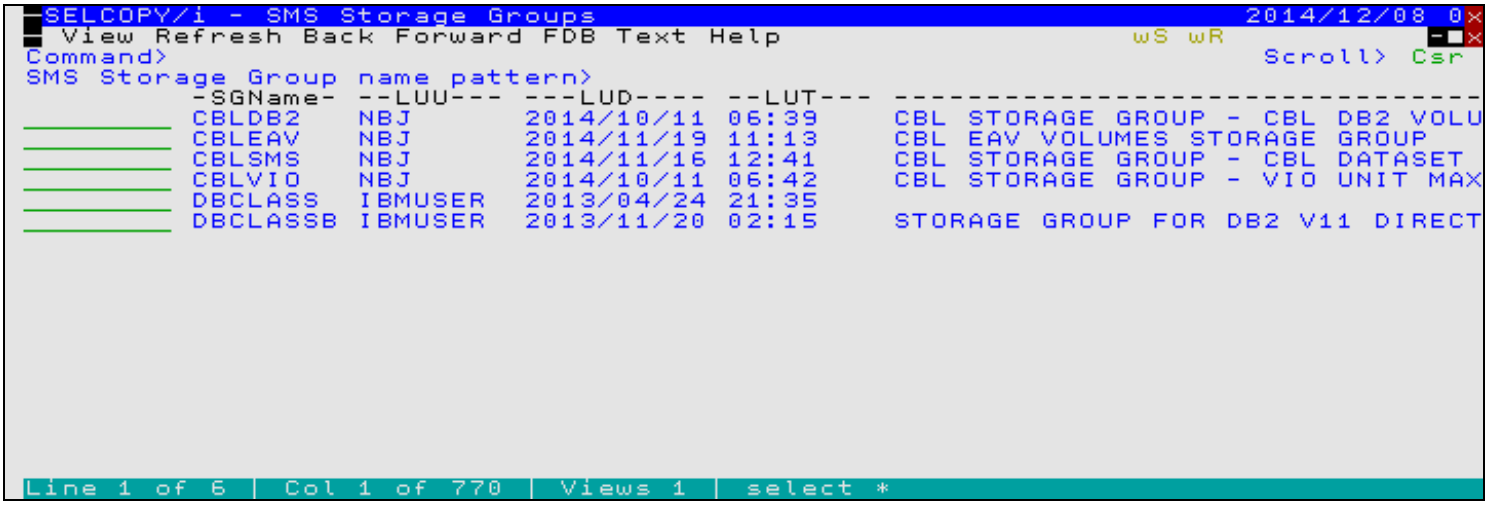


Figure 36. Storage Groups List window.

Panel Input Fields

`SMS Storage Group name pattern>`
Specify the SMS storage group. The following wild cards may only be specified.

*	A single asterisk represents zero or more characters.
%	A single percent sign represents a single character.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Display the list of volumes assigned to the storage group. (Equivalent to prefix command "V")
V	Open the SMS storage Group Volumes list to display the list of volumes assigned to the storage group.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

Name	Type	Description
StorGrp	VChar	Storage group name
LUU	Char	Last update user
LUD	Char	Last update date
LUT	Char	Last update time
Description	Char	Description of storage group
Flags	Hex	Flags
AutoBackup	BitFlag	HSM auto backup
AutoMigrate	BitFlag	HSM auto migration
AutoDump	BitFlag	HSM auto dump
Thresholds	BitFlag	Thresholds specified
GBackup	BitFlag	Guaranteed backup specified
GBNoLimit	BitFlag	Guaranteed backup no limit
IntMigrate	BitFlag	Interval migration
PrimeSpace	BitFlag	Primary space AM
SGType	Enum	Storage group type
Flags2	Hex	Flags2
OFlowSG	BitFlag	Overflow storage group specified
ExtendSG	BitFlag	Extend storage group specified
TargetSG	BitFlag	Target copy storage group specified

BPSpec	BitFlag	Breakpoint value specified
TASpec	BitFlag	Track allocation threshold specified
PPriSpec	BitFlag	Processing priority specified
Overflow	Hex	Overflow
VIOMax	Int	VIO maximum dataset size
VIOUnit	Char	VIO unit type
HiThresh	Int	High threshold percent
LoThresh	Int	Low threshold percent
DumpClass1	Char	Auto dump class 1
DumpClass2	Char	Auto dump class 2
DumpClass3	Char	Auto dump class 3
DumpClass4	Char	Auto dump class 4
DumpClass5	Char	Auto dump class 5
ABackUpSys	Char	Auto backup system
ADumpSys	Char	Auto dump system
AMigSys	Char	Auto migrate system
PPri	Int	Processing priority
GBackUPFreq	Int	Guaranteed backup frequency
OAMTSId	Char	OAM table space id
OAMFlags	Hex	OAM flags
OAMCycle	BitFlag	OAM cycle start and end given
VolFull	BitFlag	Volume full threshold bit
DriveStart	BitFlag	Drive start threshold bit
VFWriteErr	BitFlag	Volume full write error given
VFWriteErrBit	BitFlag	Volume full write error bit
OAMRetProt	BitFlag	OAM retention protection
OAMDelProt	BitFlag	OAM deletion protection
OAMCStart	Int	OAM cycle start time (hours)
OAMCEnd	Int	OAM cycle end time (hours)
VolFull	Int	Volume full threshold
DriveStart	Int	Drive start threshold
OLibrary1	VChar	Library name 1
OLibrary2	VChar	Library name 2
OLibrary3	VChar	Library name 3
OLibrary4	VChar	Library name 4
OLibrary5	VChar	Library name 5
OLibrary6	VChar	Library name 6
OLibrary7	VChar	Library name 7
OLibrary8	VChar	Library name 8
SysDataOff	Int	Offset to system data
SysDataLen	Int	Length of system data
OSMCSys	Char	OSMC system name
ExtendSG	VChar	Extend storage group name
TargetSG	VChar	Target storage group name
BreakPoint	Int	Break point value
HiTAThresh	Int	Track allocation high threshold
LoTAThresh	Int	Track allocation low threshold

List Storage Group Volumes

See List Windows for general features and commands common to all list windows.

The Storage Group Volumes List window displays the volumes assigned to a specific System Managed Storage (SMS) Storage Group. It may be opened via the following:

- Select option 15. 'StorGrps Vols' from the List Menu.
- Select 'SMS StorGrp Vols' from the Utilities/List menu in the CBLe main window menu bar.
- Enter command LSGV on the command line of any window.

The Storage Group Volumes List window displays the attributes of DASD volumes defined to a Pool type SMS storage group.

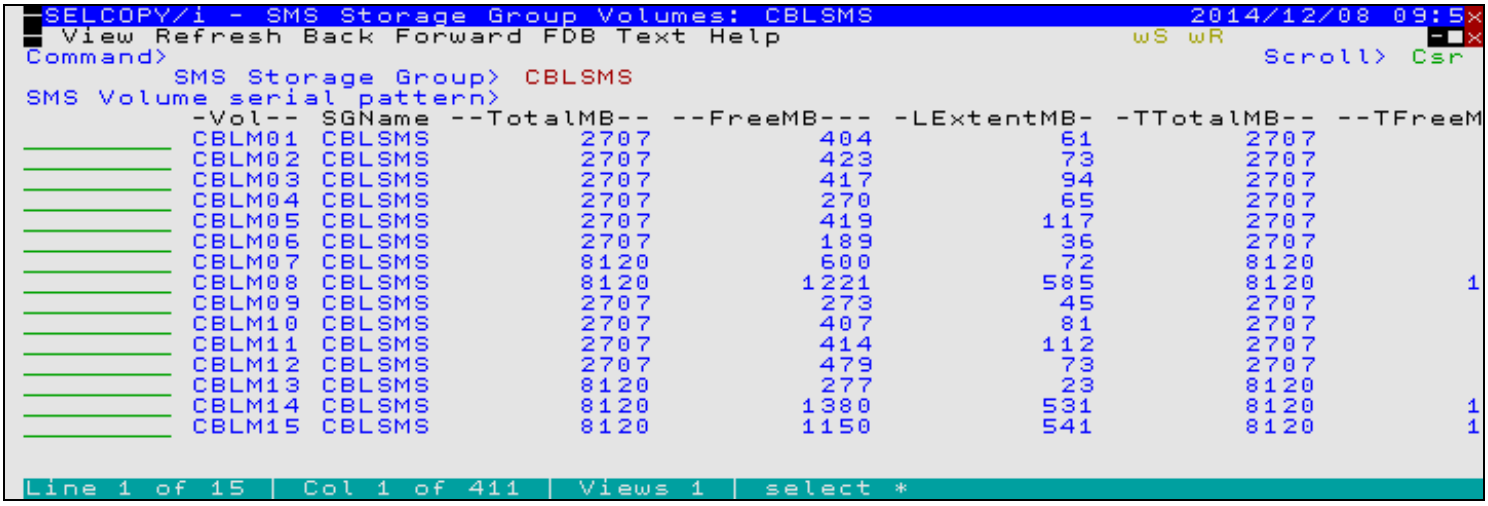


Figure 37. Storage Group Volumes List window.

Panel Input Fields

SMS Storage Group>
Specifies the name of the SMS pool storage group for which volumes are displayed.

SMS Volume serial pattern>
Specifies a volume id mask. The mask supports the following wild cards:

*	A single asterisk represents zero or more characters.
%	A single percent sign represents a single character.

By default, a volume id mask that is less than 6 characters in length and does not contain an * (asterisk) wild card will be treated as having an implied trailing * wild card.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix Line command T.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
T	Open the VTOC list window for the volume.
VC	Open an Execute CBLVCAT window and issue a LISTVTOC operation for the entry.
I	Open the volume statistics window for the volume containing the file.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

Name	Type	Description
Vol	Char	Volume serial number
StorGrp	VChar	Storage group name
TotalMB	Int	Total capacity in magabytes
FreeMB	Int	Amount free in magabytes
LExtentMB	Int	Largest free extent in magabytes
TTotalMB	Int	Track managed total capacity in magabytes
TFreeMB	Int	Track managed amount free in magabytes
TLExtentMB	Int	Track managed largest free extent in magabytes
TrackSize	UInt	Volume R1 track capacity

TotalTracks	Int	Total tracks on volume
FreeExtents	Int	Number of free extents
TotalFreeCyl	Int	Total free cylinders
TotalFreeTrk	Int	Total additional free tracks
LExtentCyl	Int	Largest free extent cylinders
LExtentTrk	Int	Largest free extent additional tracks
FragIndex	Int	Fragmentation index
TotalTTracks	Int	Total tracks in track managed space
TFreeExtents	Int	Track managed free extents
TTotalFreeCyl	Int	Track managed total free cylinders
TTotalFreeTrk	Int	Track managed total additional free tracks
TLExtentCyl	Int	Track managed largest free extent cylinders
TLExtentTrk	Int	Track managed largest free extent additional tracks
TFragIndex	Int	Track managed fragmentation index
FreeVRCount	Int	Free VTOC record count
FreeVIRCount	Int	Free VTOC index record count
UCBPtr	Hex	UCB pointer
LUU	Char	Last update user
LUD	Char	Last update date
LUT	Char	Last update time
Flags	Hex	Flags
ResetCount	Int	Volume level reset count
SGStatus	Hex	Storage group status
ULevel	Int	Update level for volume
FLUsed	Int	Full volume last percent used
TLUsed	Int	Track volume last percent used
RetStat	Hex	LSPACE Return status flags
VolStat	Hex	LSPACE Volume status flags

File Copy (=5)

Overview

File Copy (FCOPY) is an advanced copy utility supporting copy and optional remap of records between 2 files of potentially different data set organisations and geometry (RECFM, LRECL, BLKSIZE).

Features include:

- Use of structures to remap fields in source records to fields of the same name in destination records. Structures may be specified as an SDE structure, COBOL or PL1 copybook or a COBOL or PL1 ADATA file.
- Specification of a start record and/or a number of records to be copied so defining a subset of records to be copied/remapped.
- Append to or overwrite records in an existing target data set.
- Choose a pad character to be used to pad short records that are copied to longer fixed format records (e.g. copying an ESDS to RECFM=F; RECFM=V to RRDS or RECFM=F LRECL=80 to RECFM=F LRECL=100). Default pad character is blank (X'40).

File Copy supports copy of multiple PDS/PDSE library members to another new or existing library (Library Copy). This type of copy/remap is performed if the source file is a PDS/PDSE library, specified with or without a member mask, and the target file is a PDS/PDSE library with no member name specified. Note that a target PDS/PDSE library DSN with no member name is valid only for library copy.

File Copy:

Where the target file is not a PDS/PDSE library member, file copy supports copy of multiple PDS/PDSE library members to a single target file.

File Copy invokes the File Search/Update/Copy/Remap Utility (FSU) to copy or remap records and, on completion, returns the following summary message:

```
ZZSD356I FCOPY Summary: COPY - n records of m files.  x Remap Errors. y I/O Errors.
```

Library Copy:

Library copy will use the IEBCOPY facility whenever possible to perform the copy operation. It will also include copy of any member name aliases, whether or not the alias name matches the supplied member name mask. Similarly, members will be copied if their alias name matches the member name mask but the member name does not. i.e. the member name group will be copied.

Where use of IEBCOPY is not possible (e.g. libraries are of different geometry or fields in member records are to be remapped), the File Search/Update/Copy/Remap utility is invoked to copy or remap the records.

On completion of a library copy, one of the following is displayed:

- If IEBCOPY was used to perform the copy, the following informational message is returned:

  ```
  ZZSD344I FCOPY: (IEBCOPY) IGW01550I n OF m SPECIFIED MEMBERS WERE COPIED
  ```

  Furthermore, the Execute IEBCOPY window is displayed by default to report all IEBCOPY messages generated by the operation.

- If the SELCOPY/i File Search/Update/Copy/Remap utility, then, in addition to the ZZSD356I FCOPY Summary message, the following informational message is returned:

  ```
  ZZSD333I FCOPY: Members Copied=w, Replaced=x, Not Copied=y, Errors=z.
  ```

  Furthermore, the FSU - PDS Copy Statistics list window is displayed by default to report members copied and truncation/remap status of each member's records.

Source and Target File Types

The File Copy utility can copy records between any of the following file types in a single execution:

- Cataloged or uncataloged sequential (including multi-volume) datasets.
- Partitioned dataset (PDS/PDSE) members.
- GDG datasets.
- VSAM (KSDS, ESDS, RRDS, VRDS).
- HFS Files.

File Copy Panel

File Copy

The **File Copy** panel is displayed when the File Copy utility is started interactively.

This panel allows the user to invoke the File Copy utility to copy, and optionally remap, records from files matching a specified fileid mask to a single output file or multiple members of a PDS/PDSE library. The output file may potentially be of different data set organisation and geometry to the files identified by the input fileid mask.
See *"File Copy Utility"* for an overview of functionality.

The File Copy panel window may be started via the following:

- Select option 5. 'Copy/Reformat' in the SELCOPY/i Primary option menu or select option 8. 'Copy' in the Create New Datasets Menu panel.
- Select 'File Copy' from the Utilities menu.
- Execute the command FCOPY from the command line of any window.
- Execute the prefix command "**C**" from a file List type window. The resulting File Copy panel window will treat the corresponding list entry as the "From DSN" field entry.

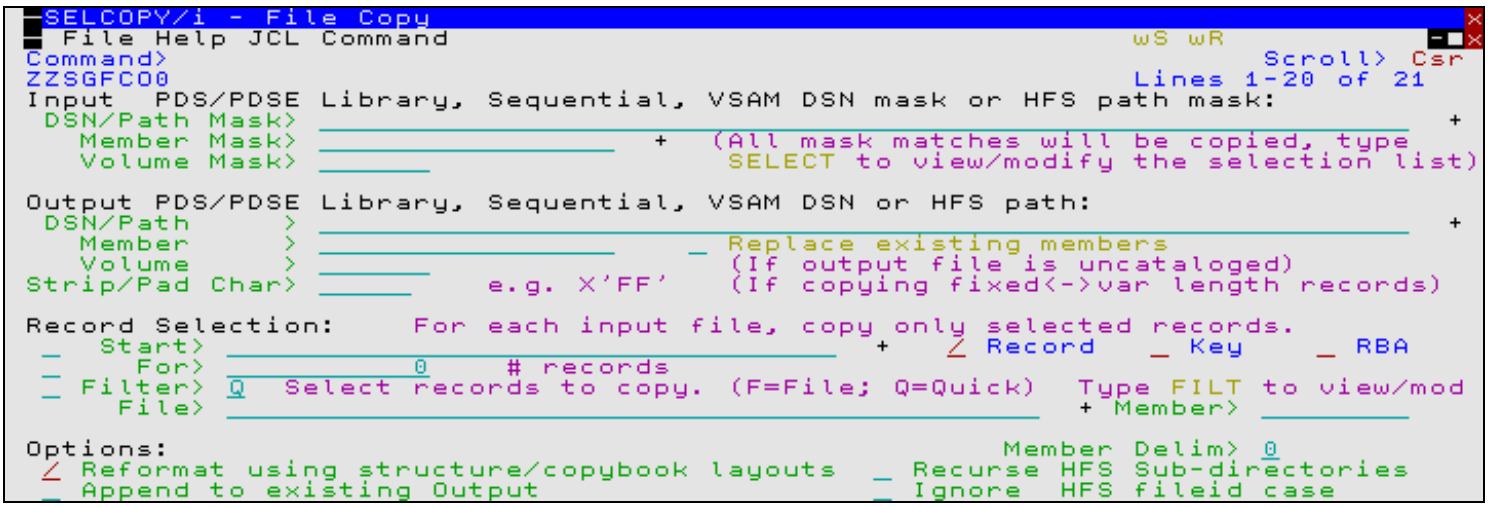


Figure 38. File Copy Panel View.

By default, field entries are populated with arguments and options that were entered the last time the utility panels were used.

Dialog option fields may be selected or de-selected by entering a non-blank or blank character respectively.

Depending on whether the **Reformat using structure/copybook layouts** option has been set or the output data set already exists, pressing the <Enter> key or, if configured, double-clicking the left mouse button, will will either verify the input fields and action the file copy, display the File Reformat panel view, or open prompts to allocate the new output dataset

Alternatively, the user may select an item from the menu bar.

On pressing the <Enter> key or selecting "Copy" or "JCL" menu items, a check is made to determine whether the target and source files exist.

- If the **source** file does not exist, the user is prompted to re-enter a different fileid.
- If the **target** file does not exist and is not an HFS file, the user is prompted to identify the DSORG of the new data set (NONVSAM, KSDS, ESDS or RRDS) before being presented with the Allocate NonVSAM or Define VSAM KSDS/ESDS/RRDS/LDS dialog window as appropriate.

  If the selected DSORG is equivalent to that of the source file, then the new data set dialog window will contain values modelled on the source file. Even where the DSORG is different, dialog Record Length fields will contain an appropriate value determined from the source file.

Menu Bar Items

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values entered in the panel so that they may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

Command
Generate the FCOPY command line syntax for field entries specified by the user and display it in a temporary CMX file text edit view. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

JCL
Generate a JCL job stream that executes the **SDEAMAIN** program with input (SDEIN) containing the FCOPY command generated for the specified panel field values.

The job stream is displayed in a temporary text edit view and may be submitted to batch using the SUBMIT command.

Panel Input Fields

PDS/PDSE Library, Sequential, VSAM DSN mask or HFS path mask:
Fields which together constitiute a fileid mask that identifies one or more files to be copied.

The utility supports use of a fileid mask and will processes all HFS files, sequential, GDG, VSAM and PDS/PDSE data sets that match the mask.

If the fileid mask contains "*" (asterisk) or "%" (percent) wild card characters, then the Select Files to Process panel will be displayed which contains a list of selectable fileids that match the specified fileid mask. Furthermore, for PDS/PDSE library entries, the Select Input Members panel may then be opened to select from a list of matching member names or member generations belonging to that library. <PF5> (SELECT) can be used to redisplay these panels after they have been closed.

Having displayed the Select Files to Process panel, a list of the selected, individual fileids will be passed to the utilty. Furthermore, if only a single fileid has been selected for processing, then the selected file DSN/Path name will replace the generic entry in the **DSN/Path Mask>** field and, if applicable, the selected volume id and member name/generation will replace the generic entries in the **Volume Mask>** and **Member Mask>** fields respectively.

DSN/Path Mask>
An unquoted entry which identifies DSN mask, a completed Fileid mask, a DDName mask or an HFS file path mask. An entry in this input field is mandatory.

- **HFS file path mask**
  Identified by the presence of "." (dot/period) as the first character or "/" (slash) anywhere within the mask, an HFS file path mask may be absolute or relative to the current HFS working directory. See USS PWD in Unix System Services (USS) Commands. "./" should prefix the mask if files in the current HFS directory are to be selected.

  Wild card characters "%" (percent), representing a single characters, and "*" (asterisk), representing zero or more characters, are supported in the name portion of the HFS file path mask. The name portion of the HFS path is the character string at the end of the path that follows the last "/" (slash) of the fileid or is the entire path name if "/" is not specified.

  If an HFS file path mask is specified, the contents of the **Volume Mask>** and **Member Mask>** fields will be ignored and **HFS Options>** settings respected.

- **DDName mask**
  If the value entered is not an HFS file path mask, is of length less than or equal to 8 characters and does not include "." (dot/period), and no Volume mask has been specified, then the value is considered to be a DDName mask. A DDName mask may be used to process non-HFS files that are currently allocated to DDNames that match the DDName mask. (e.g. SYSEXEC)

  Wild card characters "%" (percent), representing a single characters, and "*" (asterisk), representing zero or more characters, are supported in a DDName mask.

  If a DDName mask is specified, the enties in the **Member Mask>** field may still be used to select matching member names belonging to PDS/PDSE library data sets allocated to matching DDNames. **HFS Options>** settings are ignored.

- **DDName Library Concatenation**
  If the value entered is a DDName mask with no wildcard characters but with an ampersand (&) prefix, then the value is considered to be a DDName library concatenation path. A DDName library concatenation is used to process the first occurrence of a member name found within the path of libraries.

  If a DDName library concatenation is specified, the enties in the **Member Mask>** field may be used to select matching member names found within the library data set concatenation.

  Note that members will be processed in alphabetical order of member name.

- **Completed Fileid mask**
  If the value entered is not an HFS file path mask but includes a volume mask specification and/or a member mask specification, then the value is considered to be a completed Fileid mask.

A completed fileid mask is a DSN mask with a volume mask and/or one or more PDS/PDSE member name masks expressed in the following format:

```
{volmask:}data.set.name.mask{( membmask{ {,} membmask...} )}
```

Specification of one or more member masks between a single pair of "( )" (parentheses) will restrict processing to only PDS/PDSE library data sets. Multiple PDS/PDSE member masks must be separated by a "," (commma) and/or one or more intervening blanks.

Specification of a 1 to 6 character volume mask prefix followed by ":" (colon) restricts processing to only cataloged or uncataloged data sets that have extents on matching volume ids.

The fileid mask supports wild card characters as described for Volume Mask> and Member Mask> fields and for a DSN Mask below.

Fileid Mask Examples:

```
PE1.DEV.SRC.COBOL.CRKSW00(*)
SYS6.JNP*.**
OEM.TEST%%.**.CBLI.**(BOX*,D%T*,*ALL)
Z9RES1:ADCD.**
Z9RES*:ADCD.Z19.P%%%LIB(*)
```

- **DSN mask**
  If the value entered does not meet the criteria described above for an HFS file path mask, completed Fileid mask or a DDName mask, then the value is considered to be a DSN mask.

  The DSN mask is joined with any member and volume mask specified in the Volume Mask> and Member Mask> fields respectively to identify a single fileid mask.

  The following wild card characters are supported within a DSN Mask:

  `*` A single asterisk represents a DSN qualifier or zero or more characters within a DSN qualifier.
  e.g. `DEV.CBLINS.*.JCL`, `DEV.CBLINS.TEST*.ISP*LIB`, `DEV.CBLINS.*.*`

  `**` Double asterisk represents zero or more qualifiers within a DSN. Double asterisk may only be preceded or followed by the qualifier separator, "." (dot/period).
  e.g. `DEV.CBLINS.**`, `DEV.CBLINS.**.CBLE`

  `%` A single percent sign represents exactly one character within a DSN qualifier.
  e.g. `DEV.CBLINS.TEST0%.JCL`, `DEV.CBLI%%.TEST06.CBLI.%%%`

  Note that a TSO prefix is **not** applied to a DSN mask.

Volume Mask>
Optionally specify a volume name mask of maximum length 6-bytes.

Processing will be restricted to only those cataloged or uncataloged data sets that match the DSN mask **and** also have extents that exist on a volume that matches the volume mask.

The volume mask supports wild card characters as follow:

`*` A single asterisk represents a complete volume name or zero or more characters within a volume name.
e.g. `CBL*`, `*RES*`

`%` A single percent sign represents exactly one character within the volume mask.
e.g. `Z9DB9%`, `%%XV3%`

This field entry is ignored if the DSN/Path Mask> field does not contain a DSN mask.

Member Mask>
Optionally specify one or more PDS/PDSE member name masks separated by a "," (comma) and/or one or more intervening blanks.
e.g. `BLOCK, PROFILE  BOXSEQ`

If a member mask is specified, then only PDS/PDSE libraries that match the fileid mask will be selected for processing. Non-PDS/PDSE library data sets will be excluded.

If a member mask is **not** specified, then all files that match the fileid mask will be selected for processing and a default member name mask of "*" will apply to all PDS/PDSE libraries included in this selection.

Processing will be restricted to only those PDS/PDSE data sets that match the DSN mask **and** only members with a member name that matches any one of the supplied member masks.

A member mask supports wild card characters as follow:

`*` A single asterisk represents an entire member name or zero or more characters within a member name.
e.g. `CBL*5`, `BOX*`, `D*T*`

`%` A single percent sign represents exactly one character within a member name mask.
e.g. `H%, D%R*, E%A`

This field entry is ignored if the DSN/Path Mask> field does not contain a DSN mask or a DDNAme mask.

If the DSN Mask identifies a PDSE version 2 library supporting member generations (a MAXGENS value > 0), then each member mask may include a generation mask to select specific generations of members whose name matches the member mask.
e.g. `X12345.JCL(SSOPEN22.*, SSCLOSE*.>=-10)`

`Output PDS/PDSE Library, Sequential, VSAM DSN or HFS path:`
Input fields which together identify a single output sequential, VSAM or PDS/PDSE library data set, HFS file or PDS/PDSE library member.

If the output file is a PDS/PDSE library data set with no member specified, then library copy will occur. In this case, only input PDS/PDSE library members are selected for processing and any files selected by the input mask that are not library members are ignored. All selected members are copied to members of the same name in the output library.

`DSN/Path>`
Identifies the fully qualified data set name or an absolute or relative HFS file path.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

If a DSN is specified for a data set that does not already exist, a prompt data set dialog will be opened to allocate the new output file.

`Member>`
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

`Volume>`
Specifies the name of the output data set volume. This is required only if output is to an uncataloged data set.

`Replace existing members`
This is an option field which affects library copy only. If selected, any existing member in the output library that has the same name as a member to be copied from an input library, will be overwritten. In the case where members of the same name are copied from multiple input libaries, then the output library will contain the last member copied.

`Strip/Pad Char>`
Specifies a single byte pad or strip character implemeted as follows:

1. When copying fixed length records to variable length, contiguous trailing characters at the end of each record that match the specified character *char* are to be stripped.
   Default is not to strip trailing characters.

2. When copying variable length records to fixed length, records are to be padded with the specified character *char*.
   Default is to pad with the blank character (X'40').

The strip/pad character *char* may be specified in character, hexadecimal or binary string notation of length 1 byte. (e.g. 'A', C'a', X'40', B'11110001'.)

`Record Selection:`
Fields which together identify criteria by which subsets of records from every input file are selected for processing.

`Start>`
If activated, the **Start>** field specifies the first record in every file matching the fileid mask at which processing will start. Records occurring sequentially before the start record will be bypassed. If this field is not activated, records are selected beginning at record 1.

This input field may contain a record number, an RBA number (for ESDS input only), or a key string (for KSDS input only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

`Record | Key | RBA`
Identifies the type of start value specified in the **Start>** field. Enter "/" in the appropriate, mutually exclusive parameter field.

`For>`
If activated, then for each file matching the fileid mask, the **For>** field specifies the maximum number of records within that file for which processing may occur. If this field is not activated, records are selected beginning at start record and ending at the last record in the file.

`Filter>`
If activated, the **Filter>** field specifies options to either generate a new record filter or use an existing record filter file. A record filter will perform further subsetting on input file records selected for processing by the Start> and/or

For> input fields.

Filter options are as follow:

- Q On pressing <PF6>, the Quick Filter dialog panel will be opened in order to generate a temporary filter on the record data. To define a filter using formatted field names, an input copybook/structure must have first been indicating using the "Specify layout(s) for Reformat or Filter" option below.
- F Use a permanent filter identified by the sequential data set or member name in the File> field.

  On pressing <PF6>, the Create File Filter dialog panel will be opened in order to display the contents of an existing filter file or create and save a new filter file.

If option "F" is selected, then specifiaction of a filter fileid is mandatory.

File>
Identifies the fully qualified data set name of a sequential data set or PDS/PDSE library containing a record filter. Quotes are unnecessary but permitted.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the File> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Options:

Member Delim> 0|1|2
Applicable only when copying members from a PDS/PDSE library to a single output file e.g. an HFS, VSAM, or sequential dataset. This option causes a delimiter record to be written before the data for each member copied.

Enter blank to select from a list describing the available options, the format of the delimiter record depending on the option selected.

Option	Delimiter Record Format
0	No delimiter record (Default).
1	"./ ADD NAME=<member>"
2	"*>>>>>> DSN=lib.name(member) <<<<<<*"

Specify layout(s) for Reformat or Filter
Select this option only if an input and output structure is to be applied to record data in order to remap of formatted record fields as the records are copied. Note that a structure may be a COBOL Copybook, PL1 Copybook, COBOL or PL1 ADATA file or SELCOPY/i SDO.

If selected, the File Reformat panel view will open on pressing <Enter>.

Append to existing Output
Applicable to copy to a single output data set only (i.e. not library copy), select this option if the copied records are to be appended to existing records in the output data set.

Recurse HFS Sub-directories
Set this option on to process matching files within all directories and sub-directories beneath the absolute or relative directory specified by an HFS file path mask

Ignore HFS fileid case
Set this option on to disable case sensitivity when matching HFS file names with the file **name** portion of the specified HFS file path mask. The name portion of the HFS file path is the character string following the last "/" (slash) of the fileid mask or the entire HFS file path if no "/" is included in the fileid mask.

File Reformat

The **File Reformat** panel view is displayed when option **"Specify layout(s) for Reformat or Filter"** is selected in the File Copy panel.

This panel is used to specify the input and output structures, both of which are mandatory for formatted record field remap.

However, only the input structure need be specified to enable record selection using a formatted filter.

A formatted record field remap operation requires a match-up process on the input and output structures for both Record-types, and Fields within those record-types.

Record-type and Field names that are identical in both the input and output structure are automatically matched.

Type the MAP primary command to interactively match-up Record-types and Field that are not identically named.

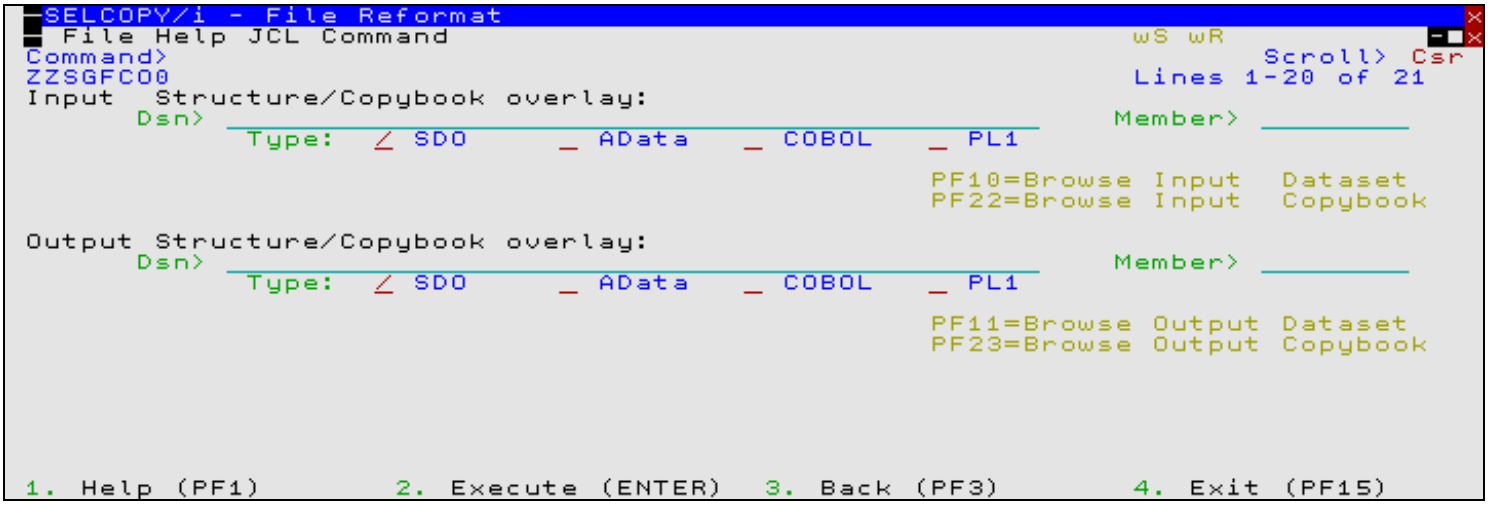


Figure 39. File Reformat Panel View.

Menu Bar Items

Menu bar items are as described for File Copy .

Panel Input Fields

`Input Structure/Copybook overlay:`
Fields which together specify a cataloged structure file (COBOL or PL1 Copybook, ADATA file or a SELCOPY/i SDO) used to map input records. The structure may be a sequential data set or a PDS/PDSE library member.

`Dsn>`
Identifies the fully qualified data set name of a sequential data set or PDS/PDSE library containing a COBOL or PL1 Copybook, ADATA file or SDO structure.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

`Member>`
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

`Type:`
Indicate the type of structure (COBOL, PL1, ADATA or SDO).

`Recompile>`
If *Structure/Copybook overlay* refers to a COBOL, PL1 or ADATA source file (not an SDO) then a compile step must be performed in order to turn the source copybook into SELCOPYi's own internal structure (SDO) format. A temporary SDO will be created, lasting the duration of the SELCOPYi session, making subsequent reference to the same copybook during the session much faster.

If, however, the copybook (or any of its included components) is modified during the session, then a recompile of the source will normally be required.

For performance reasons **N** (meaning no recompile should occur) is the default.

Specify **Y** to force a copybook recompile each time the process is run. Alternatively type the primary command: **SD DROP *<copybook_name>***

`Output Structure/Copybook overlay:`
Fields which together specify a cataloged structure file used to map output records. These fields are as described for the Input Structure/Copybook overlay.

Input record data mapped by fields in the input structure is remapped to potentially different positions and data types in the output record. These output record field positions and data types are determined by this output structure.

Primary Commands

The following primary commands are supported by both the File Copy and File Reformat panel views.

BROWSEINPUTCOPYBOOK

```
>>--+- BROWSEINPUTCOPYBOOK ------+-------------------------------------><
    +- BIC ----------------------+
```

Browse the input structure file specified by the Input Structure/Copybook overlay fields.

Once displayed, GO EDIT may be used to convert the BROWSE view to a text edit view and so allow updates to the structure source. If the source is updated, the input structure Recompile option should be set to ensure that a new copy of the structure is loaded/generated when actioning the file reformat utility.

BROWSEINPUTCOPYBOOK is assigned to <F22> by default.

BROWSEINPUTFILE

```
>>--+- BROWSEINPUTFILE ----------+-------------------------------------><
    +- BIF ----------------------+
    +- VWINP --------------------+
```

Browse the file represented by the Input PDS/PDSE Library, Sequential, VSAM DSN mask or HFS path mask fields. The file is considered to be an HFS file path if the combination of these fields describes a fileid mask containing wild card characters.

If Specify layout(s) for Reformat or Filter has been selected, then BROWSEINPUTFILE will attempt to browse the input file using the input structure specified by the Input Structure/Copybook overlay fields.

BROWSEINPUTFILE is assigned to <F19> by default.

BROWSEOUTPUTCOPYBOOK

```
>>--+- BROWSEOUTPUTCOPYBOOK -----+-------------------------------------><
    +- BOC ----------------------+
```

Browse the output structure file specified by the Output Structure/Copybook overlay fields.

Once displayed, GO EDIT may be used to convert the BROWSE view to a text edit view and so allow updates to the structure source. If the source is updated, the output structure Recompile option should be set to ensure that a new copy of the structure is loaded/generated when actioning the file reformat utility.

BROWSEOUTPUTCOPYBOOK is assigned to <F23> by default.

BROWSEOUTPUTFILE

```
>>--+- BROWSEOUTPUTFILE ---------+-------------------------------------><
    +- BOF ----------------------+
    +- VWOUT --------------------+
```

Browse the file represented by the Output PDS/PDSE Library, Sequential, VSAM DSN mask or HFS path fields.

If Specify layout(s) for Reformat or Filter has been selected, then BROWSEOUTPUTFILE will attempt to browse the output file using the output structure specified by the Output Structure/Copybook overlay fields.

BROWSEOUTPUTFILE is assigned to <F20> by default.

CMX

```
>>--+- CMX ----------------------+-------------------------------------><
    +- EDITCMX ------------------+
```

Generate command syntax. Same as menu bar item, **Command**.

CMX is assigned to <F17> by default.

FILTER

```
>>--+- FILTER --------------------+-------------------------------------><
    +- FILT ----------------------+
```

Depending on the value entered in the Filter Type field ("Q" or "F"), FILTER attempts to display either the Quick Filter Selection Criteria panel or the Create File Filter panel for the file represented by the Filter fields.

FILTER is assigned to <F6> by default.

MAP

```
>>--+- MAP ---------------------------------------------------------><
```

Interactively match-up Record-types and Field that are not identically named for use in a file reformat operation.

Record-types and Field that are identically named are matched automatically.

JCL

```
>>--+- JCL ----------------------+-------------------------------------><
    +- EDITJCL -------------------+
```

Generate a batch JCL and command syntax. Same as menu bar item, **JCL**.

JCL is assigned to <F18> by default.

RUN

```
>>--+- RUN ----------------------+-------------------------------------><
    +- EXECSYNTAX ----------------+
```

Verify input fields and execute the file copy utility. This is the default action on pressing <Enter> in the File Reformat panel view or in the File Copy panel view when the reformat option has not been selected.

SELECT

```
>>--+- SELECT -------------------+-------------------------------------><
    +- SEL ----------------------+
```

SELECT will open a sub-panel displaying all files tha match the fileid mask represented by the Input PDS/PDSE Library, Sequential, VSAM DSN mask or HFS path mask fields.

Specific files and library members may then be selected for processing by the file copy utility.

SELECT is assigned to <F5> by default.

File Copy - PDS Copy Statistics

Summary Format

The PDS Copy Statistics window is displayed following execution of the File Copy (FCOPY) utility if a Library Copy operation is performed that invokes the File Search/Update/Copy/Remap Utility. If the Library Copy invokes IEBCOPY, the Execute IEBCOPY window is displayed instead to report all IEBCOPY messages generated by the operation.

If FCOPY is executed in batch (using program SDEAMAIN), then the PDS copy statistics and all other messages are written to SDEPRINT when the job is submitted. If executed in the SELCOPY/i foreground, the PDS copy statistics are displayed in a list window (window class LISTFRAM).

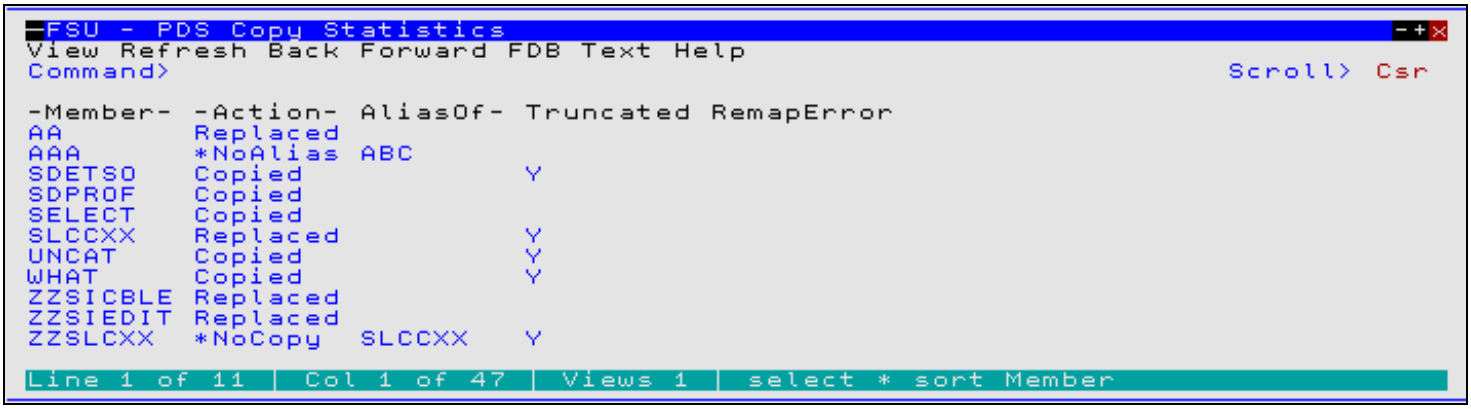


Figure 40. PDS Copy Statistics.

Copy Statistics Fields

`Member`
: Library member or alias name from the source library that has been selected for copy to the target library.

`GenA/GenR`
: The absolute (GenA) and relative (GenR) generation values for the member generation.

Columns displayed only if the source is a PDSE library version 2 which supports member generations (MAXGEN value defined) and at least one member generation or member generation mask has been specified on the input fileid mask.

`Action`
: Displays the action taken during the copy of the member or alias. Possible actions are as follow:

Copied
: Member or alias was successfully copied.

Replaced
: Applicable only if the REPLACE option has been specified, indicates that the copy of a member has successfully replaced a member or alias of the same name in the target library. If the source library member is an alias, then this action will occur only if it replaces an alias in the target library belonging to the same member group.

***NoRepl**
: Applicable only if the REPLACE option has **not** been specified, this action indicates that the member or alias name entry cannot be copied as a member or alias of the same name already exists in the target library.

***NoCopy**
: Indicates that one of the following has occurred:

1. The REPLACE option has **not** been specified and, although this member or alias name does not exist in the target library, it has failed to copy since at least one entry belonging to the same member group already exists in the target library (Action *NoRepl).

2. The REPLACE option has been specified, however at least one entry belonging to the this entry's member group exists in the target library as an entry of another member group.

***NoAlias**
: Applicable only to alias entries where the REPLACE option has been specified, this action indicates that the copy failed because a member (not an alias) of the same name exists in the target library.

***StowErr**
: Copy of the member or alias entry has failed due to a STOW error in attempting to write to the target PDS/PDSE library's directory. This may occur if writing a directory entry requires use of a new PDS directory block and none are available. If this is the case, a PDS library compress of the target library may resolve this problem.

`AliasOf`
: If the entry is an alias, then this field displays the member name for which it is an alias.
This field is blank for non-alias library members.

`Truncated`
: Contains "Y" if truncation of any of the records belonging to the member has occurred.
This field is blank if no truncation has occurred.

`RemapError`
: For copy remap only, this column contains "Y" if a field in of any of the remapped records fails to be remapped due to attempted conversion of its data to an incompatible data type. If this occurs, processing stops since the incompatibility between the source and destination record fields will apply to all library members.
This field is blank if no remap error has occurred.

Library Member Move/Copy

The **Library Member Move/Copy** panel is displayed when a move or copy line (prefix) command is actioned for members of a library member list or library member generation list.

For a library member list, this panel allows the user to move or copy members between libraries and, for a single member specification only, rename the member in the target library. For a library member generation list, it allows selective copy of member generations only.

The Library Member Move/Copy panel window may be started via the following:

- Execute the prefix command "**M**" (MOVE) or "C" (COPY) against one or more members in a library member list type window. The resulting Library Member Move/Copy panel window will move or copy all selected members in the list as appropriate.
- Execute the prefix command "C" (COPY) against one or more member generations in a library member generation list type window.

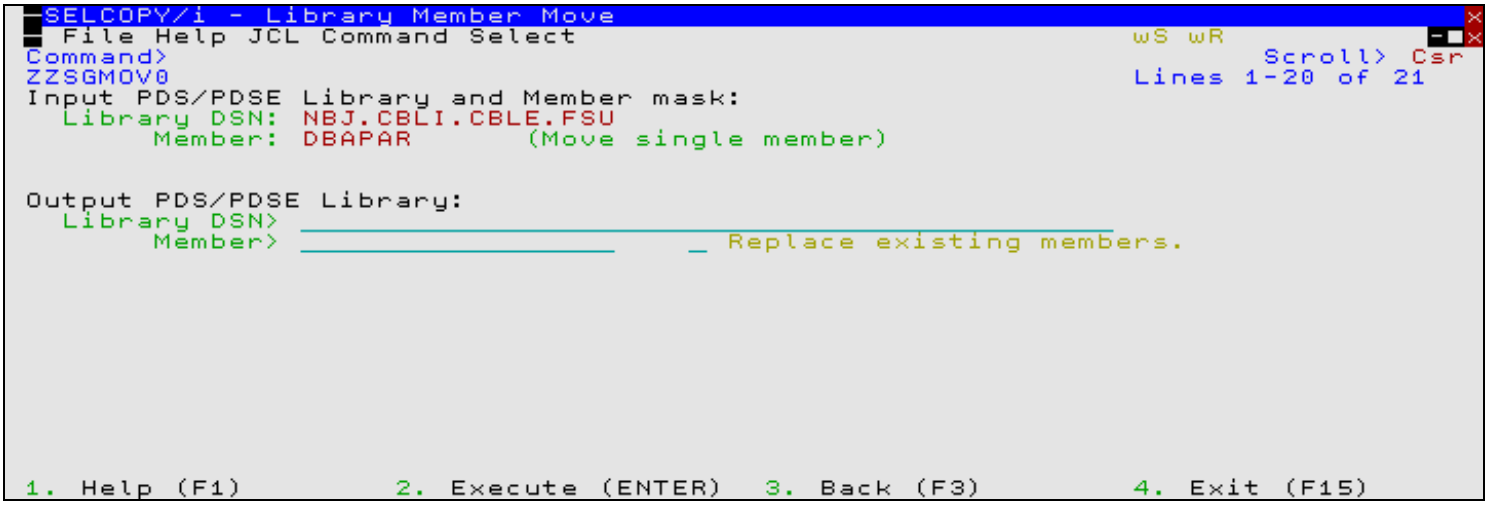


Figure 41. Library Member Move - Single member.

The appearance of the Library Member Move/Copy panel depends on whether multiple or single members or generations have been selected from the list. For a single member or generation selection, the input member field is displayed. For multiple selections, a comment is displayed indicating that the current selection of members or generations may be viewed and optionally updated using the SELECT primary command or menu option.

On pressing the <Enter> key or selecting "Command" or "JCL" menu items, a check is made to determine whether the target library exists. If not, the Allocate NonVSAM dialog is started to allocate a new library. Default allocation attributes are modelled on the source library.

Menu Bar Items

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values entered in the panel so that they may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Command
: Generate an FCOPY command line syntax, with or without parameter **MOVE**, for field entries specified by the user and display it in a temporary CMX file text edit view. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

JCL
: Generate a JCL job stream that executes the **SDEAMAIN** program with input (SDEIN) containing the FCOPY command generated for the specified panel field values.

  The job stream is displayed in a temporary text edit view and may be submitted to batch using the SUBMIT command.

SELECT

Executes the SELECT primary command to display the input list of library members.

Panel Input Fields

PDS/PDSE Library and Member mask:
Output fields that identify the library and member name mask of the member(s) or member generation(s) to be processed.

If the member mask contains "*" (asterisk) or "%" (percent) wild card characters, then the Select Input Members panel may be opened to select from a list of matching member names belonging to that library. SELECT (<PF5>) can be used to redisplay these panels after they have been closed.

If the input is a PDSE version 2 library supporting member generations, the member mask may include a member generation mask. In this case, SELECT may be entered to display and select from a list of matching member generations. Note that member generations belonging to the same member name will always be copied in ascending order of generation number. In doing so, the generation hierarchy is preserved in the output library.

Library DSN:
An unquoted entry which identifies the DSN of the input library. An entry in this input field is mandatory. Note that a TSO prefix is **not** applied to a DSN mask.

Member:
Displayed if only a single member or member generation has been selected from the library or library generation list. This field identifies the member or generation to be processed.

Output PDS/PDSE Library:
Input fields which together identify the output PDS/PDSE library and optionally a library member name.

If a member name is specified and a single input member is selected for move or copy, then the member will be renamed as it is moved/copied. If a member name is specified and multiple members have been selected for move, then the operation will fail. However, if multiple members or member generations have been selected for copy, then all members/member generations are copied and appended to the single output member.

Library DSN>
Identifies the fully qualified data set name of the target library.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

If a DSN is specified for a data set that does not already exist, a prompt data set dialog will be opened to allocate the new output library.

Member>
Specifies the name of a new or existing member within the target library to which all the data from source member(s) or member generation(s) will be copied.

Replace existing members
By default, the copy or move operation will not overwrite a member which already exists in the target library. If this option is selected, existing members may be overwritten.

PFKeys

In addition to the standard interactive panel key assignments for scrolling and navigation, the Library Member Move/Copy panel supports the following:

F1	HELP	Display context sensitive help .
F3	BACK	Close the panel.
P5	SELECT	Select from a list of files that match the Fileid mask.
F14	EXPAND	Expand an input/output field.
F19	BIF	Browse the input member or library.
F20	BOF	Browse the output member or library.

File Search/Update/Copy/Remap

Overview

File Search/Update/Copy/Remap (FSU) utility has more advanced functionality than the File Search utility which supports only a single search string on members of a single PDS(E) library.

Features of the File Search/Update/Copy/Remap utility include:

- Search and optionally update multiple HFS paths or multiple sequential, PDS/PDSE, GDG and/or VSAM data sets.
- Restrict PDS/PDSE library search and/or update to only members with names that match a member name mask.
- Restrict PDSE library member generation search to only generations with absolute or relative generation numbers that match a member generation mask.
- Search and optionally update uncataloged data sets by volume id(s).
- Specify the start record for both search and update operations.
- Restrict the number of records read for search and/or update.
- Restrict the search and/or update operation to a specific area of the file records.
- Apply a **structure** (copybook) overlay to map **input** file records and optionally restrict search/update to all or specific fields in records assigned to specific record types. This is known as a **Formatted File Search/Update**.
- For both **Unformatted** and **Formatted** input file records, optionally specify an output file to which **all** input records will be copied regardless of whether record data has been changed. This is known as **Unformatted/Formatted File Copy**.
- For **Formatted File Search/Update**, optionally specify an output file **and** output structure (copybook) to remap input record fields (i.e. alter field data type, re-order and/or delete fields) whether or not record data is changed. This is known as a **Formatted File Remap**.
- Update unformatted or formatted character data using different length search and update CHANGE strings.
- Control use of blank padding or blank absorption when character search and update CHANGE strings are of different length. Note that the CHANGE operation will fail if the length of the updated record is greater than the file's maximum record length.

Following File Search/Update/Copy/Remap execution, report output is generated in a structured format suitable for presentation to the user in an SDE window view. To generate this report output and in order to perform advanced record selection and field compare, functions and features provided by the structured data environment (SDE) are used. Therefore, the File Search/Update/Copy/Remap utility is only available to users who have a licensed version of SELCOPY installed and operational on their system.

During execution, a progress window is displayed which allows the user to interrupt processing at any point using the Attention key.

Source File Types

The File Search/Update/Copy/Remap utility can process records from any of the following file types in a single execution:

- Cataloged or uncataloged sequential (including multi-volume) datasets.
- Partitioned dataset (PDS/PDSE) members.
- Partitioned dataset (PDSE version 2) member generations.
- GDG datasets.
- VSAM (KSDS, ESDS, RRDS, VRDS).
- HFS Files.
- (DB2 Tables planned but not yet supported).

Output Report

The report generated by the File Search/Update/Copy/Remap utility is a **structured data file**. This is designed to be browsed (not printed) from within a SELCOPY/i session using a structure definition file (SDO) which is also generated automatically during execution of the search/update.

Unless a report DSN is specified, then following execution of the utility in the foreground, the report is generated in storage and automatically displayed in a SELCOPY/i SDE edit view.

For unformatted or formatted **immediate** file update (i.e. input file record data is updated), closing the in-storage generated report will prompt the user to save the report and SDO data sets. This is so a record of file updates may be kept and, if required, may be used as input to the File Update Undo utility to roll back all updates actioned during this execution.

If required, execute command FSUEND to close the report window and save the in-storage report and SDO files generated for a file search, copy, non-immediate update or remap operation.

A list of previously generated reports is displayed on selecting "Reports" from the File Search/Update/Copy/Remap menu bar. To display a report from this list either position the cursor on the required entry then press the <Enter> key or, if configured, double-click the left mouse button on the required entry. Alternatively, generated reports may be viewed using the FSUOUT *<fsu_report_fileid>* command.

See File Search/Update/Copy/Remap Output for a detailed description of the generated output report.

Unformatted File Search/Update/Copy

Unformatted file search, update or copy is the most commonly used form of the utility, acting on **text** data files containing unformatted records.

By definition, Unformatted File Search/Update/Copy operates on records without application of a structure (SDO) or COBOL/PL1 copybook to format record data. i.e. each record is treated as a single character string.

In general, Unformatted File Search/Update/Copy processing proceeds as follows:

1. Sequentially read a record from a file matching an INPUT fileid mask.
2. Check that the record falls within the range of records to be selected for processing as specified by STARTREC/STARTKEY/STARTRBA and FOR syntax. (Note that a direct read will have been performed for VSAM KSDS, ESDS or RRDS.)
3. Apply any **search** criteria on the record as specified by a WHERE expression.
4. Apply any **search** criteria on the record as specified by one or more FIND operations.
5. Apply any data changes on the record as specified by one or more CHANGE operations.
6. If **no** OUTPUT fileid is specified and the record data has been changed by a CHANGE operation, then **update** (replace) the original input file record with the changed record.
7. If an OUTPUT fileid is specified, then **copy** the record to this file regardless of whether record data satisfies search criteria or has been changed.

Processing of the current record stops and continues with the next input record if no OUTPUT fileid is specified (for **copy**) and the record's data does not satisfy specified search criteria or if the record is not within the range of records selected for processing.

Note that search and data change functionality is based on the structured data edit (SDE) FIND, WHERE and CHANGE commands which operate on individual formatted data fields. For unformatted records, the record data occupies a single data field of data type CHAR and length equal to the file's maximum record length. This field has field reference number #1 and field name "Record", either of which may be used as a parameter to WHERE, FIND and/or CHANGE.

Unformatted File Search

Unformatted file search uses WHERE and/or FIND operations to specify search criteria and so select then report only those records that satisfy all of the search criteria.

If both WHERE and FIND criteria are specified, then a record will first have to satisfy the the WHERE expression before the FIND criteria is checked.

Unformatted File Update

Unformatted file update uses CHANGE operations to change one or more occurrences of a character search string to the specified character replace string. The changed record is then written back to the input file replacing the original record read.

Since an update-in-place is performed, the length of the updated record cannot be changed. Any CHANGE operation that results in a change to the record length will flag an error against that record in the output report.

File update should not be actioned without first performing a test run (FSU parameter NOUPDATE) where no records are actually updated but an file update output report is still generated. This allows the user to correct or accept any CHANGE errors before re-running the utility to update the records.

File update will open data sets for update-in-place processing instead of simply for input. An exclusive ENQ will be set when the data set is opened, and reset when it is closed.

Optional file search criteria, as specified for Unformatted File Search, may be used to filter input records before any CHANGE operation is performed. If no search criteria is specified then CHANGE operations will apply to all input records.

Unformatted File Copy

Unformatted file copy copies **all** records within the range of records selected for processing, from all selected input files to a single output file.

If the output file is a PDS/PDSE library then only input library members will be copied, potentially replacing existing members of the same name in the output library.

Optional change operations, as specified for Unformatted File Update, may be used to change record data as it is being copied. Records are copied regardless of whether or not they satify supplied search criteria or have been changed by a CHANGE operation.

Formatted File Search/Update/Copy/Remap

Formatted file search, update and copy is more advanced than the equivalent unformatted operations and also supports additional functionality to remap record data. Formatted operations are invoked where an SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file is specified to map input record data fields for use in a search, update, copy or remap operation.

Records are treated as comprising a number of data fields of pre-determined lengths and of various data types. Each field within the record may be referenced independently (by field name or field reference number) allowing the user to be more descriminate when selecting records and fields within records to be searched and/or changed.

If a COBOL copybook, PL1 include file or an ADATA file (generated from a COBOL or PL1 compilation) is specified, then this file will be used to generate a temporary SDO before proceeding with record formatting. Note that a non-temporary SDO may be generated from the COBOL/PL1/ADATA file using the SDE command, CREATE STRUCTURE.

Each input record is assigned a record type (RTO), defined by the specified or generated SDO, and the field definitions defined by that RTO are used to map the data within the record. SDE determines the record type to be assigned to each record based on any USE WHEN conditions saved in the SDO and the individual record's length. See *"Record Type Assignment"* in the *"SELCOPY/i Structured Data Editor (SDE)"* publication.

In general, Formatted File Search/Update/Copy/Remap processing proceeds as follows:

1. Sequentially read a record from a file matching an INPUT fileid mask.
2. Check that the record falls within the range of records to be selected for processing as specified by STARTREC/STARTKEY/STARTRBA and FOR syntax. (Note that a direct read will have been performed for VSAM KSDS, ESDS or RRDS.)
3. Assign a record type (RTO) to the record.
4. Check that the record is assigned the record type specified by VIEW, otherwise the default record type.
5. Apply any **search** criteria on the record as specified by a WHERE expression.
6. Apply any **search** criteria on the record as specified by one or more FIND operations. FIND search criteria are restricted to a list of record data fields specified by SELECT.
7. Apply any data changes on the record as specified by one or more CHANGE operations. CHANGE operations are restricted to a list of record data fields specified by SELECT.
8. If **no** OUTPUT fileid is specified and the record data has been changed by a CHANGE operation, then **update** (replace) the original input file record with the changed record.
9. If an OUTPUT fileid is specified, then **copy** the record to this file regardless of whether the record is assigned the record type specified by VIEW, or whether the record data satisfies search criteria or has been changed.
10. If an OUTPUT fileid and is specified with accompanying SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file, then copy the record as described in step 9. If, however, the name of an output SDO record type matches that assigned to the record, then the field structure of the output record type will be used to **remap** the input record data fields.

Processing of the current record stops and continues with the next input record if no OUTPUT fileid is specified (for **copy** or **remap**) and the record's data does not satisfy specified search criteria, the record is not within the range of records selected for processing or if the assigned record type does not match that specified by VIEW.

Formatted File Search

Formatted file search uses WHERE and/or FIND operations to specify search criteria and so select then report only those records that satisfy all of the search criteria.

SDE WHERE and FIND operations apply only to records assigned the default record type, as identified by the VIEW operation. Fields to be searched may be identified specifically by name or reference number in the WHERE expression and/or FIND command syntax, however, FIND is restricted to only those fields identified by the SELECT operation.

Formatted search criteria are sensitive to the data type and length of the formatted fields and so appropriate action is taken when testing a field. e.g. an arithmetic compare for a numeric data field.

If both WHERE and FIND criteria are specified, then a record will first have to satisfy the the WHERE expression before the FIND criteria is checked.

Formatted File Update

Formatted file update uses CHANGE operations to change one or more occurrences of a search string to the specified replace string. The changed record is then written back to the input file replacing the original record read.

Since an update-in-place is performed, the length of the updated record cannot be changed. For formatted records, any CHANGE operation applied to the expanded data that results in a change to the **unexpanded** record length will flag an error against that record in the output report.

File update should not be actioned without first performing a test run (FSU parameter NOUPDATE) where no records are actually updated but an file update output report is still generated. This allows the user to correct or accept any CHANGE errors before re-running the utility to update the records.

File update will open data sets for update-in-place processing instead of simply for input. An exclusive ENQ will be set when the data set is opened, and reset when it is closed.

SDE CHANGE operations apply only to records assigned the default record type, as identified by the VIEW operation. Fields that are eligible for change may be identified specifically by name or reference number in the CHANGE command syntax these are restricted to only those fields identified by the SELECT operation.

Formatted data CHANGE operations are sensitive to the data type and length of the formatted fields and so appropriate action is taken when changing field data. e.g. maintain the separate length field of a changed XVARCHAR or VARCHAR field.

Optional file search criteria, as specified for Formatted File Search, may be used to filter input records before any CHANGE operation is performed. If no search criteria is specified, then CHANGE operations will apply to all input records assigned the default record type.

Formatted File Copy

The only difference between formatted and unformatted file copy is in the use of an SDE structure to optionally change field data. If a copy is to be performed without performing changes to the record data, then Unformatted File Copy may be used.

Formatted file copy copies **all** records within the range of records selected for processing, from all selected input files to a single output file.

If the output file is a PDS/PDSE library then only input library members will be copied, potentially replacing existing members of the same name in the output library.

Optional change operations, as specified for Formatted File Update, may be used to change record data as it is being copied. Records are copied regardless of whether or not they are assigned the default record type, satisfy any supplied search criteria or have been changed by a CHANGE operation.

Formatted File Remap

Formatted file remap copies **all** records within the range of records selected for processing, from all selected input files to a single output file. Furthermore, it uses an output SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file to potentially remap the data fields belonging to those records assigned a record type that also exists in the output SDO.

For these records only, remap involves building a new output record using the field structure defined by the matching output record type. i.e The output record will contain **all** fields defined by this record type with field data initialised to default values.

Output record fields then inherit values from input record fields of the same name performing conversion between the input and output field data types as required. This has the effect of filtering, repositioning and reformatting data from input record fields before writing them to the output file.

Where matching input and output fields are of incompatible data types or where input data is invalid in the output field or would be truncated, then the output field does **not** inherit the input field value and a remap error is flagged for that field.

If a remap error occurs, output for the current data set is stopped, an Interrupt record is written to the report and, if output is to a PDS/PDSE library, processing continues at the next input library member.

Optional change operations, as specified for Formatted File Update, may be used to change data in the input record fields before it is copied or remapped to the output record. Records are copied or remapped regardless of whether or not they are assigned the default record type, satisfy any supplied search criteria or have been changed by a CHANGE operation.

File Search/Update/Copy/Remap Panels

The **File Search**/**Update**/**Copy**/**Remap** panels (ZZSGFSU0) are displayed when the File Search/Update/Copy/Remap Files utility is started interactively.

These panels allow the user to scan data sets, HFS files and/or PDS/PDSE members for search strings, optionally change data in selected records and then update the input record, or copy/remap input records to a single output file or multiple members of a PDS/PDSE library.
See *"File Search/Update/Copy/Remap Utility"* for a detailed description of functionality.

The File Search/Update/Copy/Remap utility panel views and their sub-panels are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select option 6. 'Search/Update' in the SELCOPY/i Primary option menu.
- Select 'File Search/Update/Copy/Remap' from the Utilities menu.
- Execute the command FSU with no parameters from the command line of any window.

- Execute the prefix command "**F**" from a file List type window. The resulting File Search/Update/Copy/Remap panel window will treat the corresponding list entry as the INPUT fileid mask.

By default, field entries are populated with arguments and options that were entered the last time the Compare Files Utility panels were used. Also, many field entries are optional and need to be activated by entering "/" in the preceding field.

Basic File Search Panel

FSU: Basic File Search Panel

The **FSU: Basic File Search** panel view is the first displayed when the File Search/Update/Copy/Remap Files utility is started interactively.

This panel provides facility to perform a search on a single character string or numeric value occurring within records belonging to one or more files that match a specified fileid mask. Optional record selection criteria may be applied to each of the selected input files.

The search is actioned using optional parameters supported by the standard edit FIND command.

It also includes an option to start the Extended File Search/Update/Copy/Remap panels to perform more advanced functions supported by the utility.

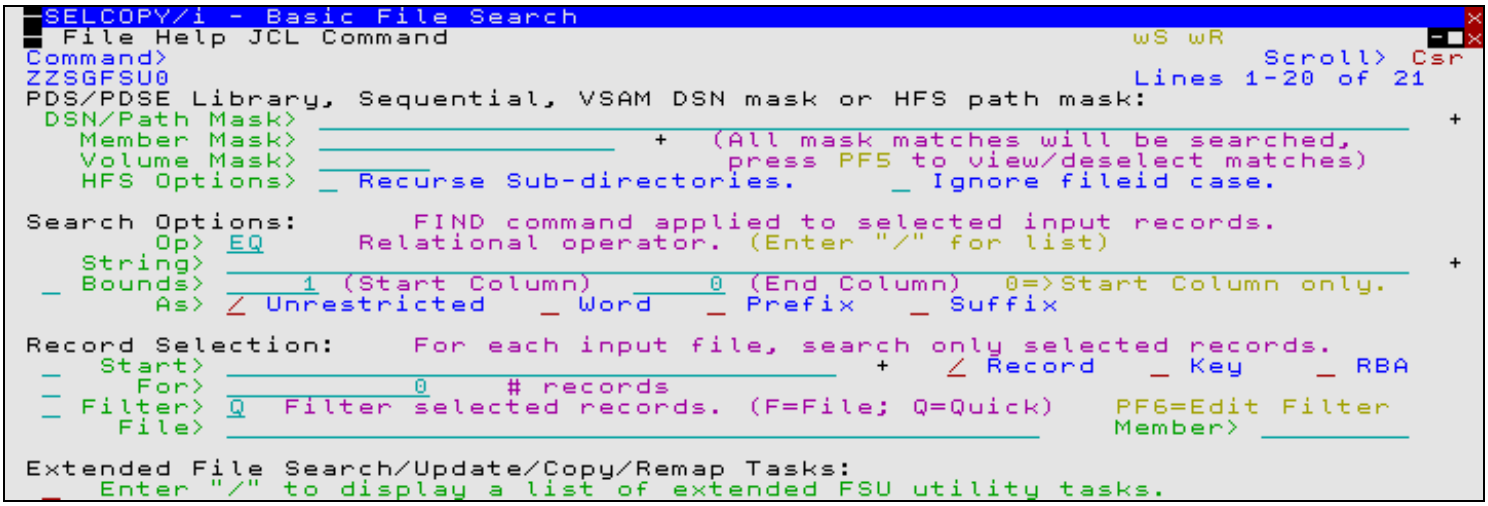


Figure 42. Basic File Search Panel.

Depending on whether the Extended File Search/Update/Copy/Remap Tasks option has been set, pressing the <Enter> key or, if configured, double-clicking the left mouse button, will will either verify the input fields and action the basic file search, or display the FSU: Extended File Search, Update, Copy & Remap Tasks panel view.

If the basic search is actioned, then as the utility executes a progress window is displayed which allows the user the opportunity to interrupt processing using the Attention key.

Alternatively, the user may select an item from the menu bar.

Menu Bar Items

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values entered in the panel so that they may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Command
: Generate the FSU command line syntax for field entries specified by the user and display it in a temporary CMX file text edit view. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

JCL
: Generate a JCL job stream that executes the **SDEAMAIN** program with input (SDEIN) containing the FSU command generated for the specified panel field values.

Note that batch execution of FSU requires specification of a new or existing output report data set name. See the FSU REPORT *fileid* parameter for details.

The job stream is displayed in a temporary text edit view and may be submitted to batch using the SUBMIT command.

Panel Input Fields

`PDS/PDSE Library, Sequential, VSAM DSN mask or HFS path mask:`

Fields which together constitiute a fileid mask that identifies one or more files to be searched.

The utility supports use of a fileid mask and will processes all HFS files, sequential, GDG, VSAM and PDS/PDSE data sets that match the mask.

To be more selective, press <PF5> (SELECT) to display the Select Files to Process panel which contains a list of selectable fileids that match the specified fileid mask. Furthermore, for PDS/PDSE library entries, the Select Input Members panel may then be opened to select from a list of matching member names or member generations belonging to that library.

Having displayed the Select Files to Process panel, a list of the selected, individual fileids will be passed to the utilty instead of the generic fileid mask. Furthermore, if only a single fileid has been selected for processing, then the selected file DSN/Path name will replace the generic entry in the **DSN/Path Mask>** field and, if applicable, the selected volume id and member name will replace the generic entries in the **Volume Mask>** and **Member Mask>** fields respectively.

These fields collectively correspond to the FSU parameter INPUT.

`DSN/Path Mask>`

An unquoted entry which identifies DSN mask, a Concatenated Library Directory path (&DDName) a completed Fileid mask, a DDName mask or an HFS file path mask. An entry in this input field is mandatory.

- **Concatenated Library Directory path (&DDName)**
  If the value entered starts with an ampersand (&) immediately followed by a DDName, then it is treated as Concatenated Library Directory path. (e.g. &SYSEXEC)

  This means that only the first occurrence along the directory path of each member that matches the specified **Member Mask>** will be processed. Note that members are processed in alphabetical order of member name.

- **HFS file path mask**
  Identified by the presence of "." (dot/period) as the first character or "/" (slash) anywhere within the mask, an HFS file path mask may be absolute or relative to the current HFS working directory. See USS PWD in Unix System Services (USS) Commands. "./" should prefix the mask if files in the current HFS directory are to be selected.

  Wild card characters "%" (percent), representing a single characters, and "*" (asterisk), representing zero or more characters, are supported in the name portion of the HFS file path mask. The name portion of the HFS path is the character string at the end of the path that follows the last "/" (slash) of the fileid or is the entire path name if "/" is not specified.

  If an HFS file path mask is specified, the contents of the **Volume Mask>** and **Member Mask>** fields will be ignored and **HFS Options>** settings respected.

- **DDName mask**
  If the value entered is not an HFS file path mask, is of length less than or equal to 8 characters and does not include "." (dot/period), and no Volume mask has been specified, then the value is considered to be a DDName mask. A DDName mask may be used to process non-HFS files that are currently allocated to DDNames that match the DDName mask. (e.g. SYSEXEC)

  Wild card characters "%" (percent), representing a single characters, and "*" (asterisk), representing zero or more characters, are supported in a DDName mask.

  If a DDName mask is specified, the enties in the **Member Mask>** field may still be used to select matching member names belonging to PDS/PDSE library data sets allocated to matching DDNames. **HFS Options>** settings are ignored.

- **Completed Fileid mask**
  If the value entered is not an HFS file path mask but includes a volume mask specification and/or a member mask specification, then the value is considered to be a completed Fileid mask.

  A completed fileid mask is a DSN mask with a volume mask and/or one or more PDS/PDSE member name masks expressed in the following format:

  ```
  {volmask:}data.set.name.mask{( membmask{ {,} membmask...} )}
  ```

  Specification of one or more member masks between a single pair of "( )" (parentheses) will restrict processing to only PDS/PDSE library data sets. Multiple PDS/PDSE member masks must be separated by a "," (commma) and/or one or more intervening blanks.

  Specification of a 1 to 6 character volume mask prefix followed by ":" (colon) restricts processing to only cataloged or uncataloged data sets that have extents on matching volume ids.

The fileid mask supports wild card characters as described for Volume Mask> and Member Mask> fields and for a DSN Mask below.

Fileid Mask Examples:

```
PE1.DEV.SRC.COBOL.CRKSW00(*)
SYS6.JNP*.**
OEM.TEST%%.**.CBLI.**(BOX*,D%T*,*ALL)
Z9RES1:ADCD.**
Z9RES*:ADCD.Z19.P%%%LIB(*)
```

- **DSN mask**
  If the value entered does not meet the criteria described above for an HFS file path mask, completed Fileid mask or a DDName mask, then the value is considered to be a DSN mask.

  The DSN mask is joined with any member and volume mask specified in the Volume Mask> and Member Mask> fields respectively to identify a single fileid mask.

  The following wild card characters are supported within a DSN Mask:

  `*` A single asterisk represents a DSN qualifier or zero or more characters within a DSN qualifier.
  e.g. `DEV.CBLINS.*.JCL`, `DEV.CBLINS.TEST*.ISP*LIB`, `DEV.CBLINS.*.*`

  `**` Double asterisk represents zero or more qualifiers within a DSN. Double asterisk may only be preceded or followed by the qualifier separator, "." (dot/period).
  e.g. `DEV.CBLINS.**`, `DEV.CBLINS.**.CBLE`

  `%` A single percent sign represents exactly one character within a DSN qualifier.
  e.g. `DEV.CBLINS.TEST0%.JCL`, `DEV.CBLI%%.TEST06.CBLI.%%%`

  Note that a TSO prefix is **not** applied to a DSN mask.

Volume Mask>
Optionally specify a volume name mask of maximum length 6-bytes.

Processing will be restricted to only those cataloged or uncataloged data sets that match the DSN mask **and** also have extents that exist on a volume that matches the volume mask.
The volume mask supports wild card characters as follow:

`*` A single asterisk represents a complete volume name or zero or more characters within a volume name.
e.g. `CBL*`, `*RES*`

`%` A single percent sign represents exactly one character within the volume mask.
e.g. `Z9DB9%`, `%%XV3%`

This field entry is ignored if the DSN/Path Mask> field does not contain a DSN mask.

Member Mask>
Optionally specify one or more PDS/PDSE member name or member generation masks separated by a "," (comma) and/or one or more intervening blanks.
e.g. `BLOCK, PROFILE  BOXSEQ`

If a value is entered in this field, then only PDS/PDSE libraries that match the fileid mask will be selected for processing. Non-PDS/PDSE library data sets will be excluded.

If **no** value is entered in this field, then all files that match the fileid mask will be selected for processing and a default member name mask of "*" will apply to all PDS/PDSE libraries included in this selection.

Processing will be restricted to only those PDS/PDSE data sets that match the DSN mask **and** only members or member generations that match any one of the supplied masks.
A member mask supports wild card characters as follow:

`*` A single asterisk represents an entire member name or zero or more characters within a member name.
e.g. `CBL*5`, `BOX*`, `D*T*`

`%` A single percent sign represents exactly one character within a member name mask.
e.g. `H%`, `D%R*`, `E%A`

This field entry is ignored if the DSN/Path Mask> field does not contain a DSN mask or a DDNAme mask.

HFS Options>
Specify options that govern HFS file selection.

Recurse Sub-directories
Set this option on to process matching files within all directories and sub-directories beneath the absolute or relative directory specified by the HFS file path mask
This field corresponds to the FSU HFS Options parameter RECURSE.

Ignore fileid case
Set this option on to disable case sensitivity when matching HFS file names with the file **name** portion of the specified HFS file path mask. The name portion of the HFS file path is the character string following the last "/" (slash) of the fileid mask or the entire HFS file path if no "/" is included in the fileid mask. This field corresponds to the FSU HFS Options parameter CASEIGN.

These field entries are ignored if the DSN/Path Mask> field does not contain an HFS file path mask.

Search Options:
Fields which together constitiute an edit FIND command used to search unformatted input file records.

These fields collectively correspond to the FSU parameter FIND.

Op>
Identify the relational operator used when comparing the record data against the search value.

If testing for a non-equality relationship using a character string search value, then the EBCDIC values assigned to characters in the search string and record data determine the relationship. (e.g. GT, LT)

Enter "/" to display a list of valid entries and a brief description.

String>
Specify the search value. This may be one of the following:

1. An unquoted numeric value.
2. A quoted character string (case insensitive search). e.g. 'John Doe' is the same as 'JOHN DOE'.
3. A quoted character string prefixed with "C" (case sensitive search). e.g. C'John Doe'.
4. A quoted hexadecimal string prefixed with "X". e.g. X'00'.
5. A quoted picture string prefixed with "P". e.g. P'USER###'.
   See the FIND command for details on supported picture string special characters.
6. A quoted unix-style "Regular Expression" string prefixed with "R". e.g. R'[Pp][Ff](:d#)'
   See Regular Expressions for details on supported regular expression string special characters.

Limit>
The maximum number of FIND hits (records hit) to report per file, or library member.

Specify a value of **zero (0)** to report all hits.

Efficiency gains may be achieved by specifying **Limit=1** where it is required only to establish whether or not each member contains a particular string, since once the first hit is reported further processing for that member will be bypassed.

Note that *LIMIT n is ignored if the FSU operation is a file-copy or update-in-place. e.g. if copying a library and making in-flight changes of one or more strings, then members will be copied complete and strings will be updated (where present) in all records, in spite of LIMIT being specified.*

Context>
Display a specified number of input records immediately before and after each hit in order to provide context.

Bounds>
Optionally activate the **Bounds>** input fields to specify the record columns between which the search will occur. If not activated, the search will include all columns within the length of the input records.

Start Column
Specifies the start (or only) record column from which the scan for the search string will begin. Record data in positions or fields that occur before this start column value is not searched.

A negative value represents a position in the record relative to the end of the record. i.e. whereas position 1 references the 1st character in the record, position -1 references the last character.

This field corresponds to SDE FIND parameter *pos1*.

End Column
Specifies the end record data column beyond which no part of the search value may be found. Only record data between the start and end columns will be searched.

A negative value represents a position in the record relative to the end of the record. i.e. whereas position 1 references the 1st character in the record, position -1 references the last character.

If a 0 (zero) end column is specified then the end column is the start column number plus the length of the search value minus 1. i.e. the search string will only be found if it begins in the start column.

This field corresponds to SDE FIND parameter *pos2*.

As>
Enter "/" in the appropriate radio button field to apply restrictions to the position of the search string within record data as follow:

Unrestricted	The search value may be found anywhere within the specified column bounds of the input records.
Word	

	The search value may only be found if it is complete word which falls within the specified column bounds of the input records. i.e. the value must either be preceded by a non-alphanumeric character or begin at the start column bound, and either precede a non-alphanumeric character or finish in the end column bound.
Prefix	The search value may only be found if it occurs at the start of a word and falls within the specified column bounds of the input records. i.e. the value must either be preceded by a non-alphanumeric character or begin at the start column bound, and be followed by an alphanumeric character.
Suffix	The search value may only be found if it occurs at the end of a word and falls within the specified column bounds of the input records. i.e. the value must be preceded by an alphanumeric character and must either precede a non-alphanumeric character or finish in the end column bound.

These option fields correspond to SDE FIND parameters PREFIX, SUFFIX and WORD.

Record Selection:

Fields which together identify criteria by which subsets of records from every input file are selected for processing.

Start>

If activated, the **Start>** field specifies the first record in every file matching the fileid mask at which processing will start. Records occurring sequentially before the start record will be bypassed. If this field is not activated, records are selected beginning at record 1.

This input field may contain a record number, an RBA number (for ESDS input only), or a key string (for KSDS input only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

This field corresponds to the FSU parameters STARTREC *recno*, STARTRBA *rba* and STARTKEY *key*.

Record | Key | RBA

Identifies the type of start value specified in the **Start>** field. Enter "/" in the appropriate, mutually exclusive parameter field.

For>

If activated, then for each file matching the fileid mask, the **For>** field specifies the maximum number of records within that file for which processing may occur. If this field is not activated, records are selected beginning at start record and ending at the last record in the file.

This field corresponds to the FSU parameter FOR.

Filter>

If activated, the **Filter>** field specifies options to either generate a new record filter or use an existing record filter file. A record filter will perform further subsetting on input file records selected for processing by the Start> and/or For> input fields.

Filter options are as follow:

- `Q` On pressing <PF6>, the Quick Filter dialog panel will be opened in order to generate a temporary filter on the unformatted record data.
- `F` Use a permanent filter identified by the sequential data set or member name in the File> field.

  On pressing <PF6>, the Create File Filter dialog panel will be opened in order to display the contents of an existing filter file or create and save a new filter file.

If option "F" is selected, then specifiaction of a filter fileid is mandatory.

File>

Identifies the fully qualified data set name of a sequential data set or PDS/PDSE library containing a record filter. Quotes are unnecessary but permitted.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>

If the File> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Extended File Search/Update/Copy/Remap Tasks:

Select this option to bypass execution of the the file search when <Enter> is pressed and instead display a list of FSU utility functions ordered by task.

Primary Commands

The following commands are supported by the Basic File Search panel view.

BROWSEINPUTFILE

```
>>--+- BROWSEInputfile ----------+-------------------------------------><
    +- BIF ----------------------+
    +- VWINP --------------------+
```

Browse the fileid represented by the PDS/PDSE Library, Sequential, VSAM DSN mask or HFS path mask fields. The file is considered to be an HFS file path if the combination of these fields describes a fileid mask containing wild card characters.

BROWSEINPUTFILE is assigned to <F19> by default.

CMX

```
>>--+- CMX ----------------------+-------------------------------------><
    +- EDITCMX ------------------+
```

Generate command syntax. Same as menu bar item, **Command**.

CMX is assigned to <F17> by default.

FILTER

```
>>---- FILTer -------------------------------------------------------><
```

Depending on the value entered in the Filter Type field ("Q" or "F"), FILTER attempts to display either the Quick Filter Selection Criteria panel or the Create File Filter panel for the file represented by the Filter fields.

FILTER is assigned to <F6> by default.

JCL

```
>>--+- JCL ----------------------+-------------------------------------><
    +- EDITJCL ------------------+
```

Generate a batch JCL and command syntax. Same as menu bar item, **JCL**.

JCL is assigned to <F18> by default.

RUN

```
>>--+- RUN ----------------------+-------------------------------------><
    +- EXECSYNtax ---------------+
```

Verify input fields and execute the file search utility. This is the default action on pressing <Enter> when the Extended File Search/Update/Copy/Remap Tasks option has not been selected.

SELECT

```
>>---- SELect -------------------------------------------------------><
```

SELECT will open a sub-panel displaying all files tha match the fileid mask represented by the PDS/PDSE Library, Sequential, VSAM DSN mask or HFS path mask fields.

Specific files and library members may then be selected for processing by the file copy utility.

SELECT is assigned to <F5> by default.

Extended File Search/Update/Copy/Remap Panels

FSU: Extended File Search, Update, Copy & Remap Tasks

The **FSU: Extended File Search, Update, Copy & Remap Tasks** panel view is displayed when the Extended File Search/Update/Copy/Remap Tasks option has been selected in the Basic File Search panel.

This panel provides a list of selectable tasks that may be performed using the FSU utility. Selecting one of these tasks will begin a sequence of panel views which prompt the user for required and optional parameters, used to generate an FSU command stream. On completing entries in these panel views, the generated command may be executed in the foreground, inserted in a JCL job stream for batch execution or displayed in an edit view in a format suitable for execution using the ACTION facility.

Following selection of one of these options, the associated task description is displayed in all subsequent panel views.

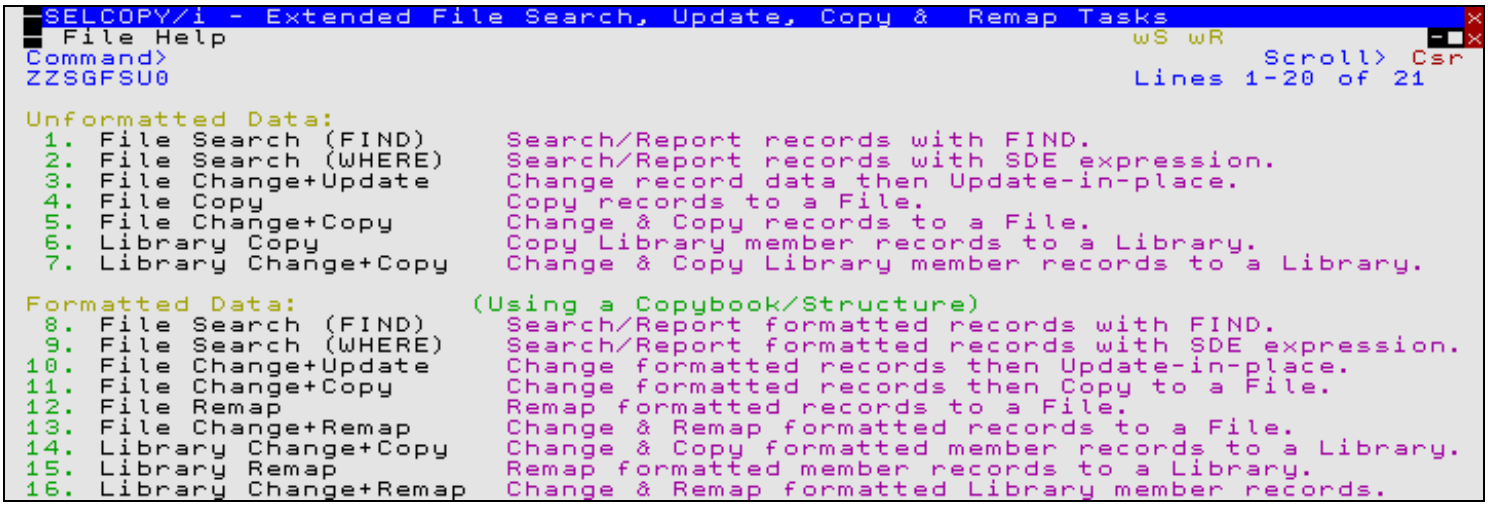


Figure 43. FSU: Extended File Search, Update, Copy & Remap Tasks.

Menu Bar Items

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Panel Options

File Search, Update, Copy and Remap utility operations may be grouped as applying to unformatted or formatted file records. Record data is considered to be formatted if they are mapped by a structure (COBOL or PL/1 Copybook or SELCOPY/i SDO).

1. Unformatted File Search (FIND)
: Like the basic search panel, this option will generate a report of file records that satisfy a FIND operation search value.

  Additional features include:

  1. Specification of multiple input file masks allowing a mixture of input HFS files, sequential, VSAM and GDG data sets and library members.
  2. Additional HFS file input record formats.
  3. Specification of multiple FIND operations and control of whether a record must satisfy **any** or **all** of the FIND operations.
  4. Optional write of report output to DASD data set.

2. Unformatted File Search (WHERE)
: This option is similar to Unformatted File Search (FIND) except that, instead of using FIND operations, the record search identifies records to be reported using an edit WHERE operation which specifies an SDE expression.

  This task may optionally perform additional FIND operations on records selected by the SDE expression so that only records that satisfy both the SDE expression **and** FIND search criteria are reported.

3. Unformatted File Change+Update
: This option performs one or more edit CHANGE operations on selected input records and, if data is changed, perform an in-place update on the input record. If multiple CHANGE operations are specified, the option is provided to attempt **all**

CHANGE operations or to stop following the **first successful** CHANGE operation.

This task may optionally perform additional FIND and/or WHERE search operations to ensure that only records that satisfy these operations' search criteria are satisfied before attempting to perform a CHANGE on a record's data.

Note that record update does not allow a change to a record's length so any CHANGE operation that would result in a record length change will fail.

4. Unformatted File Copy

Like File Copy this option will copy records from selected input files to a single output sequential, VSAM or GDG data set, HFS file or PDS/PDSE library member.

Additional features include:

1. Specification of multiple input file masks allowing a mixture of input HFS files, sequential, VSAM and GDG data sets and library members.
2. Specification of HFS file input record formats.
3. Optionally, display report output and/or write it to a DASD data set.

5. Unformatted File Change+Copy

This option is similar to Unformatted File Copy except that records may be changed as they are copied to the output file.

As detailed under Unformatted File Change+Update, one or more edit CHANGE operations may be specified with optional FIND and/or WHERE search operations to provide additional record vetting.

Unlike update, copy allows changes to a record's length so, unless the operation extends the record length beyond the defined maximum, a CHANGE operation that results in a record length change will succeed.

6. Unformatted Library Copy

This option is similar to Unformatted File Copy except that only input PDS/PDSE library members are selected for processing and the output data set must be a PDS/PDSE library DSN. Any files selected by the input mask that are not library members are ignored. All selected members are copied to members of the same name in the output library.

Options exist to allocate the output library data set modelled on the first input library processed and to replace existing members if necessary.

Library copy has advantages over IEBCOPY in that members can be copied between libraries of different DCB geometry.

7. Unformatted Library Change+Copy

This option is similar to Unformatted Library Copy, except that, like Unformatted File Change+Copy, records may be changed as they are copied to the output file.

8. Formatted File Search (FIND)

This option provides similar functionality to Unformatted File Search (FIND) with the following differences:

1. Specification of an input structure file is mandatory. A structure may be a COBOL or PL1 Copybook, ADATA data set or a SELCOPY/i SDO structure.
2. Specification of an individual record-type mapping defined within the specified structure file is mandatory. Only input data records that are assigned this record-type mapping are searched. Input data records not assigned this record-type mapping are bypassed.
3. Whether specifying a single or multiple FIND operations, the column bounds, which restrict the search to a specific area of the input records, are denoted by a start and end field names instead of character positions.
4. Arithmetic comparisons are performed for numeric search values when tested against numeric fields regardless of their source data type.

9. Formatted File Search (WHERE)

This option provides similar functionality to Unformatted File Search (WHERE) but with the same differences as described by Formatted File Search (FIND) above.

Unlike Unformatted File Search (WHERE), the specified SDE expression may perform tests on specific fields within the formatted records.

10. Formatted File Change+Update

This option is similar to Unformatted File Change+Update but with the additional, mandatory specification of input structure data set and record-type.

The specified CHANGE operation search and replace values apply to individual fields and are sensitive to field data-type. Likewise, column bounds are specified using field names.

Any record vetting using FIND and/or WHERE, performed prior to execution of a CHANGE operation, also apples to individual fields and is sensitive to field data-type.

11. Formatted File Change+Copy

Like Unformatted File Change+Copy, this option allows records to be changed as they are copied to the output file.

The specified CHANGE operation search and replace values apply to individual fields and are sensitive to field data-type. Likewise, column bounds are specified using field names.

Any record vetting using FIND and/or WHERE, performed prior to execution of a CHANGE operation, also apples to individual fields and is sensitive to field data-type.

12. Formatted File Remap

Like Unformatted File Copy, this option will copy records from selected input files to a single output sequential, VSAM or GDG data set, HFS file or PDS/PDSE library member. However, input record data mapped by fields in the input structure is remapped to potentially different positions and data types in the output record. These output record field positions and data types are determined by an output structure.

Specification of an input and output structure file is mandatory. A structure may be a COBOL or PL1 Copybook, ADATA data set or a SELCOPY/i SDO structure. Note that specification of a record-type mapping is **not** necessary for remap.

13. Formatted File Change+Remap

This option provides the same functionality as Formatted File Remap with the added ability to change data in fields as it is remapped to the output file.

Unlike Formatted File Remap, specification of a record-type mapping is mandatory to identify to identify the records to which the specified CHANGE, and optionally FIND and/or WHERE, operations are to be applied.

Note that CHANGE, FIND and WHERE operations are applied to the record data following remap. i.e. using the record-type definition belonging to the output structure.

14. Formatted Library Change+Copy

Like Unformatted Library Change+Copy, this option will only copy PDS/PDSE library members that have been selected for processing and the output data set must be a PDS/PDSE library DSN.

Like Formatted File Change+Copy, CHANGE, and optionally FIND and/or WHERE, operations may be applied to the selected formatted records as they are copied to the output library member.

15. Formatted Library Remap

This option is similar to Formatted File Remap except that only selected input records belonging to PDS/PDSE library members are copied and remapped to members of the same name in an output PDS/PDSE library.

16. Formatted Library Change+Remap

This option provides the same functionality as Formatted Library Remap with the added ability to change data in fields as it is remapped to the output members.

Specification of a record-type mapping is mandatory to identify to identify the records to which the specified CHANGE, and optionally FIND and/or WHERE, operations are to be applied.

Note that CHANGE, FIND and WHERE operations are applied to the record data following remap. i.e. using the record-type definition belonging to the output structure.

FSU: Input File(s) Specification

The **FSU: Input File(s) Specification** panel view is common to all the utility tasks and is always the first view displayed in the sequence.

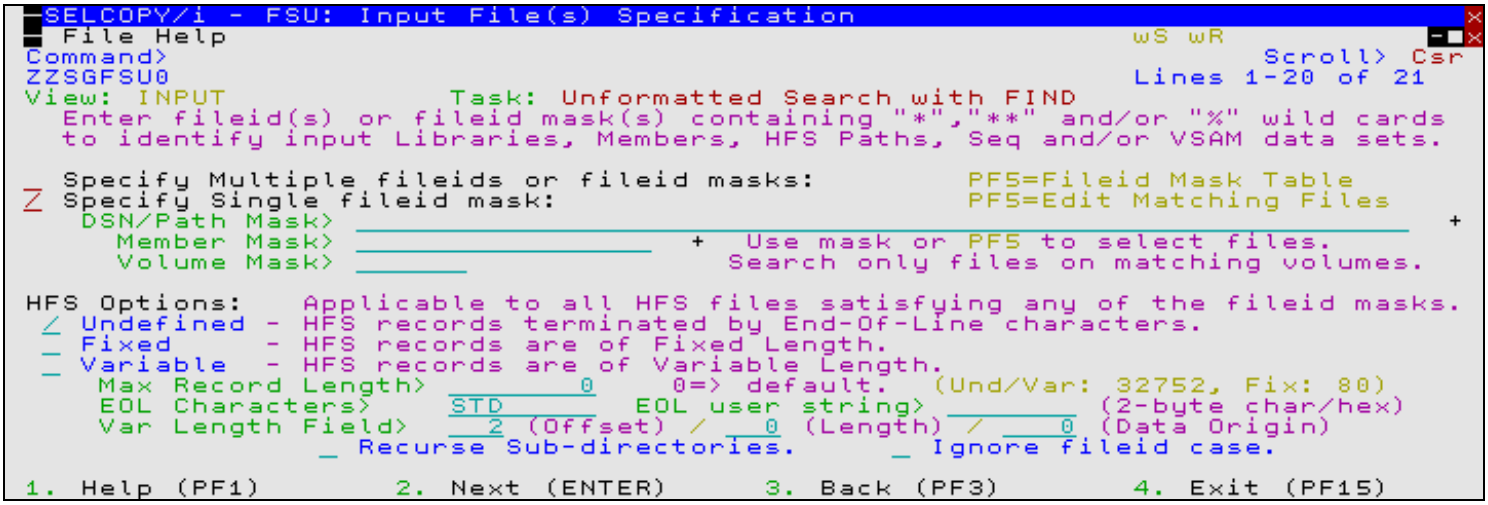


Figure 44. FSU: Input File(s) Specification.

Menu Bar Items

File
The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

Panel Fields

Fileid Mask Specification
Choose between 2 mutually exclusive options which identify the method by which input files will be selected:

Specify Multiple fileids or fileid masks:
Select this option to open the Multiple Fileid Masks panel to enter more than one fileid mask. For each fileid mask, further selection and deselection may be performed on the list of files, library members and/or library member generations that match that mask.

The list of fileid masks may contain any combination of HFS file path masks, Completed Fileid masks, DDName masks and DSN masks. See the DSN/Path Mask> field in the the FSU: Basic File Search panel for a description of these different types of fileid mask.

If this option is selected, <Enter> will display the **Multiple Fileid Masks** panel. When a list of fileid masks has been entered and closed, primary command SELECT (assigned to the <F5> by default) may be used to redisplay the panel.

Specify Single fileid mask:
Select this option if only one fileid mask is needed to identify all the required input files.

If selected, the contents of the **DSN/Path Mask>**, **Member Mask>** and **Volume Mask>** fields are used to constitute the generated fileid mask. See description of PDS/PDSE Library, Sequential, VSAM DSN mask or HFS path mask: in the **FSU: Basic File Search** panel for details.

Primary command SELECT (assigned to the <F5> by default) will display the Select Files to Process panel which contains a list of selectable fileids that match the specified fileid mask. Furthermore, for PDS/PDSE library entries, the Select Input Members panel may be opened to select from a list of matching member names or member generations belonging to that library.

HFS Options:
Options and values that apply to **all** input HFS files. These are ignored for non-HFS input files.
Undefined | Fixed | Variable
Identify the format of input HFS records.
Undefined indicates that records are terminated by an End-of-Line (EOL) string.
Fixed indicates that all records are of a fixed length as defined by a specified LRECL.
Variable indicates that all records are of variable length as defined by a length field within the data.

Max Record Length>
Applicable to each of the record formats, this value defines the LRECL (maximum length) of input records. A record longer than this value will be chopped into multiple records.
A 0 (zero) value implies the default which is 32752 for Undefined and Variable record formats and 80 for Fixed record format.

EOL Characters>
Applicable to Undefined record format only, choose from one of the following EOL character combinations:

STD	-	Any standard line-end.
CR	X'0D'	Carriage Return.
LF	X'0A'	Line Feed.
NL	X'15'	New Line.
CRLF	X'0D0A'	Carriage Return + Line Feed.
LFCR	X'0A0D'	Line Feed + Carriage Return.
CRNL	X'0D15'	Carriage Return + New Line.
user	-	A 2-byte user string specified in **EOL user string>**

EOL user string>
Applicable only if **EOL Characters>** is set to **user**, this field specifies the user supplied 2-byte EOL string. It may be specified in character or hexadecimal notation. (e.g. '##', X'FFFF')

Var Length Field>
Applicable to Variable record format only, these fields identify the location of the record length fields within the data.

(Offset)
Offset of the record length field from the start of the record. Default is 0. (i.e. the length field is at the start of the record.)

(Length)
Length (number of bytes) of the record length field. Default is 2.

(Data Origin)
Offset into the record data at which the value in the record length field is to be applied. Default is 0. (i.e. the record length include the length field.)

Recurse Sub-directories
Set this option on to process matching files within all directories and sub-directories beneath the absolute or relative directory specified by the HFS file path mask

This field corresponds to the FSU HFS Options parameter RECURSE.

Ignore fileid case
Set this option on to disable case sensitivity when matching HFS file names with the file **name** portion of the specified HFS file path mask. The name portion of the HFS file path is the character string following the last "/" (slash) of the fileid mask or the entire HFS file path if no "/" is included in the fileid mask.

This field corresponds to the FSU HFS Options parameter CASEIGN.

FSU (unformatted): Record Selection

The **FSU (unformatted): Record Selection** panel view provides the opportunity to enter unformatted record selection criteria to be applied to **every** selected input file.

Any specified filter must operate on unformatted record data. i.e. areas of the record may be tested based only on data position and length within the record.

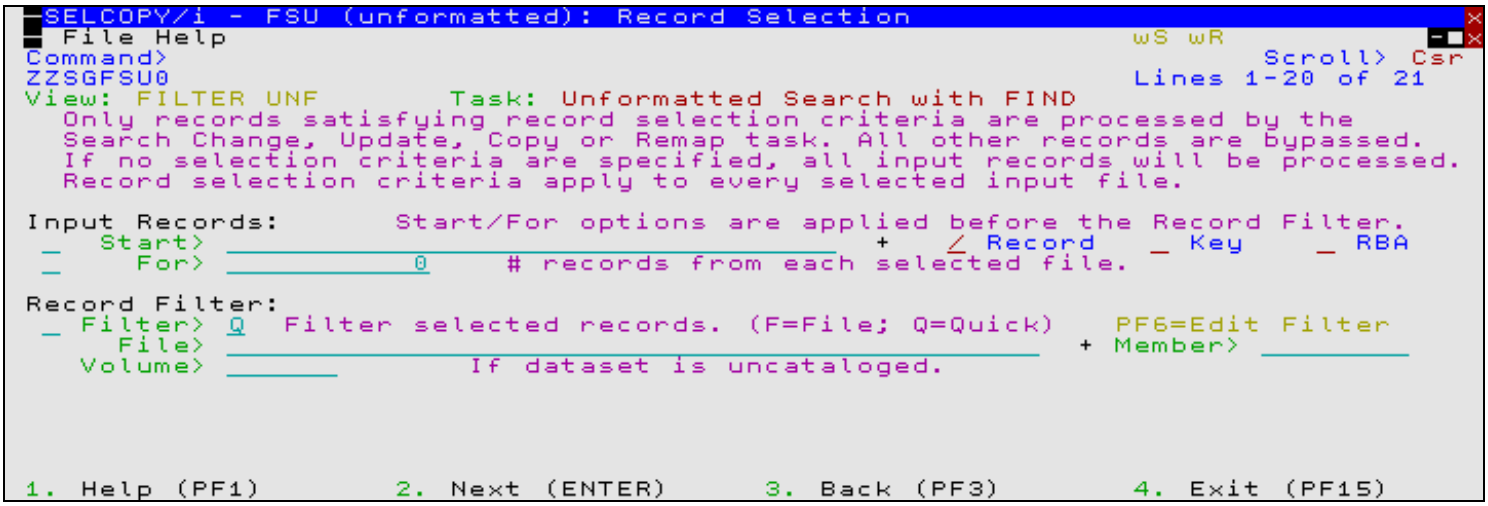


Figure 45. FSU (unformatted): Record Selection.

Menu Bar Items

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Panel Fields

Input Records:
: The Start> and For> fields together specify the master subset of selected records on which subsequent filtering and processing will occur. These field values are applied **separately** to **every** input file. Input records that fall outside the range of records selected by this subset will not be processed.

  See the FSU: Basic File Search panel for description of the Start> and For> input field values.

  The **Max>** field places a restriction on the total number of input records that may be processed **from all input files**. This number does not include input records bypassed due to a Start> option.

  For these fields to take affect, they must first be activated by selecting the appropriate option field(s).

Record Filter:
: Specifies a record filter to perform further subsetting on input file records selected for processing by the **Start>** and/or **For>** input fields.

  See the FSU: Basic File Search panel for description of the Filter>, File> and Member> input field values. The **Volume>** field allows specification of a DASD volume for uncataloged Filter data sets.

FSU (formatted): Record Selection

The **FSU (formatted): Record Selection** panel view provides the opportunity to enter formatted record selection criteria to be applied to **every** selected input file.

Any specified filter must operate on formatted record data. i.e. individual, named fields within the formatted record may be tested.

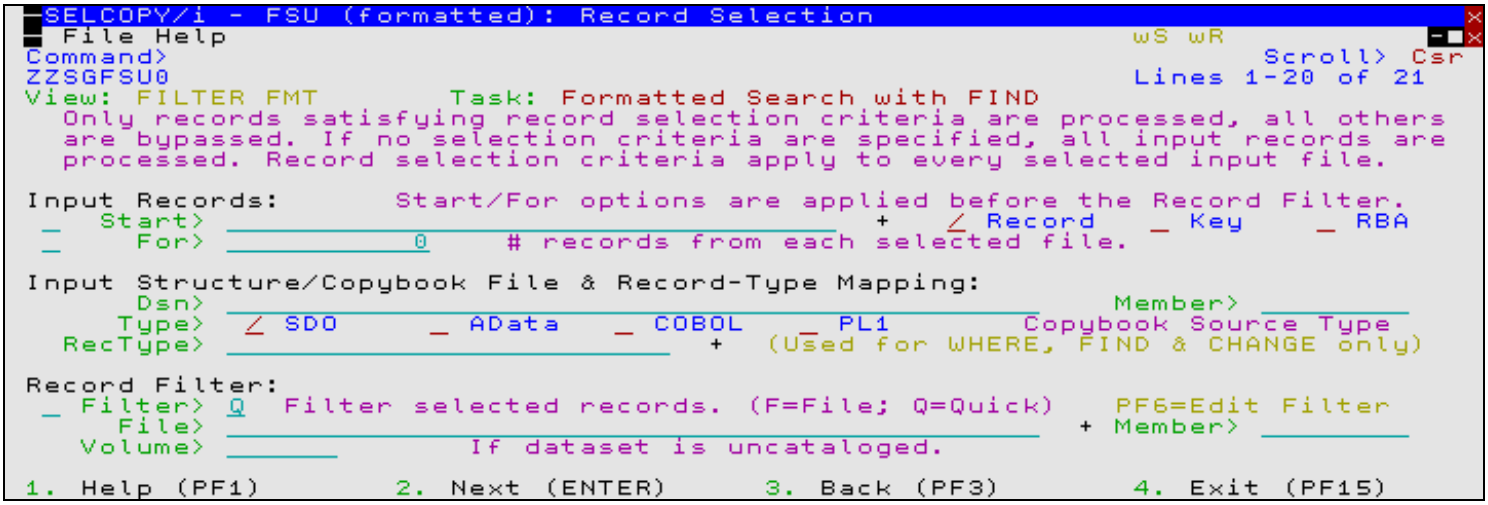


Figure 46. FSU (formatted): Record Selection.

Menu Bar Items

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Panel Fields

Input Records:
: The Start> and For> fields together specify the master subset of selected records on which subsequent filtering and processing will occur. These field values are applied **separately** to **every** input file. Input records that fall outside the range of records selected by this subset will not be processed.

  See the FSU: Basic File Search panel for description of the Start> and For> input field values.

  The **Max>** field places a restriction on the total number of input records that may be processed **from all input files**. This number does not include input records bypassed due to a Start> option.

  For these fields to take affect, they must first be activated by selecting the appropriate option field(s).

Input Structure/Copybook File & Record-Type Mapping:
: Fields which together specify a cataloged structure file (COBOL or PL1 Copybook, ADATA file or a SELCOPY/i SDO) and, if WHERE, FIND and/or CHANGE operations are to be performed, a mandatory record-type as defined within the structure. The structure may be a sequential data set or a PDS/PDSE library member.

  If a record filter is to be generated, this structure will be passed to the appropriate panel.

  Dsn>
  : Identifies the fully qualified data set name of a sequential data set or PDS/PDSE library containing a COBOL or PL1 Copybook, ADATA file or SDO structure.

    A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

  Member>
  : If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

    A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

  Type>
  : Indicate the type of structure (COBOL, PL1, ADATA or SDO).

RecType>
Identifies the name of a record-type record mapping defined within the structure. Records that satisfy this record-type selection criteria will be assigned this record-type and will be included in any WHERE, FIND and/or CHANGE operations.

The SDE: Select Record-Type panel will be automatically opened to display a selectable list of record-types if the structure contains multiple record types. Otherwise the single record-type will be inserted automatically.

Recompile>
If *Structure/Copybook overlay* refers to a COBOL, PL1 or ADATA source file (not an SDO) then a compile step must be performed in order to turn the source copybook into SELCOPYi's own internal structure (SDO) format. A temporary SDO will be created, lasting the duration of the SELCOPYi session, making subsequent reference to the same copybook during the session much faster.

If, however, the copybook (or any of its included components) is modified during the session, then a recompile of the source will normally be required.

For performance reasons **N** (meaning no recompile should occur) is the default.

Specify **Y** to force a copybook recompile each time the process is run. Alternatively type the primary command:
SD DROP *<copybook_name>*

Record Filter:
Specifies a record filter to perform further subsetting on input file records selected for processing by the **Start>** and/or **For>** input fields.

See the FSU: Basic File Search panel for description of the Filter>, File> and Member> input field values. The **Volume>** field allows specification of a DASD volume for uncataloged Filter data sets.

FSU (unformatted): Search records using WHERE expression

The **FSU (unformatted): Search records using WHERE expression** panel view is displayed only if the option to search unformatted records using an SDE expression has been requested. This may be for a search and report of matching records with or without a subsequent FIND operation, or as a record vetting operation prior to actioning a CHANGE operation.

The specified SDE expression must operate on unformatted record data. i.e. areas of the record may be tested based only on data position and length within the record.

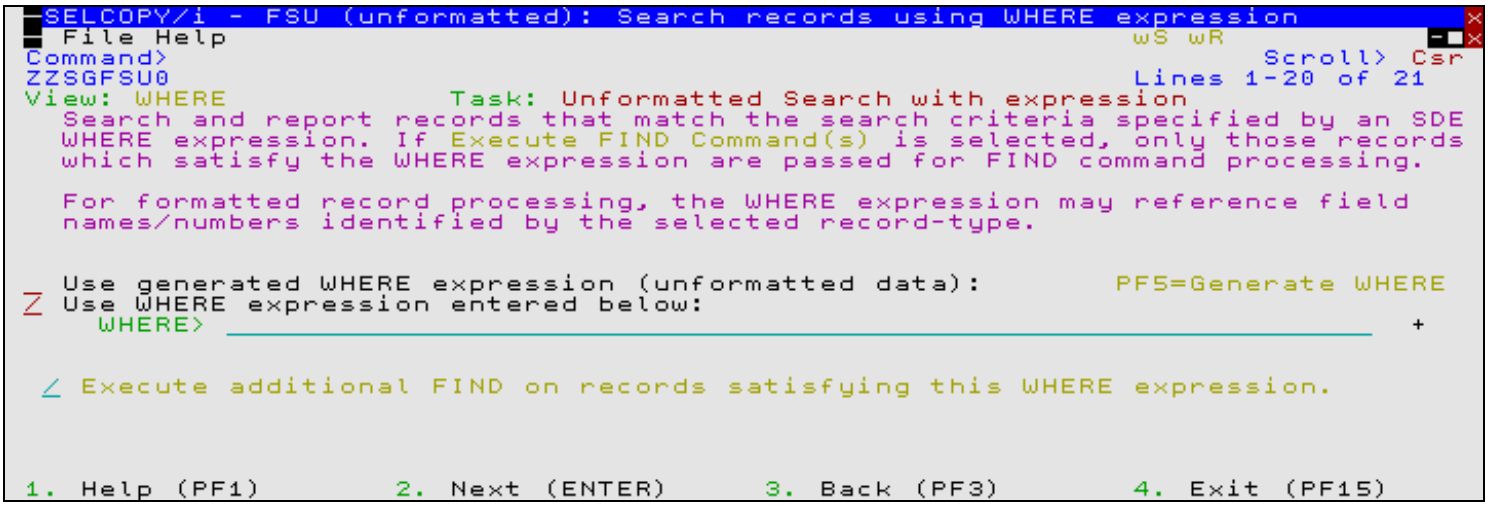


Figure 47. FSU (unformatted): Search records using WHERE expression.

Menu Bar Items

File
The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

Panel Fields

WHERE Expression Specification
Choose between 2 mutually exclusive options which identify the method by which a WHERE expression is to be specified:

Use generated WHERE expression (unformatted data):
Select this option to open the Unformatted Selection Criteria panel to generate an expression consisting of one or more sub-expressions separated by logical AND and/or logical OR operators.

The sub-expressions will test field position and length within unformatted record data. SELCOPY/i supports more complex expressions and function use than can be generated by this panel (see SDE Expressions.) If more complex expression is required, this panel may be used to generate the basic expression and later modified on editing the generated FSU utility command.

If this option is selected, <Enter> will display the **Unformatted Selection Criteria** panel. When a WHERE expression has been entered and the panel closed, primary command SELECT (assigned to the <F5> by default) may be used to redisplay the panel.

Use WHERE expression entered below:
Select this option to manually enter the required WHERE expression.

WHERE>
Enter a valid SDE expression. Beware that not validation is performed on this field.

EXPAND (assigned to <F14> by default) may be used to expand the input field and so enter a value which is longer than the visible input field area.

Execute additional FIND on records satisfying this WHERE expression.
Select this option if FSU is to perform a FIND operation on records that satisfy the supplied WHERE expression.

This allows an extra level of record searching for file search operations or for file change record vetting.

FSU (formatted): Search records using WHERE expression

The **FSU (formatted): Search records using WHERE expression** panel view is displayed only if the option to search formatted records using an SDE expression has been requested. This may be for a search and report of matching records with or without a subsequent FIND operation, or as a record vetting operation prior to actioning a CHANGE operation.

The specified SDE expression must operate on the formatted data belonging only to records that are assigned the nominated record-type. i.e. individual, named fields within the formatted record may be tested.

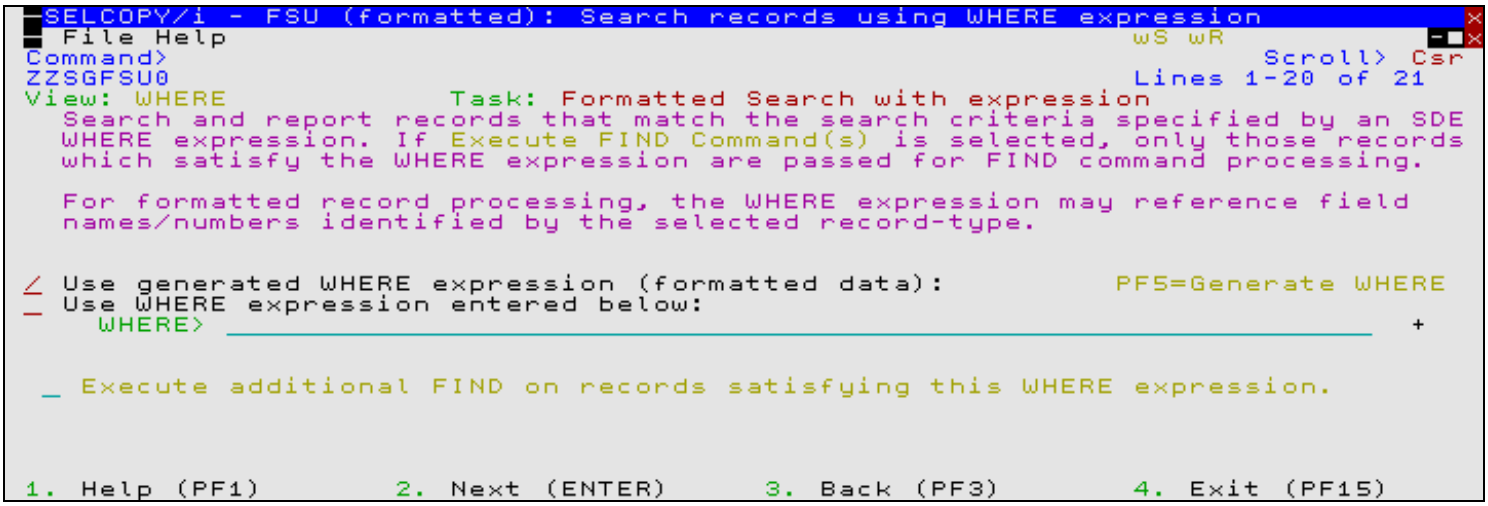


Figure 48. FSU (formatted): Search records using WHERE expression.

Menu Bar Items

File
The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

Panel Fields

WHERE Expression Specification
Choose between 2 mutually exclusive options which identify the method by which a WHERE expression is to be specified:

Use generated WHERE expression (formatted data):
Select this option to open the Formatted Record Expression panel to generate an expression involving formatted record fields as mapped by the record-type nominated in the FSU (formatted): Record Selection panel view.

SELCOPY/i supports more complex expressions and function use than can be generated by this panel (see SDE Expressions.) If more complex expression is required, this panel may be used to generate the basic expression and later modified on editing the generated FSU utility command.

If this option is selected, <Enter> will display the **Formatted Record Expression** panel. When a WHERE expression has been entered and the panel closed, primary command SELECT (assigned to the <F5> by default) may be used to redisplay the panel.

Use WHERE expression entered below:
Select this option to manually enter the required WHERE expression.

WHERE>
Enter a valid SDE expression. Beware that not validation is performed on this field.

EXPAND (assigned to <F14> by default) may be used to expand the input field and so enter a value which is longer than the visible input field area.

Execute additional FIND on records satisfying this WHERE expression.
Select this option if FSU is to perform a FIND operation on the formatted records fields for records that satisfy the supplied WHERE expression.

This allows an extra level of record searching for file search operations or for file change record vetting.

FSU (unformatted): Search Records using the FIND Command

The **FSU (unformatted): Search Records using the FIND Command** panel view is displayed only if the option to search unformatted records using one or more FIND operations has been requested. This may be for a search and report of matching records, or as a record vetting operation prior to actioning a CHANGE operation.

The specified FIND search values will operate on the unformatted record data. If column bounds are specified, these must correspond to positions within the input records.

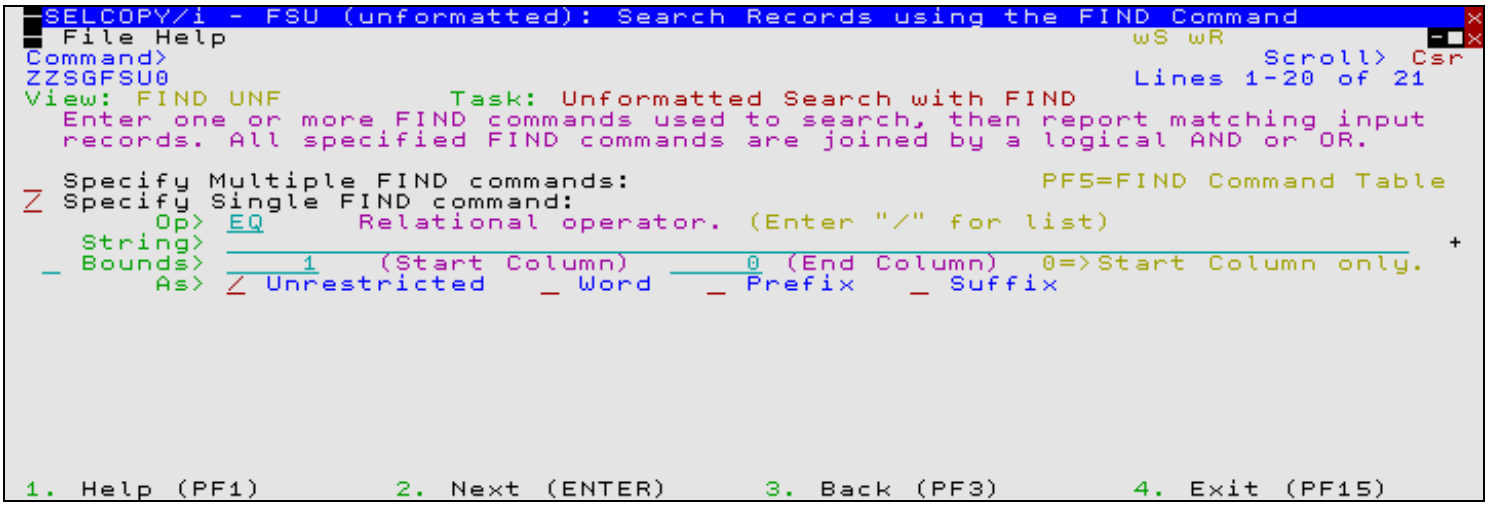


Figure 49. FSU (unformatted): Search records using the FIND Command.

Menu Bar Items

File
The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

Panel Fields

FIND Command Specification
Choose between 2 mutually exclusive options which identify the method by which FIND command(s) are to be specified:

Specify Multiple FIND commands:
Select this option to open the Multiple Find Commands (unformatted) panel to enter more than one FIND command operating on the unformatted record data.

The search values are compared with the unformatted record data and any specified column bounds are expressed as numeric positions in the record data. All generated FIND operations are applied to an input record and logical operator AND or OR is specified to indicate whether **all** or **any** of the FIND operations must be successful in order for the record to be selected.

If this option is selected, <Enter> will display the **Multiple Find Commands** panel. When a list of FIND commands has been entered and closed, primary command SELECT (assigned to the <F5> by default) may be used to redisplay the panel.

Specify Single FIND command:
Select this option if only one FIND command is required. The fields that follow together constitiute an edit FIND command used to search unformatted input file records.

See description of Search Options: in the **FSU: Basic File Search** panel for details of the **Op>**, **String>**, **Bounds>** and **As>** input fields.

FSU (formatted): Search Records using the FIND Command

The **FSU (formatted): Search Records using the FIND Command** panel view is displayed only if the option to search formatted records using one or more FIND operations has been requested. This may be for a search and report of matching records, or as a record vetting operation prior to actioning a CHANGE operation.

The specified FIND commands will operate on the formatted data belonging only to records that are assigned the nominated record-type. The search values apply to individual fields and are sensitive to field data-type. If column bounds are specified, these must correspond to field names within the formatted input record.

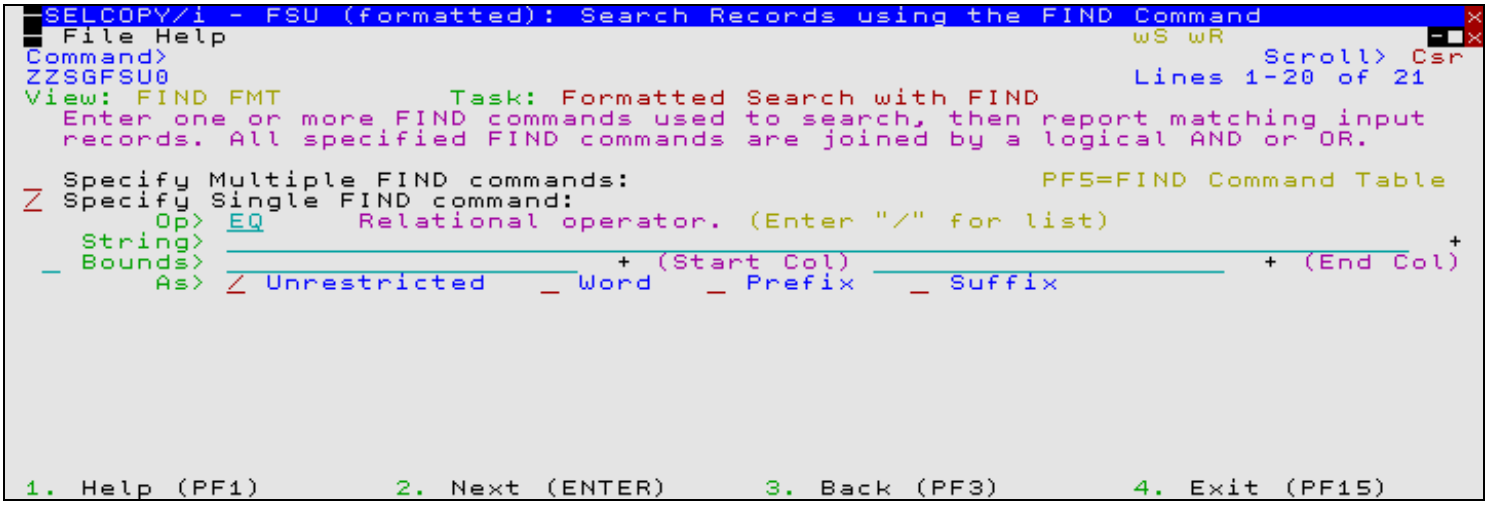


Figure 50. FSU (formatted): Search records using the FIND Command.

Menu Bar Items

File
The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

Panel Fields

FIND Command Specification
Choose between 2 mutually exclusive options which identify the method by which FIND command(s) are to be specified:

Specify Multiple FIND commands:
Select this option to open the Multiple Find Commands (formatted) panel to enter more than one FIND command operating on the formatted record data.

The search values are compared with the individual formatted fields within the record data and any specified column bounds are expressed as field names. All generated FIND operations are applied to an input record and logical operator AND or OR is specified to indicate whether **all** or **any** of the FIND operations must be successful in order for the record to be selected.

If this option is selected, <Enter> will display the **Multiple Find Commands** panel. When a list of FIND commands has been entered and closed, primary command SELECT (assigned to the <F5> by default) may be used to redisplay the panel.

Specify Single FIND command:
Select this option if only one FIND command is required. The fields that follow together constitiute an edit FIND command used to search formatted input file records.

Op>
Identify the relational operator used when comparing the field data against the search value.

If testing for a non-equality relationship using a character string search value, then the EBCDIC values assigned to characters in the search string and record data determine the relationship. (e.g. GT, LT)

Enter "/" to display a list of valid entries and a brief description.

String>
Specify the search value. This may be one of the following:

1. Applicable to fields of any data type, an unquoted numeric value. For numeric fields, the numeric search value and source field data are converted so that they are of the same data

type prior to performing an arithmetic comparison.

2. Applicable only to character fields, a quoted character string (case insensitive search). e.g. 'John Doe' is the same as 'JOHN DOE'.

3. Applicable only to character fields, a quoted character string prefixed with "C" (case sensitive search). e.g. C'John Doe'.

4. Applicable only to character fields, a quoted hexadecimal string prefixed with "X". e.g. X'00'.

5. Applicable only to character fields, a quoted picture string prefixed with "P". e.g. P'USER###'. See the FIND command for details on supported picture string special characters.

6. Applicable only to character fields, a quoted unix-style "Regular Expression" string prefixed with "R". e.g. R'[Pp][Ff](:d#)'
See Regular Expressions for details on supported regular expression string special characters.

Bounds>

Optionally activate the **Bounds>** input fields to specify the named record columns (fields) between which the search will occur. If not activated, the search will include all fields within the length of the input records.

Start Column

Specifies the start (or only) record field from which the scan for the search value will begin. Record data in fields that occur before this start column value is not searched.

This field corresponds to SDE FIND parameter *field_col/field_col1*.

End Column

Specifies the end record field. Only record data between the start and end columns will be searched.

If a 0 (zero) end column is specified then only the start column field is searched.

This field corresponds to SDE FIND parameter *filed_col2*.

As>

Applicable only to character fields (data-type "AN"), enter "/" in the appropriate radio button field to apply restrictions to the position of the search string within the field as follow:

Unrestricted	The search value may be found anywhere within the character field.
Word	The search value may only be found if it is complete word within the character field. i.e. the value must either be preceded by a non-alphanumeric character or begin at the first character of the field, and either precede a non-alphanumeric character or finish in the last character of the field.
Prefix	The search value may only be found if it occurs at the start of a word within the character field. i.e. the value must either be preceded by a non-alphanumeric character or begin at the first character of the field, and be followed by an alphanumeric character.
Suffix	The search value may only be found if it occurs at the end of a word within the character field. i.e. the value must be preceded by an alphanumeric character and must either precede a non-alphanumeric character or finish in the last character of the field.

These option fields correspond to SDE FIND parameters PREFIX, SUFFIX and WORD.

FSU (unformatted): Change record data using the CHANGE command

The **FSU (unformatted): Change record data using the CHANGE command** panel view is displayed only if the option to change data in unformatted records has been requested.

The specified CHANGE search and replace values will operate on the unformatted record data. If column bounds are specified, these must correspond to positions within the input records.

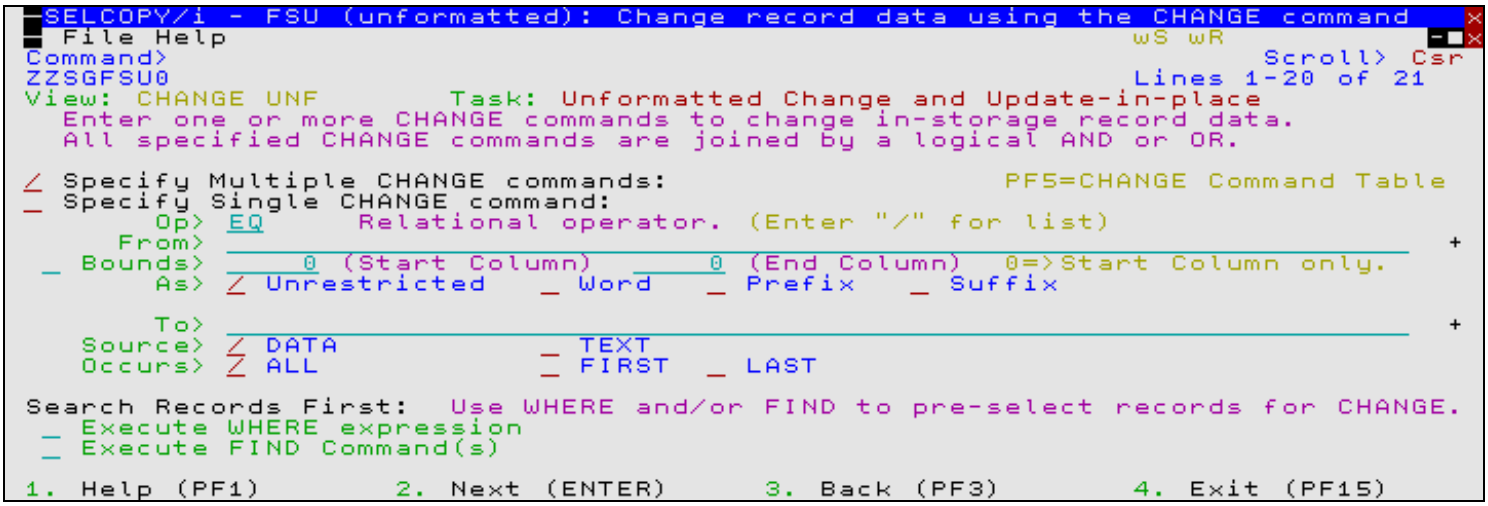


Figure 51. FSU (unformatted): Change record data using the CHANGE command.

Menu Bar Items

File
The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

Panel Fields

Change Command Specification
Choose between 2 mutually exclusive options which identify the method by which CHANGE command(s) are to be specified:

Specify Multiple CHANGE commands:
Select this option to open the Multiple Change Commands (unformatted) panel to enter more than one CHANGE command operating on the unformatted record data.

The search values are compared with the unformatted record data and any specified column bounds are expressed as numeric positions in the record data. All generated CHANGE operations are applied to an input record and logical operator AND or OR is specified to indicate whether **all** CHANGE operations or only the **first** successful CHANGE operation is to applied to the record data.

If this option is selected, <Enter> will display the **Multiple Change Commands** panel. When a list of CHANGE commands has been entered and closed, primary command SELECT (assigned to the <F5> by default) may be used to redisplay the panel.

Specify Single CHANGE command:
Select this option if only one CHANGE command is required. The fields that follow together constitiute an edit CHANGE command used to search for then replace a string value within unformatted input file records.

Op>
Identify the relational operator used when comparing the record data against the CHANGE operation search value.

If testing for a non-equality relationship using a character string search value, then the EBCDIC values assigned to characters in the search string and record data determine the relationship. (e.g. GT, LT)

Enter "/" to display a list of valid entries and a brief description.

From>
Specify the CHANGE operation search value. This may be one of the following:

1. An unquoted numeric value which will be processed as a character string.

2. A quoted character string (case insensitive search). e.g. 'John Doe' is the same as 'JOHN DOE'.
3. A quoted character string prefixed with "C" (case sensitive search). e.g. C'John Doe'.
4. A quoted hexadecimal string prefixed with "X". e.g. X'00'.
5. A quoted picture string prefixed with "P". e.g. P'USER###'.
   See the CHANGE command for details on supported search value picture string special characters.
6. A quoted unix-style "Regular Expression" string prefixed with "R". e.g. R'[Pp][Ff](:d#)'
   See Regular Expressions for details on supported regular expression string special characters.

Bounds>

Optionally activate the **Bounds>** input fields to specify the record columns between which both the search and replace values must occur in order to perform a successful CHANGE operation.

If not activated, the search will include all columns within the length of the input records. If an in-place update is to be performed, the replace string must not result in a change to the record length. However, if records are to be copied the replace string may extend the length of the record to a value not greater than the defined maximum record length.

Start Column

Specifies the start (or only) record column from which the scan for the CHANGE operation search string will begin. Record data in positions or fields that occur before this start column value is excluded from the CHANGE operation.

This field corresponds to SDE CHANGE parameter *pos1*.

End Column

Specifies the end record data column beyond which no part of the search value may be found and no part of the replace string may occupy. Only record data between the start and end columns will be searched and potentially replaced.

If a 0 (zero) end column is specified then the end column is the start column number plus the length of the search value minus 1. i.e. the search value will only be found if it begins in the start column.

This field corresponds to SDE CHANGE parameter *pos2*.

As>

Enter "/" in the appropriate radio button field to apply restrictions to the position of the search string within record data as follow:

Unrestricted	The search value may be found anywhere within the specified column bounds of the input records.
Word	The search value may only be found if it is a complete word which falls within the specified column bounds of the input records. i.e. the value must either be preceded by a non-alphanumeric character or begin at the start column bound, and either precede a non-alphanumeric character or finish in the end column bound.
Prefix	The search value may only be found if it occurs at the start of a word and falls within the specified column bounds of the input records. i.e. the value must either be preceded by a non-alphanumeric character or begin at the start column bound, and be followed by an alphanumeric character.
Suffix	The search value may only be found if it occurs at the end of a word and falls within the specified column bounds of the input records. i.e. the value must be preceded by an alphanumeric character and must either precede a non-alphanumeric character or finish in the end column bound.

These option fields correspond to SDE CHANGE parameters PREFIX, SUFFIX and WORD.

To>

Specify the CHANGE operation replace value. This may be specified in any of the formats described for the search value above.

For a replace value specified as a quoted picture string, see the CHANGE command for details on supported replace value picture string special characters.

If the CHANGE operation search value was specified as a regular expression involving tagged sub-expressions, then the replace string may contain tag references to these tagged sub-expressions.

Source>

Identifies the format of the record source character data:

DATA

Indicates that records are to be treated as data so that inserting the replace value performs no special treatment of any multiple, consecutive blanks that occur to the right of the replaced data.

If the length of the replace value is greater than that of the search value, all data to the right of the replaced data will be shifted right.

If the length of the replace value is less than that of the search value, all data to the right of the replaced data will be shifted left.

`TEXT`
Indicates that records are to be treated as formatted character text so that, in an effort to maintain the record positions of non-blank text to the right of the changed text, when inserting the replacement text, consecutive blanks that occur to the right of this text are absorbed or added as approriate.

If the length of the replace value is greater than that of the search value, consecutive balnks to the right of the replaced text will be absorbed before non-blank text is shifted right. Note that blank delimitation of non-blank text is preserved.

If the length of the replace value is less than that of the search value, then blanks will be inserted immediately before the first blank character occurring to the right of the replaced text. The number of blanks inserted will be equal to the difference between the lengths of the search and replace values.

`Occurs>`
Identifies the occurrence of the search value within the input record to which the CHANGE operation will apply:

`ALL`
Attempt to change all occurrences of the search value found within the input record.

`FIRST`
Attempt to change only the first occurrence of the search value found within the input record.

`LAST`
Attempt to change only the last occurrence of the search value found within the input record.

`File Update:`
Applicable only to File Change and Update, this option indicates whether or not the input files are to be updated when the utility is executed.

If this option is set to "Yes", the utility will indicate perform the update in place of changed record data in the input files. Note that, before the panel executes the utility, a pop-up message box will warn the user of the impending record updates and allow the operation to be cancelled. To recover record updates, please see the File Update Undo (FSUUNDO) utility.

If this option is set to "No", the utility report output will indicate which input records would be updated without actually performing the file update. It is recommended that this is option is seleceted first in order to review the utility report before performing the actual update operation.

`Search Records First:`
If record vetting is to be performed on selected records prior to attempting the CHANGE operations, select one or both of the record search methods to be used.

`Execute WHERE expression`
If this option is selected, the FSU unformatted Search records using a WHERE expression panel will be displayed next.

`Execute FIND Command(s)`
If this option is selected, the FSU unformatted Search records using the FIND command panel will be displayed next.

If both options are selected, the **Execute WHERE expression** panel will be displayed first.

FSU (formatted): Change record data using the CHANGE command

The **FSU (formatted): Change record data using the CHANGE command** panel view is displayed only if the option to change data in formatted records has been requested.

The specified CHANGE commands will operate on the formatted data belonging only to records that are assigned the nominated record-type. The search and replace values apply to individual fields and are sensitive to field data-type. If column bounds are specified, these must correspond to field names within the formatted input record.

If record data is also being remapped, then the CHANGE operations will apply to the data in the remapped output record, i.e. not the source input data record.

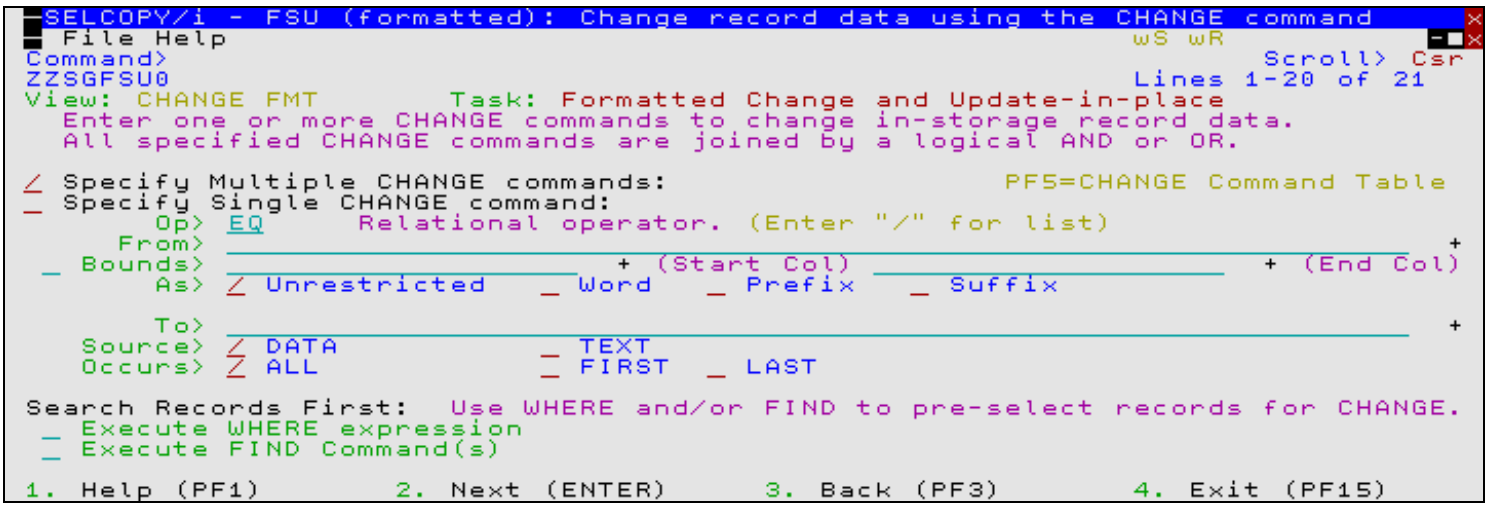


Figure 52. FSU (formatted): Change record data using the CHANGE command.

Menu Bar Items

File
The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

Panel Fields

Change Command Specification
Choose between 2 mutually exclusive options which identify the method by which CHANGE command(s) are to be specified:

Specify Multiple CHANGE commands:
Select this option to open the Multiple Change Commands (formatted) panel to enter more than one CHANGE command operating on the formatted record data.

The search values are compared with the individual formatted fields within the record data, any specified column bounds are expressed as field names and replace values are converted into a data type that matches that of the source field. All generated CHANGE operations are applied to an input record and logical operator AND or OR is specified to indicate whether **all** CHANGE operations or only the **first** successful CHANGE operation is to applied to the record data.

If this option is selected, <Enter> will display the **Multiple Change Commands** panel. When a list of CHANGE commands has been entered and closed, primary command SELECT (assigned to the <F5> by default) may be used to redisplay the panel.

Specify Single CHANGE command:
Select this option if only one CHANGE command is required. The fields that follow together constitiute an edit CHANGE command used to search for then replace a string value within formatted input file records.

Op>
Identify the relational operator used when comparing the field data against the CHANGE search value.

If testing for a non-equality relationship using a character string search value, then the EBCDIC values assigned to characters in the search string and record data determine the relationship. (e.g. GT, LT)

Enter "/" to display a list of valid entries and a brief description.

From>

Specify the CHANGE operation search value. This may be one of the following:

1. Applicable to fields of any data type, an unquoted numeric value. For numeric fields, the numeric search value and source field data are converted so that they are of the same data type prior to performing an arithmetic comparison.
2. Applicable only to character fields, a quoted character string (case insensitive search). e.g. 'John Doe' is the same as 'JOHN DOE'.
3. Applicable only to character fields, a quoted character string prefixed with "C" (case sensitive search). e.g. C'John Doe'.
4. Applicable only to character fields, a quoted hexadecimal string prefixed with "X". e.g. X'00'.
5. Applicable only to character fields, a quoted picture string prefixed with "P". e.g. P'USER###'. See the CHANGE command for details on supported search value picture string special characters.
6. Applicable only to character fields, a quoted unix-style "Regular Expression" string prefixed with "R". e.g. R'[Pp][Ff](:d#)'
   See Regular Expressions for details on supported regular expression string special characters.

Bounds>

Optionally activate the **Bounds>** input fields to specify the named record columns (fields) between which the CHANGE operation will occur.

If not activated, the operation will include all columns within the length of the input records. If an in-place update is to be performed, the replace string must not result in a change to the record length. However, if records are to be copied/remapped the replace string may extend the length of the record to a value not greater than the defined maximum output record length.

Start Column

Specifies the start (or only) record field from which the scan for the CHANGE operation search string will begin. Record data in fields that occur before this start column value is not included.

This field corresponds to SDE CHANGE parameter *field_col/field_col1*.

End Column

Specifies the end record data field column. Only record data between the start and end columns will be searched and potentially replaced.

If no end column is specified then only the start column field is searched.

This field corresponds to SDE CHANGE parameter *filed_col2*.

As>

Applicable only to character fields (data-type "AN"), enter "/" in the appropriate radio button field to apply restrictions to the position of the search value within the field as follow:

Unrestricted	The search value may be found anywhere within the character field.
Word	The search value may only be found if it is complete word within the character field. i.e. the value must either be preceded by a non-alphanumeric character or begin at the first character of the field, and either precede a non-alphanumeric character or finish in the last character of the field.
Prefix	The search value may only be found if it occurs at the start of a word within the character field. i.e. the value must either be preceded by a non-alphanumeric character or begin at the first character of the field, and be followed by an alphanumeric character.
Suffix	The search value may only be found if it occurs at the end of a word within the character field. i.e. the value must be preceded by an alphanumeric character and must either precede a non-alphanumeric character or finish in the last character of the field.

These option fields correspond to SDE CHANGE parameters PREFIX, SUFFIX and WORD.

To>

Specify the CHANGE operation replace value. This may be specified in any of the formats described for the search value above. For numeric fields, the replace value will be converted into the appropriate source data format.

For a replace value specified as a quoted picture string, see the CHANGE command for details on supported replace value picture string special characters.

If the CHANGE operation search value was specified as a regular expression involving tagged sub-expressions, then the replace string may contain tag references to these tagged sub-expressions.

Source>
For character fields only, these options identify the format of the source character data:

DATA
Indicates that character fields are to be treated as data so that inserting the replace value performs no special treatment of any multiple, consecutive blanks that occur to the right of the replaced data.

If the length of the replace value is greater than that of the search value, all data to the right of the replaced data will be shifted right.

If the length of the replace value is less than that of the search value, all data to the right of the replaced data will be shifted left.

TEXT
Indicates that character fields are to be treated as formatted character text so that, in an effort to maintain the position of non-blank text which appear to the right of the changed text, when inserting the replacement text, consecutive blanks that occur to the right of this text are absorbed or added as approriate.

If the length of the replace value is greater than that of the search value, consecutive balnks to the right of the replaced text will be absorbed before non-blank text is shifted right. Note that blank delimitation of non-blank text is preserved.

If the length of the replace value is less than that of the search value, then blanks will be inserted immediately before the first blank character occurring to the right of the replaced text. The number of blanks inserted will be equal to the difference between the lengths of the search and replace values.

Occurs>
Identifies the occurrence of the search value within the input record to which the CHANGE operation will apply:

ALL
Attempt to change all occurrences of the search value found within the input record fields.

FIRST
Attempt to change only the first occurrence of the search value found within the input record fields

LAST
Attempt to change only the last occurrence of the search value found within the input record fields

File Update:
Applicable only to File Change and Update, this option indicates whether or not the input files are to be updated when the utility is executed.

If this option is set to "Yes", the utility will indicate perform the update in place of changed record data in the input files. Note that, before the panel executes the utility, a pop-up message box will warn the user of the impending record updates and allow the operation to be cancelled. To recover record updates, please see the File Update Undo (FSUUNDO) utility.

If this option is set to "No", the utility report output will indicate which input records would be updated without actually performing the file update. It is recommended that this is option is seleceted first in order to review the utility report before performing the actual update operation.

Search Records First:
If record vetting is to be performed on selected records prior to attempting the CHANGE operations, select one or both of the record search methods to be used.

Execute WHERE expression
If this option is selected, the FSU formatted Search records using a WHERE expression panel will be displayed next.

Execute FIND Command(s)
If this option is selected, the FSU formatted Search records using the FIND command panel will be displayed next.

If both options are selected, the **Execute WHERE expression** panel will be displayed first.

FSU: Copy Selected records to an Output File

The **FSU: Copy Selected records to an Output File** panel view is displayed following specification of any WHERE, FIND and CHANGE operation parameters and when records are to be copied to a single output file. The output file may be a sequential, VSAM or GDG data set, an HFS file, or a PDS/PDSE library member.

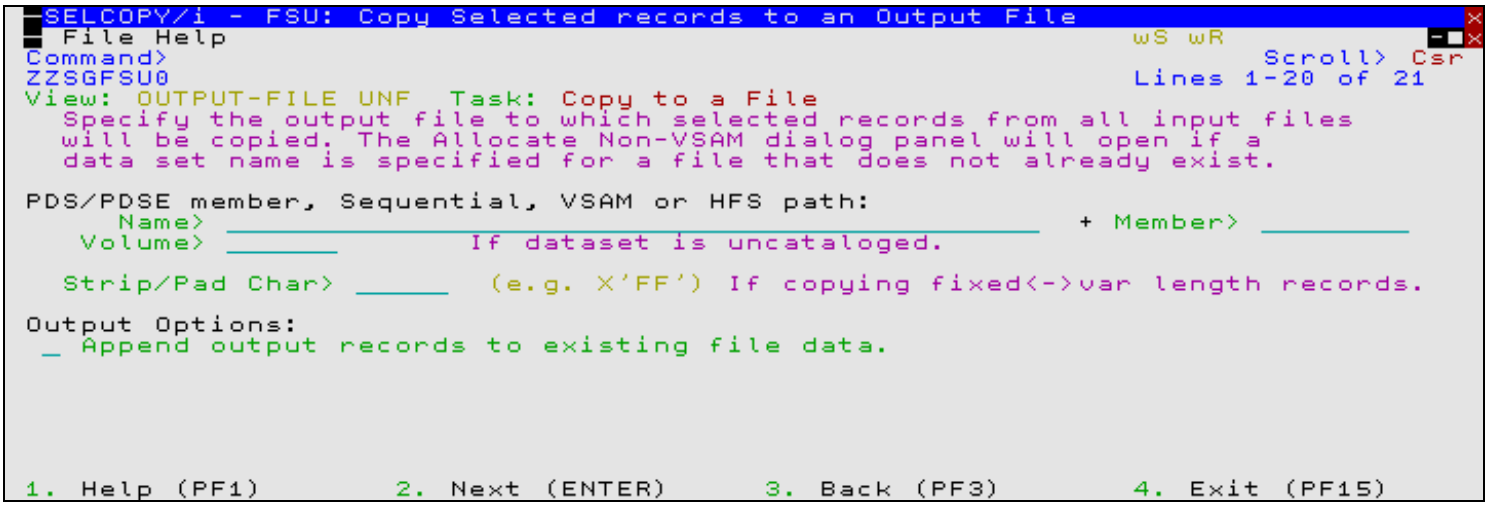


Figure 53. FSU: Copy Selected records to an Output File.

Menu Bar Items

File
The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

Panel Fields

PDS/PDSE member, Sequential, VSAM or HFS path:
Input fields which together identify a single output sequential or VSAM data set, HFS file or PDS/PDSE library member.

Name>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

If a DSN is specified for a data set that does not already exist, the Allocate non-VSAM data set dialog will be opened to allocate the new output file.

Member>
If the Name> field contains the DSN of a PDS/PDSE library, then this field specifies the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Volume>
Specifies the name of the output data set volume. This is required only if output is to an uncataloged data set.

Max>
Limits the total number of records that may be written to the output file. A value of zero indicates that no back stop limit should be imposed to curtail the prevailing selection criteria.

For this field to take affect, it must first be activated by placing a "/" in the option field to left hand side.

Strip/Pad Char>
Specifies a single byte pad or strip character implemeted as follows:

1. When copying fixed length records to variable length, contiguous trailing characters at the end of each record that match the specified character *char* are to be stripped.
   Default is not to strip trailing characters.

2. When copying variable length records to fixed length, records are to be padded with the specified character *char*. Default is to pad with the blank character (X'40').

The strip/pad character *char* may be specified in character, hexadecimal or binary string notation of length 1 byte. (e.g. 'A', C'a', X'40', B'11110001'.)

Output Options:

Miscellaneous options for output to a single file.

Append output records to existing file data.

Append copied records to existing data in the output file.

Delimit library members written to the output file.

Applicable only when copying members from a PDS/PDSE library to the output file. This option causes a delimiter record to be written before the data belonging to each member copied.

Enter blank to select from a list of available options, each describing the format of the delimiter record to be inserted.

Option	Delimiter Record Format
0	No delimiter record (Default).
1	"./ ADD NAME=<member>"
2	"*>>>>>> DSN=lib.name(member) <<<<<<*"

FSU: Remap Fields in Selected records to an Output File

The **FSU: Remap Fields in Selected records to an Output File** panel view is displayed following specification of any WHERE, FIND and CHANGE operation parameters and when fields in formatted records are to be remapped to records in a single output file. The output file may be a sequential, VSAM or GDG data set, an HFS file, or a PDS/PDSE library member.

Specification of an output structure is mandatory so that fields of **"matched"** field names belonging to record-type definitions of **"matched"** record-type name within the input and output structures can be remapped.

A formatted record field remap operation requires a match-up process on the input and output structures for both Record-types, and Fields within those record-types.

Record-type and Field names that are identical in both the input and output structure are automatically matched.

Type the MAP primary command to interactively match-up Record-types and Field that are not identically named.

A record-type field definition in the input structure that corresponds to a "matched" record-type and field definition in the output structure may be of different source data types and may be located at different offsets within the record.

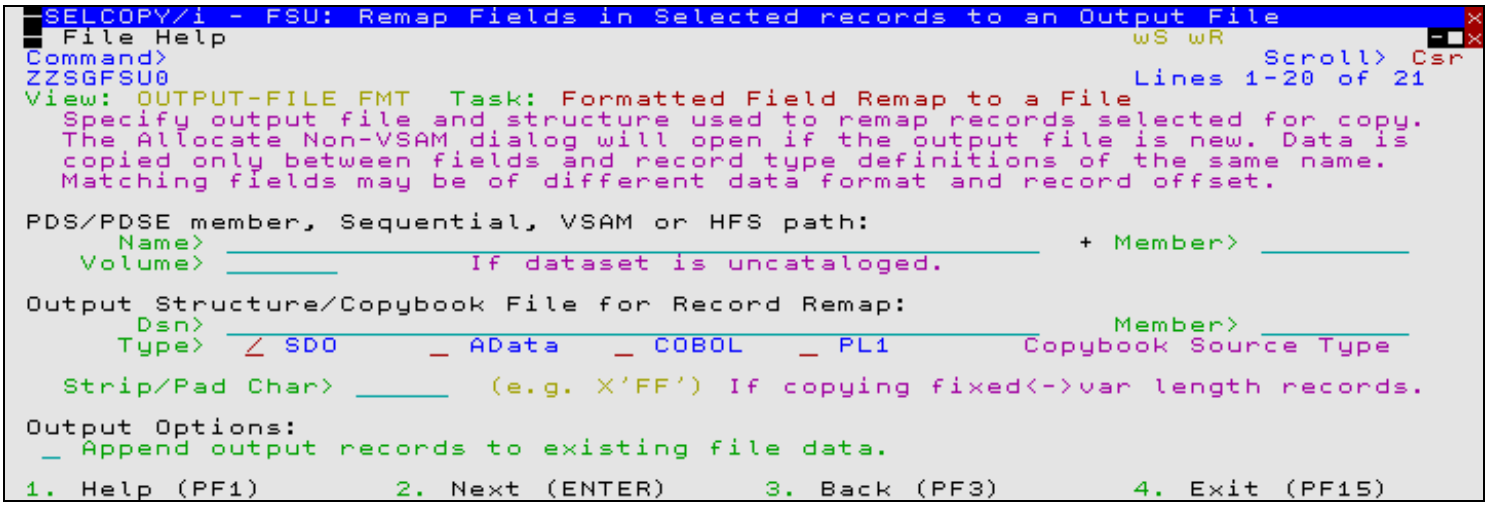


Figure 54. FSU: Remap Fields in Selected records to an Output File.

Menu Bar Items

File
The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

Panel Fields

PDS/PDSE member, Sequential, VSAM or HFS path:
Input fields which together identify a single output sequential or VSAM data set, HFS file or PDS/PDSE library member.

Name>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

If a DSN is specified for a data set that does not already exist, the Allocate non-VSAM data set dialog will be opened to allocate the new output file.

Member>
If the Name> field contains the DSN of a PDS/PDSE library, then this field specifies the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Volume>
Specifies the name of the output data set volume. This is required only if output is to an uncataloged data set.

Max>
Limits the total number of records that may be written to the output file. A value of zero indicates that no back stop limit should be imposed to curtail the prevailing selection criteria.

For this field to take affect, it must first be activated by placing a "/" in the option field to left hand side.

Output Structure/Copybook File for Record Remap:
Mandatory input fields which together specify a cataloged structure file (COBOL or PL1 Copybook, ADATA file or a SELCOPY/i SDO.) The structure may be a sequential data set or a PDS/PDSE library member.

Record-type mappings defined in this structure, which have the same name as record-types defined in the input structure, will be used to remap field data in records to which they are assigned.

Dsn>
Identifies the fully qualified data set name of an existing sequential data set or PDS/PDSE library containing a COBOL or PL1 Copybook, ADATA file or SDO structure.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Type>
Indicate the type of structure (COBOL, PL1, ADATA or SDO).

Recompile>
If *Structure/Copybook overlay* refers to a COBOL, PL1 or ADATA source file (not an SDO) then a compile step must be performed in order to turn the source copybook into SELCOPYi's own internal structure (SDO) format. A temporary SDO will be created, lasting the duration of the SELCOPYi session, making subsequent reference to the same copybook during the session much faster.

If, however, the copybook (or any of its included components) is modified during the session, then a recompile of the source will normally be required.

For performance reasons **N** (meaning no recompile should occur) is the default.

Specify **Y** to force a copybook recompile each time the process is run. Alternatively type the primary command:
SD DROP *<copybook_name>*

Strip/Pad Char>
Specifies a single byte pad or strip character implemeted as follows:

1. When copying fixed length records to variable length, contiguous trailing characters at the end of each record that match the specified character *char* are to be stripped.
   Default is not to strip trailing characters.

2. When copying variable length records to fixed length, records are to be padded with the specified character *char*.
   Default is to pad with the blank character (X'40').

The strip/pad character *char* may be specified in character, hexadecimal or binary string notation of length 1 byte. (e.g. 'A', C'a', X'40', B'11110001'.)

Output Options:
Miscellaneous options for output to a single file.

Append output records to existing file data.
Append copied records to existing data in the output file.

Delimit library members written to the output file.
Applicable only when copying members from a PDS/PDSE library to the output file. This option causes a delimiter record to be written before the data belonging to each member copied.

Enter blank to select from a list of available options, each describing the format of the delimiter record to be inserted.

Option	Delimiter Record Format
0	No delimiter record (Default).
1	"./ ADD NAME=<member>"
2	"*>>>>>> DSN=lib.name(member) <<<<<<*"

FSU: Copy Selected records to Output Library Members

The **FSU: Copy Selected records to Output Library Members** panel view is displayed following specification of any WHERE, FIND and CHANGE operation parameters and when input members are to be copied to members of the same name in an output PDS/PDSE library. i.e. Library Copy.

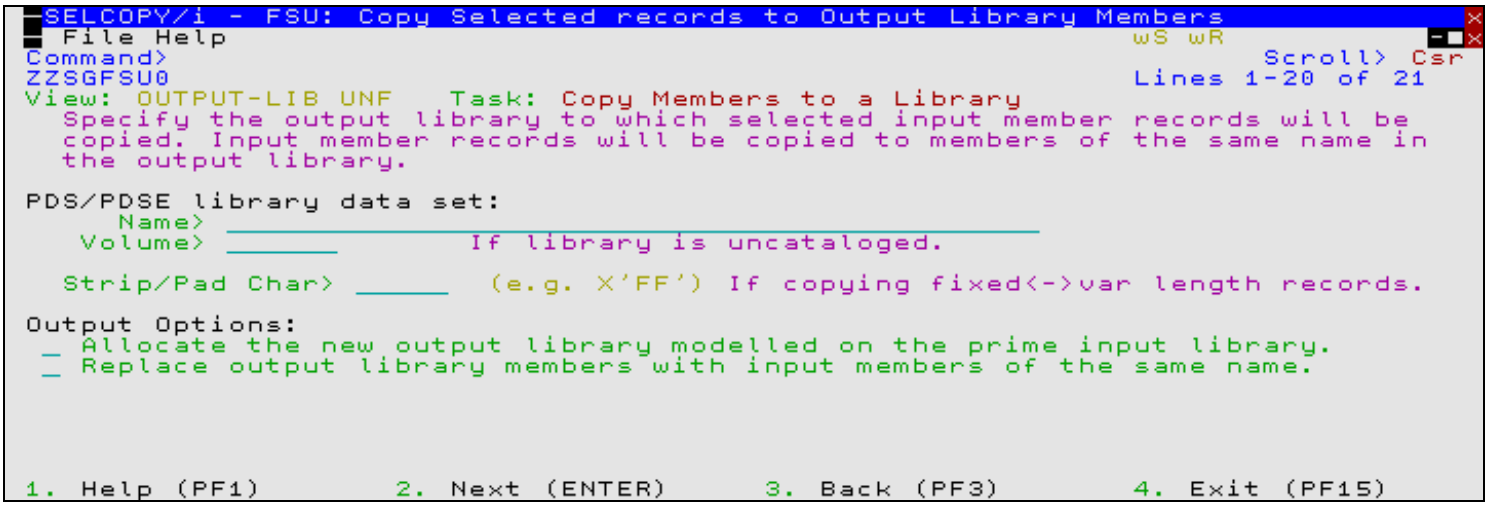


Figure 55. FSU: Copy Selected records to Output Library Members.

Menu Bar Items

File
The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

Panel Fields

PDS/PDSE library data set:
Input fields which together identify a single output PDS/PDSE library to which input library members will be copied.

Name>
Identifies the fully qualified data set name of a new or existing PDS/PDSE library.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

If the Allocate the new output library modelled on the prime input library option is **not** set and the DSN specified is for a data set that does not already exist, then the Allocate non-VSAM data set dialog will be opened to allocate the new output library.

Volume>
Specifies the name of the output data set volume. This is required only if output is to an uncataloged library data set.

Strip/Pad Char>
Specifies a single byte pad or strip character implemeted as follows:

1. When copying fixed length records to variable length, contiguous trailing characters at the end of each record that match the specified character *char* are to be stripped.
   Default is not to strip trailing characters.

2. When copying variable length records to fixed length, records are to be padded with the specified character *char*.
   Default is to pad with the blank character (X'40').

The strip/pad character *char* may be specified in character, hexadecimal or binary string notation of length 1 byte. (e.g. 'A', C'a', X'40', B'11110001'.)

Output Options:
Miscellaneous options for output library members.

Allocate the new output library modelled on the prime input library
If the output library does not already exist, allocate it modelled on the input library.

If members are copied from more than one library, then the output library will be modelled on the first input library read

Replace output library members with input members of the same name.
If an existing member in the output library has the same name as a member to be copied from an input library, it will be overwritten. In the case where members of the same name are copied from multiple input libaries, then the output library will contain the member copied last.

FSU: Remap Fields in Selected records to Output Library Members

The **FSU: Remap Fields in Selected records to Output Library Members** panel view is displayed following specification of any WHERE, FIND and CHANGE operation parameters and when fields in formatted library member records are to be remapped to records in members of the same name in an output PDS/PDSE library. i.e. Library Remap.

Specification of an output structure is mandatory so that fields of **"matched"** field names belonging to record-type definitions of **"matched"** record-type name within the input and output structures can be remapped.

A formatted record field remap operation requires a match-up process on the input and output structures for both Record-types, and Fields within those record-types.

Record-type and Field names that are identical in both the input and output structure are automatically matched.

Type the MAP primary command to interactively match-up Record-types and Field that are not identically named.

A record-type field definition in the input structure that corresponds to a "matched" record-type and field definition in the output structure may be of different source data types and may be located at different offsets within the record.

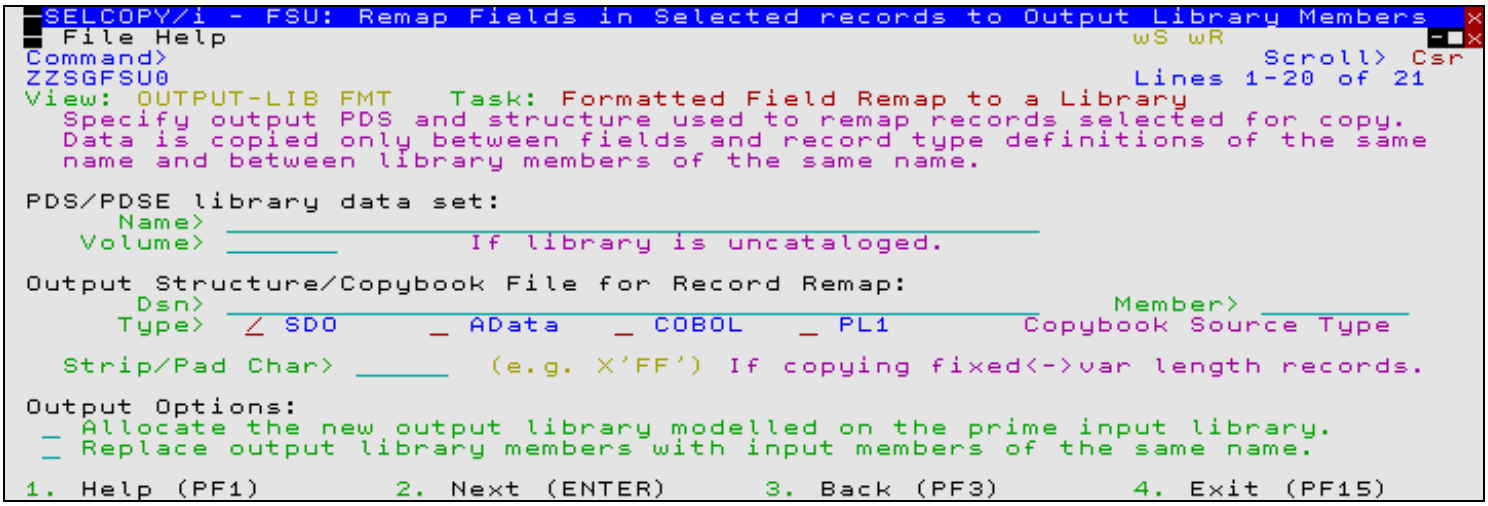


Figure 56. FSU: Remap Fields in Selected records to Output Library Members.

Menu Bar Items

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Panel Fields

PDS/PDSE library data set:
: Input fields which together identify a single output PDS/PDSE library to which records from input library members will be remapped.

  Name>
  : Identifies the fully qualified data set name of a new or existing PDS/PDSE library.

    A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

    If the Allocate the new output library modelled on the prime input library option is **not** set and the DSN specified is for a data set that does not already exist, then the Allocate non-VSAM data set dialog will be opened to allocate the new output library.

  Volume>
  : Specifies the name of the output data set volume. This is required only if output is to an uncataloged library data set.

Output Structure/Copybook File for Record Remap:
: Mandatory input fields which together specify a cataloged structure file (COBOL or PL1 Copybook, ADATA file or a SELCOPY/i SDO.) The structure may be a sequential data set or a PDS/PDSE library member.

  Record-type mappings defined in this structure, which have the same name as record-types defined in the input structure, will be used to remap field data in records to which they are assigned.

Dsn>
Identifies the fully qualified data set name of an existing sequential data set or PDS/PDSE library containing a COBOL or PL1 Copybook, ADATA file or SDO structure.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Type>
Indicate the type of structure (COBOL, PL1, ADATA or SDO).

Recompile>
If *Structure/Copybook overlay* refers to a COBOL, PL1 or ADATA source file (not an SDO) then a compile step must be performed in order to turn the source copybook into SELCOPYi's own internal structure (SDO) format. A temporary SDO will be created, lasting the duration of the SELCOPYi session, making subsequent reference to the same copybook during the session much faster.

If, however, the copybook (or any of its included components) is modified during the session, then a recompile of the source will normally be required.

For performance reasons **N** (meaning no recompile should occur) is the default.

Specify **Y** to force a copybook recompile each time the process is run. Alternatively type the primary command:
SD DROP *<copybook_name>*

Strip/Pad Char>
Specifies a single byte pad or strip character implemeted as follows:

1. When copying fixed length records to variable length, contiguous trailing characters at the end of each record that match the specified character *char* are to be stripped.
   Default is not to strip trailing characters.

2. When copying variable length records to fixed length, records are to be padded with the specified character *char*.
   Default is to pad with the blank character (X'40').

The strip/pad character *char* may be specified in character, hexadecimal or binary string notation of length 1 byte. (e.g. 'A', C'a', X'40', B'11110001'.)

Output Options:
Miscellaneous options for output library members.

Allocate the new output library modelled on the prime input library
If the output library does not already exist, allocate it modelled on the input library.

If members are copied from more than one library, then the output library will be modelled on the first input library read

Replace output library members with input members of the same name.
If an existing member in the output library has the same name as a member to be copied from an input library, it will be overwritten. In the case where members of the same name are copied from multiple input libaries, then the output library will contain the member copied last.

FSU: Report File

The **FSU: Report File** panel view is the last view displayed in the sequence. It provides the opportunity to optionally suppress generation of the utility output report, suppress automatic display of the output report following execution and to select the destination of the output report.

The FSU report is a structured data file designed to be browsed (not printed) using a SELCOPY/i structure definition object (SDO), which is also generated by the utility.

The associated SDO fileid is constructed simply by adding '.SDO' to the DSN of the sequential or PDS/PDSE DSN used for the report output. The report DSN is therefore restricted to 40 bytes in length. e.g. If *fileid* is ZX1234.SELCOPYI.FSU.REPORT(XYZ001), the allocated SDO is ZX1234.SELCOPYI.FSU.REPORT.SDO(XYZ001).

Pressing <Enter> in this panel will execute the generated FSU command in the foreground.

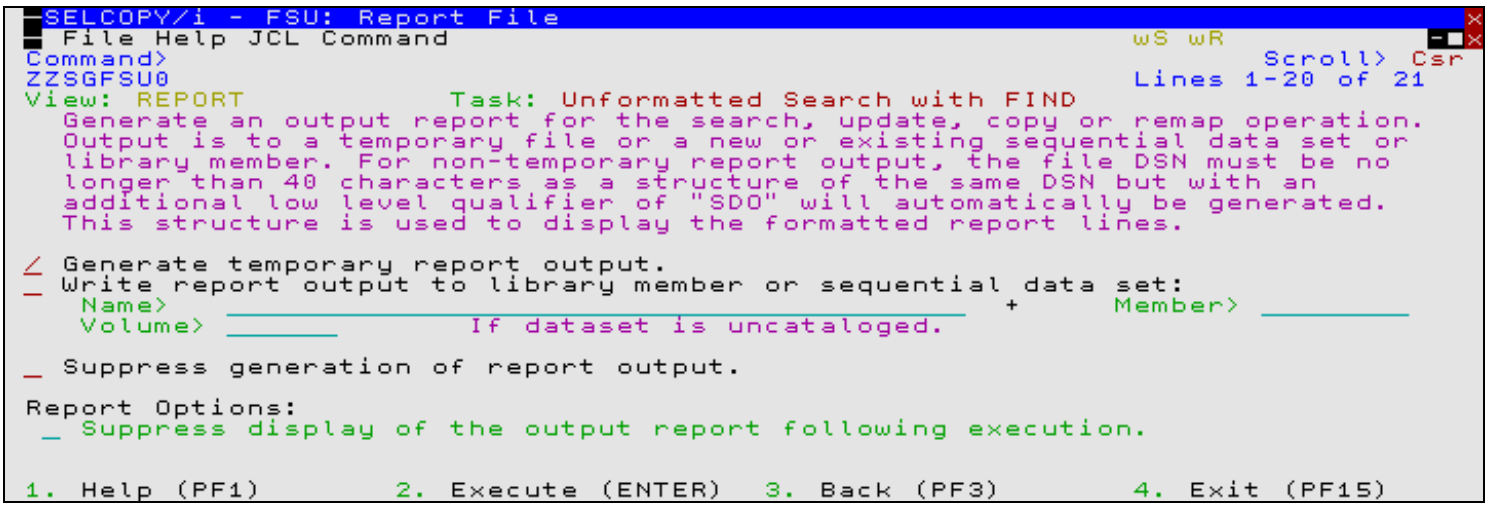


Figure 57. FSU: Report File.

Menu Bar Items

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

JCL
: Generate a JCL job stream that executes the **SDEAMAIN** program with input (SDEIN) containing the FSU command generated for the specified panel field values.

  The job stream is displayed in a temporary text edit view and may be submitted to batch using the SUBMIT command.

Command
: Generate the FSU command line syntax for field entries specified by the user and display it in a temporary CMX file text edit view. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

Panel Fields

Report Output Specification
: Choose between 3 mutually exclusive options which identify how the output report is to be created:

  Generate temporary report output.
  : Select this option to write report output and its accompanying SDO to temporary, in-storage files of DSN "*user*.FSU.Dyyyyddd.Thhmmss" and "*user*.FSU.Dyyyyddd.Thhmmss.SDO" respectively.

    Neither file is written to DASD volume unless the user elects to do so on exit of the report using **FSUEND**.

  Write report output to library member or sequential data set:
  : Select this option to write report output and its accompanying SDO to a sequential DASD data set or PDS/PDSE library member. This option should be used if generating JCL for batch SDEAMAIN execution.

    If the specified report file and/or its accompanying SDO file do not already exist, then they will automatically be allocated by FSU relying on SMS ACS to select a suitable storage group of eligable DASD volumes. The report file is allocated using DCB geometry RECFM=VB, LRECL=32756, BLKSIZE=0 and a space allocation of TRACKS(150,75). The SDO file is allocated using DCB geometry RECFM=VB, LRECL=16380, BLKSIZE=0 and

a space allocation of TRACKS(2,2).

Input fields which together identify a single output sequential data set or PDS/PDSE library member.

Name>
Identifies the fully qualified data set name of length no longer than 40 characters.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Name> field contains the DSN of a PDS/PDSE library, then this field specifies the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Volume>
Specifies the name of the report output data set volume. This is required only if output is to an uncataloged data set.

Suppress generation of report output.
Select this option to suppress generation of report output. This may only be of practical use when performing a copy or remap task without changing data.

Report Options:
Miscellaneous options for report output.

Limit>
The maximum number of hits (records hit) to report per file, or library member.

Specify a value of **zero (0)** to report all hits.

Efficiency gains may be achieved by specifying **Limit=1** where it is required only to establish whether or not each member contains a particular string, since once the first hit is reported further processing for that member will be bypassed.

Context>
Display a specified number of input records immediately before and after each hit in order to provide context.

Suppress display of the output report following execution.
This option suppresses display of the report during and after execution and should not be used if report output is to a temporary, in-storage file.

Primary Commands

The following primary commands are supported by selected views in the Extended File Search sequence of panel views. If issued from a panel view in which the command is not valid, the message "ZZSP102E Primary command not valid in the current context" is displayed.

BROWSEINPUTCOPYBOOK

```
>>--+- BROWSEINPUTCopybook ------+-------------------------------------><
    +- BIC ----------------------+
```

If the search is to be performed on formatted record data, then BROWSEINPUTCOPYBOOK will attempt to browse the input structure file specified by the Input Structure/Copybook File fields of the FSU (formatted): Record Selection panel view.

Once displayed, GO EDIT may be used to convert the BROWSE view to a text edit view and so allow updates to the structure source. If the source is updated, the input structure Recompile option should be set to ensure that a new copy of the structure is loaded/generated when actioning the search utility.

BROWSEINPUTCOPYBOOK is assigned to <F22> by default.

BROWSEINPUTFILE

```
>>--+- BROWSEInputfile ----------+-------------------------------------><
    +- BIF ----------------------+
    +- VWINP --------------------+
```

Browse the fileid represented by the DSN/Path, Member and Volume mask fields. The file is considered to be an HFS file path if the combination of these fields describes a fileid mask containing wild card characters.

If the search is to be performed on formatted record data, then BROWSEINPUTFILE will attempt to browse the input file using the input structure specified by the Input Structure/Copybook File fields of the FSU (formatted): Record Selection panel view.

BROWSEINPUTFILE is assigned to <F19> by default.

BROWSEOUTPUTCOPYBOOK

```
>>--+- BROWSEOUTPUTCopybook -----+-------------------------------------><
    +- BOC ----------------------+
```

If the utility is to remap fields in formatted record data, then BROWSEOUTPUTCOPYBOOK will attempt to browse the output structure file specified by the Output Structure/Copybook File fields of the FSU: Remap Fields to an Output File or Output Library panel view.

Once displayed, GO EDIT may be used to convert the BROWSE view to a text edit view and so allow updates to the structure source. If the source is updated, the output structure Recompile option should be set to ensure that a new copy of the structure is loaded/generated when actioning the file remap utility.

BROWSEOUTPUTCOPYBOOK is assigned to <F23> by default.

BROWSEOUTPUTFILE

```
>>--+- BROWSEOutputfile ---------+-------------------------------------><
    +- BOF ----------------------+
    +- VWOUT --------------------+
```

Browse the Output file represented by fields that constitute the PDS/PDSE Library, PDS/PDSE Member, Sequential or VSAM data set or HFS path in the Copy Output File, Remap Output File, Copy Output Library or Remap Output Library Members panel.

If the utility is to remap fields in formatted record data, then BROWSEOUTPUTFILE will attempt to browse the output file using the output structure specified by the Output Structure/Copybook overlay fields.

BROWSEOUTPUTFILE is assigned to <F20> by default.

CMX

```
>>--+- CMX ----------------------+-------------------------------------><
    +- EDITCMX ------------------+
```

Generate command syntax. Same as menu bar item, **Command** available from the FSU: Report File view.

CMX is assigned to <F17> by default.

FILTER

```
>>---- FILTer ------------------------------------------------------><
```

Depending on the value entered in the Filter Type field ("Q" or "F") of the unformatted or formatted Record Selection panel views, FILTER attempts to display either the Quick Filter Selection Criteria panel or the Create File Filter panel for the file represented by the Record Filter fields.

FILTER is assigned to <F6> by default.

MAP

```
>>--+- MAP -------------------------------------------------------><
```

Interactively match-up Record-types and Field that are not identically named for use in a file reformat operation.

Record-types and Field that are identically named are matched automatically.

JCL

```
>>--+- JCL ----------------------+-----------------------------------><
    +- EDITJCL ------------------+
```

Generate a batch JCL and command syntax. Same as menu bar item, **JCL** available from the FSU: Report File view.

JCL is assigned to <F18> by default.

RUN

```
>>--+- RUN ----------------------+-----------------------------------><
    +- EXECSYNtax ---------------+
```

Verify input fields in all applicable panel views and execute the FSU utility to search, update, copy or remap file data as directed. This is the default action on pressing <Enter> from the last panel view in the sequence (i.e. the FSU: Report File panel view.)

SELECT

```
>>---- SELect ------------------------------------------------------><
```

SELECT will open a sub-panel that is applicable to the current panel view. Each of the panel views listed below identify the sub-panel opened on executing SELECT. In all other panel views, SELECT is invalid.

- **FSU: Input File(s) Specification**
  If Specify Multiple fileids or fileid masks has been selected, SELECT will open the Multiple Fileid Masks sub-panel.

  If Specify Single fileid mask has been selected, SELECT will open the Select Files to Process sub-panel
- **FSU (unformatted): Search records using WHERE expression**
  Opens the Unformatted Selection Criteria sub-panel to generate a WHERE expression based on unformatted text.
- **FSU (formatted): Search records using WHERE expression**
  Opens the Formatted Record Expression sub-panel to generate a WHERE expression that involves formatted record field references.
- **FSU (unformatted): Search Records using the FIND Command**
  Opens the Multiple Find Commands (unformatted) sub-panel to generate one or more FIND commands based on unformatted text.
- **FSU (formatted): Search Records using the FIND Command**
  Opens the Multiple Find Commands (formatted) sub-panel to generate one or more FIND commands that may include formatted record field references.
- **FSU (unformatted): Change record data using the CHANGE command**
  Opens the Multiple Change Commands (unformatted) sub-panel to generate one or more CHANGE commands based on unformatted text.
- **FSU (formatted): Change record data using the CHANGE command**
  Opens the Multiple Change Commands (formatted) sub-panel to generate one or more CHANGE commands that may include formatted record field references.

SELECT is assigned to <F5> by default.

File Selection Panels

Multiple Fileid Masks

The **Multiple Fileid Masks** panel view (ZZSGFSU3) is displayed when an option is taken to specify multiple fileids or fileid masks for the File Search/Update/Copy/Remap (FSU) utility.

This panel is an interactive panel window (window class WINWIPO0) and includes an embedded table.

Standard table edit primary and line commands must be used to insert, delete or exclude table rows as appropriate. The table entries each identify a fileid mask which may identify one or more sequential or VSAM data sets, HFS file paths and/or PDS/PDSE libraries.

For each table entry, the Select Files to Process panel may be displayed containing a list of selectable fileids that match the selected fileid mask. Furthermore, for each PDS/PDSE library entry in the **Select Files to Process** list, the Select Input Members panel may then be opened to select from a list of matching member names or member generations belonging to that library.

Having displayed the **Select Files to Process** panel for a fileid mask, a list of the selected, individual fileids will be passed to the utilty instead of the generic fileid mask.

On completing entries in this panel view, execute BACK (assigned to <F3> by default) to close the panel and return to the utility panel view.

Table View

The panel initially displays the embedded table with column names **Sel**, **DSN/HFS Mask**, **Member Mask** and **Volume Mask**.

Place the cursor on the required table entry and press the <Enter> key or, if configured, double-click the left mouse button to display the Select Files to Process panel.

If the input column area is not large enough to accomodate the required entry value, display the table row in zoomed format and expand the field entries as required. To display a table row in single row (zoomed) format, execute primary command ZOOM (assigned to <F17> by default) with the cursor positioned on the required table row entry.

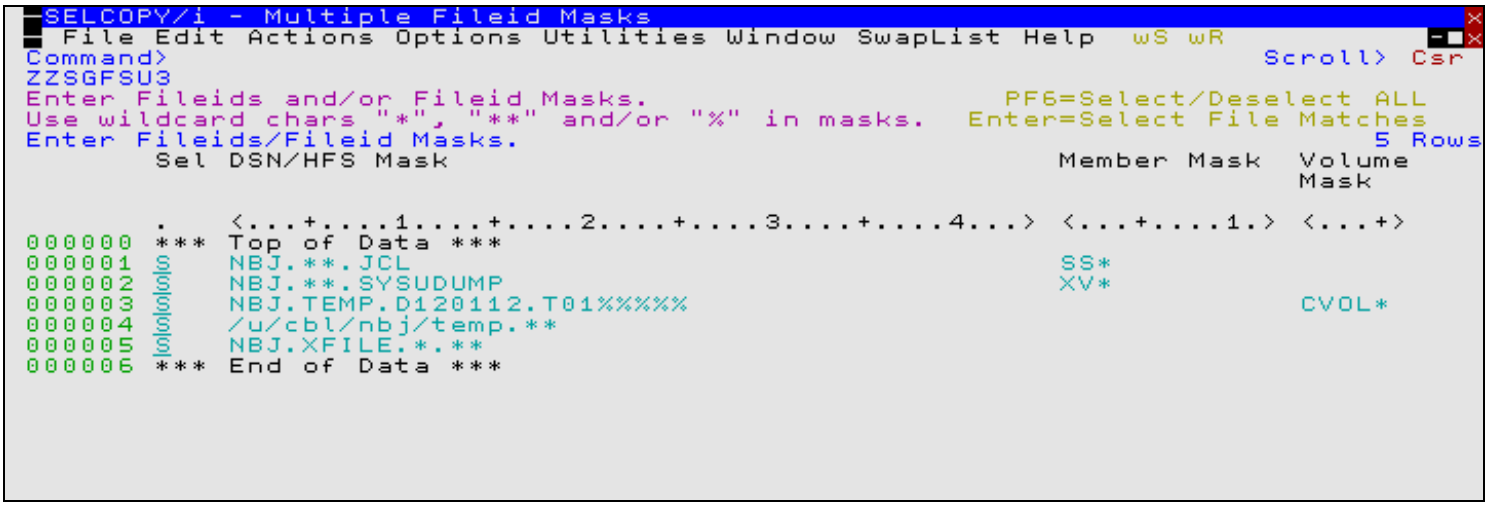


Figure 58. Multiple Fileid Masks.

Single Row (Zoomed) View

This panel displays the table column entries for an individual table row as panel input fields. Panel primary command EXPAND (assigned to <F14> by default) may be used to expand an input field and so allow entry of a value which exceeds the visible width of the input field area.

Execute primary command SELECT (assigned to the <F5> by default) to display the Select Files to Process panel from this panel. Execute panel primary command BACK (assigned to <F3> by default) to close the zoomed panel view.

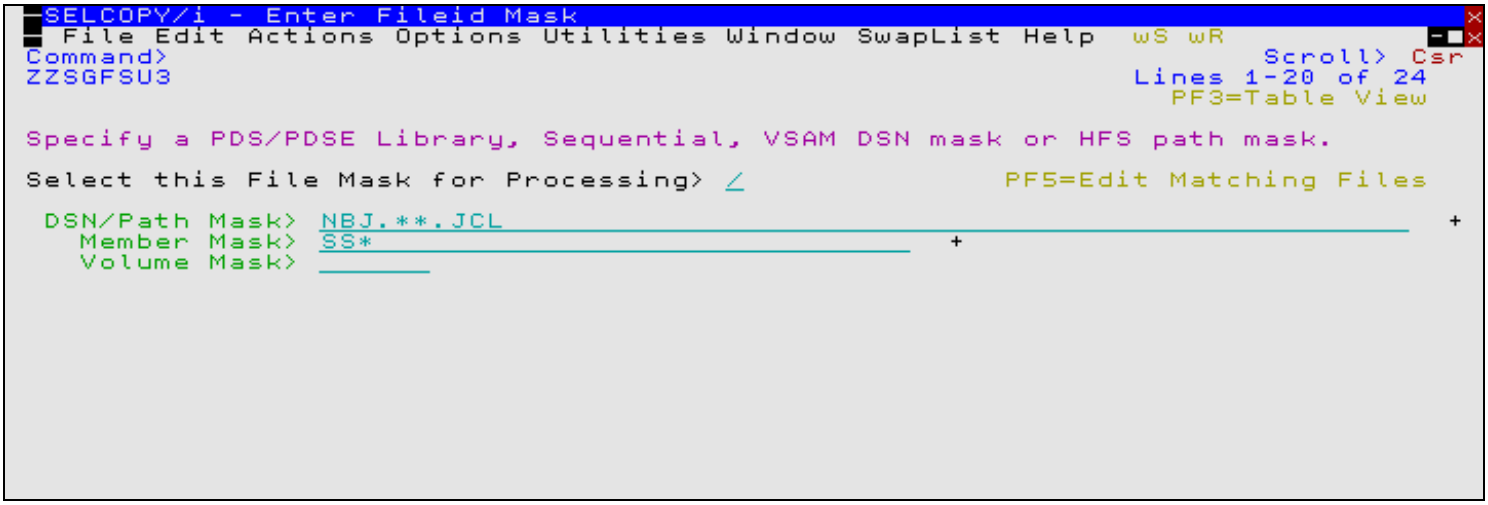


Figure 59. Enter Fileid Mask (Zoomed View).

Input Values

Fields which together constitiute a fileid mask that identifies one or more files to be processed by the utility.

`Sel>`
Corresponds to zoomed panel option "Select this File Mask for Processing>". Enter "S" (or any non-blank value) in this field to select the entry for processing. If left blank, processing for the fileid mask identified by this table entry is bypassed.

`DSN/HFS Path Mask>`
An unquoted entry which identifies DSN mask, a completed Fileid mask, a DDName mask or an HFS file path mask. An entry in this input field is mandatory.

- ◊ **HFS file path mask**
  Identified by the presence of "." (dot/period) as the first character or "/" (slash) anywhere within the mask, an HFS file path mask may be absolute or relative to the current HFS working directory. See USS PWD in Unix System Services (USS) Commands. "./" should prefix the mask if files in the current HFS directory are to be selected.

  Wild card characters "%" (percent), representing a single characters, and "*" (asterisk), representing zero or more characters, are supported in the name portion of the HFS file path mask. The name portion of the HFS path is the character string at the end of the path that follows the last "/" (slash) of the fileid or is the entire path name if "/" is not specified.

  If an HFS file path mask is specified, the contents of the **Volume Mask>** and **Member Mask>** fields will be ignored and **HFS Options>** settings respected.

- ◊ **DDName mask**
  If the value entered is not an HFS file path mask, is of length less than or equal to 8 characters and does not include "." (dot/period), and no Volume mask has been specified, then the value is considered to be a DDName mask. A DDName mask may be used to process non-HFS files that are currently allocated to DDNames that match the DDName mask. (e.g. SYSEXEC)

  Wild card characters "%" (percent), representing a single characters, and "*" (asterisk), representing zero or more characters, are supported in a DDName mask.

  If a DDName mask is specified, the enties in the **Member Mask>** field may still be used to select matching member names or member generations belonging to PDS/PDSE library data sets allocated to matching DDNames. **HFS Options>** settings are ignored.

- ◊ **Completed Fileid mask**
  If the value entered is not an HFS file path mask but includes a volume mask specification and/or a member mask specification, then the value is considered to be a completed Fileid mask.

  A completed fileid mask is a DSN mask with a volume mask and/or one or more PDS/PDSE member name or generation masks expressed in the following format:

  ```
  {volmask:}data.set.name.mask{( membmask{ {,} membmask...} )}
  ```

  Specification of one or more member masks between a single pair of "( )" (parentheses) will restrict processing to only PDS/PDSE library data sets. Multiple PDS/PDSE member masks must be separated by a "," (commma) and/or one or more intervening blanks.

  Specification of a 1 to 6 character volume mask prefix followed by ":" (colon) restricts processing to only cataloged or uncataloged data sets that have extents on matching volume ids.

  The fileid mask supports wild card characters as described for Volume Mask> and Member Mask> fields and for a DSN Mask below.

Fileid Mask Examples:

```
PE1.DEV.SRC.COBOL.CRKSW00(*)
SYS6.JNP*.**
OEM.TEST%%.**.CBLI.**(BOX*,D%T*,*ALL)
Z9RES1:ADCD.**
Z9RES*:ADCD.Z19.P%%%LIB(*)
```

◊ **DSN mask**
If the value entered does not meet the criteria described above for an HFS file path mask, completed Fileid mask or a DDName mask, then the value is considered to be a DSN mask.

The DSN mask is joined with any member and volume mask specified in the Volume Mask> and Member Mask> fields respectively to identify a single fileid mask.

The following wild card characters are supported within a DSN Mask:

*	A single asterisk represents a DSN qualifier or zero or more characters within a DSN qualifier. e.g. `DEV.CBLINS.*.JCL`, `DEV.CBLINS.TEST*.ISP*LIB`, `DEV.CBLINS.*.*`
**	Double asterisk represents zero or more qualifiers within a DSN. Double asterisk may only be preceded or followed by the qualifier separator, "." (dot/period). e.g. `DEV.CBLINS.**`, `DEV.CBLINS.**.CBLE`
%	A single percent sign represents exactly one character within a DSN qualifier. e.g. `DEV.CBLINS.TEST0%.JCL`, `DEV.CBLI%%.TEST06.CBLI.%%%`

Note that a TSO prefix is **not** applied to a DSN mask.

Volume Mask>
Optionally specify a volume name mask of maximum length 6-bytes.

Processing will be restricted to only those cataloged or uncataloged data sets that match the DSN mask **and** also have extents that exist on a volume that matches the volume mask.

The volume mask supports wild card characters as follow:

*	A single asterisk represents a complete volume name or zero or more characters within a volume name. e.g. `CBL*`, `*RES*`
%	A single percent sign represents exactly one character within the volume mask. e.g. `Z9DB9%`, `%%XV3%`

This field entry is ignored if the DSN/Path Mask> field does not contain a DSN mask.

Member Mask>
Optionally specify one or more PDS/PDSE member name or member generation masks separated by a "," (comma) and/or one or more intervening blanks.
e.g. `BLOCK, PROFILE  BOXSEQ`

If a member mask is specified, then only PDS/PDSE libraries that match the fileid mask will be selected for processing. Non-PDS/PDSE library data sets will be excluded.

If a member mask is **not** specified, then all files that match the fileid mask will be selected for processing and a default member name mask of "*" will apply to all PDS/PDSE libraries included in this selection.

Processing will be restricted to only those PDS/PDSE data sets that match the DSN mask **and** only members with a member name that matches any one of the supplied member masks.

A member mask supports wild card characters as follow:

*	A single asterisk represents an entire member name or zero or more characters within a member name. e.g. `CBL*5`, `BOX*`, `D*T*`
%	A single percent sign represents exactly one character within a member name mask. e.g. `H%`, `D%R*`, `E%A`

This field entry is ignored if the DSN/Path Mask> field does not contain a DSN mask or a DDNAme mask.

Select Files to Process

The **Select Files to Process** panel view (ZZSGFSU1) is displayed when a user elects to display the individual sequential, VSAM or PDS/PDSE library data sets or HFS files selected by a specific fileid mask.

This panel is an interactive panel window (window class WINWIPO0) and includes an embedded table.

Standard table edit primary and line commands may be used to delete or exclude table rows as appropriate. Alternatively, select/deselect entries for processing by inserting/removing "S" from the **Sel** column. The table entries each identify the fileid of a sequential or VSAM data sets, HFS file paths or PDS/PDSE library.

For each PDS/PDSE library table entry, the Select Input Members panel may be opened to select from a list of members that match the member name mask entered in the Member Mask column.

Having displayed the **Select Input Members** panel for a library member mask, a list of the selected, individual member names will be passed to the utilty instead of the generic library member mask.

On completing entries in this panel view, execute BACK (assigned to <F3> by default) to close the panel and return to the utility panel view.

Table View

The panel initially displays the embedded table with column names **Sel**, **FileName**, **Member Mask**, **Volume**, **Org**, **VSAM**, **GDG**, **RecFm**, **Lrecl**, **BlkSz** and **Select Members**. Execute LEFT and RIGHT (assigned, by default, to <F10> and <F11> respectively) to display columns that are not in view.

Only columns **Sel** and **Member Mask** are enterable. All other column entries are included for information only.

For library entries (identified as having and entry in the **Member Mask** column) place the cursor on the required table entry and press the <Enter> key or, if configured, double-click the left mouse button to display the Select Input Members panel for the specified member or member generation mask(s).

If the input member mask column area is not large enough to accomodate the required entry value, display the table row in zoomed format and expand the field entries as required. To display a table row in single row (zoomed) format, execute primary command ZOOM (assigned to <F17> by default) with the cursor positioned on the required table row entry.

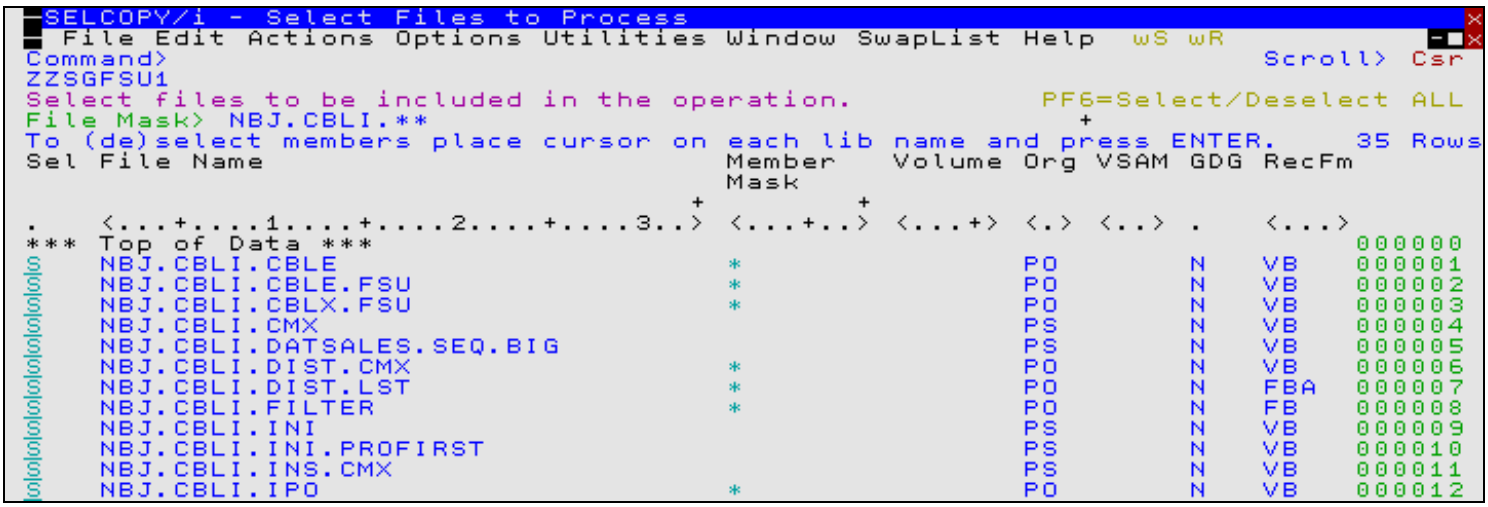


Figure 60. Select Files to Process.

Single Row (Zoomed) View

This panel view displays the table column entries for an individual table row as panel input fields. Panel primary command EXPAND (assigned to <F14> by default) may be used to expand an input field and so allow entry of a value which exceeds the visible width of the input field area.

Execute primary command SELECT (assigned to the <F5> by default) to display the Select Input Members panel from this panel. Execute panel primary command BACK (assigned to <F3> by default) to close the zoomed panel view.

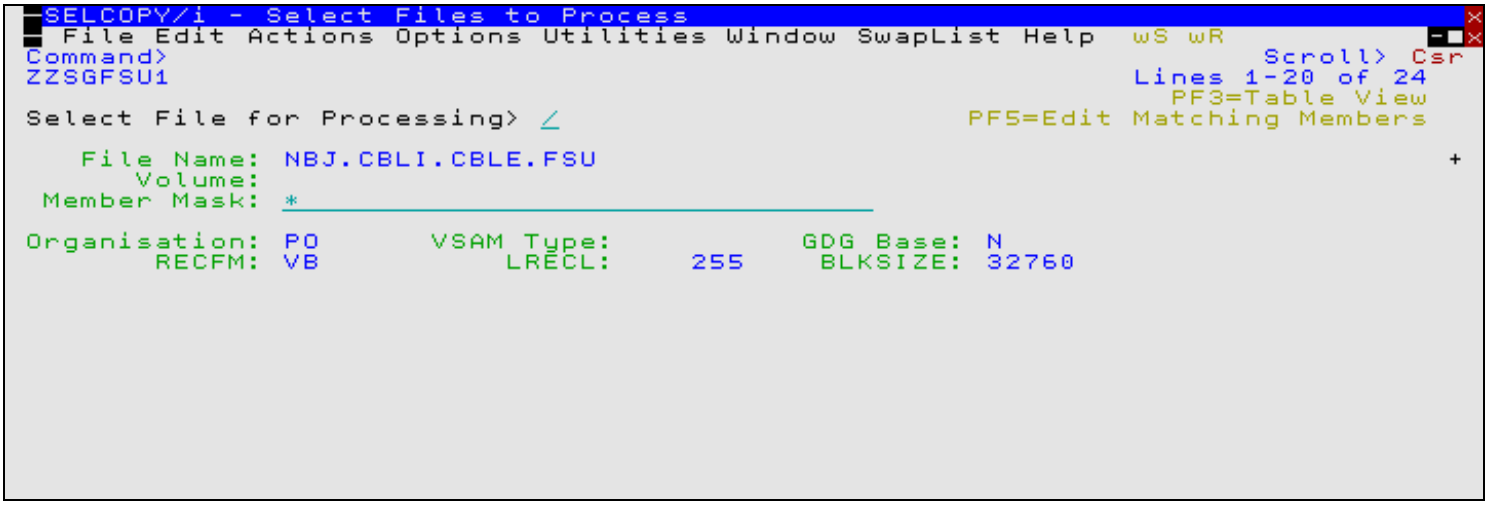


Figure 61. Select Files to Process (Zoomed View).

Input Values

Fields which identify the file to be processed by the utility.

`Sel>`
Corresponds to zoomed panel option "Select File for Processing>". Enter "S" (or any non-blank value) in this field to select the entry for processing. If left blank, processing for the fileid identified by this table entry is bypassed.

`File Name:`
A non-enterable field identifying the DSN of a sequential, VSAM or PDS/PDSE library data set or an HFS file path selected by the supplied fileid mask.

`Volume:`
Contains a value only if the specified fileid mask includes a volume mask.

This is a non-enterable field identifying the volume id of the primary volume on which the file extents exist.

`Member Mask:`
An enterable field in which one or more PDS/PDSE member name masks, separated by a "," (comma) and/or one or more intervening blanks, may be entered. By default, this field contains the member mask supplied in the fileid mask, otherwise "*" (asterisk).

See Member Mask in the description of the Multiple Fileid Masks panel for details.

`Organisation:`
A non-enterable field which identifies the data set organisation of the file. e.g. PS (Physical Sequential), PO (Partitioned Organised), VS (VSAM) or HFS.

`VSAM Type:`
Applicable to VSAM data sets only, a non-enterable field which identifies the type of VSAM data set organisation.

`GDG Base:`
A non-enterable field which identifies whether or not the file is a GDG base. Possible values are Y (Yes) or N (No).

`RECFM:`
Applicable to sequential and PDS/PDSE library data sets only, a non-enterable field which identifies the defined record format of the file.

`LRECL:`
Applicable to non-HFS files only, a non-enterable field which identifies the defined maximum record length of the file.

`BLKSIZE:`
Applicable to sequential and PDS/PDSE library data sets only, a non-enterable field which identifies the file's defined block size.

`Select Members`
Displayed as a column in table view only.

Applicable to PDS/PDSE library data sets only, a non-enterable field which identifies the number of member selected after the **Select Innput Members** panel has been displayed.

Select Input Members

The **Select Input Members** panel view (ZZSGFSU2) is displayed when a user elects to display the individual library members selected by a specific member mask or member generations selected by a specific member generations mask.
This panel is an interactive panel window (window class WINWIPO0) and includes an embedded table.

Standard table edit primary and line commands may be used to delete or exclude table rows as appropriate. Alternatively, select/deselect entries for processing by inserting/removing "S" from the **Sel** column. The table entries each identify the name of a PDS/PDSE library member.

On completing entries in this panel view, execute BACK (assigned to <F3> by default) to close the panel and return to the parent panel view.

Table View

The panel initially displays the embedded table with column names **Sel**, **Member**, **GenA** and **GenR** (member generations only), **Alias**, **Created**, **LastMod**, **Cursize**, **Inisize**, **TTR**, **User** and **AliasOf**. Execute LEFT and RIGHT (assigned, by default, to <F10> and <F11> respectively) to display columns that are not in view.

Only column **Sel** is enterable. All other column entries are included for information only.

The primary command FIND may be executed in this panel view to condense the table so that only rows of members containing records that match a specified find string criterion. FIND will open the Select Input Members - FIND sub-panel.

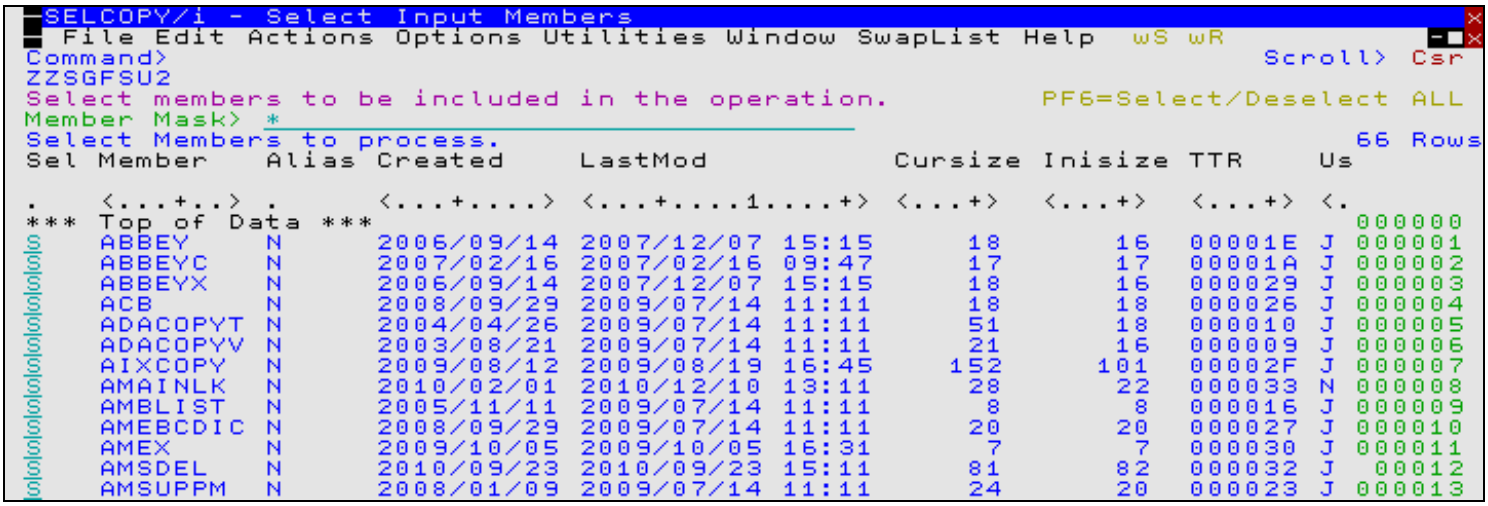


Figure 62. Select Input Members.

Single Row (Zoomed) View

This panel view displays the table column entries for an individual table row as panel input fields.

Execute panel primary command BACK (assigned to <F3> by default) to close the zoomed panel view and return to the table panel view.

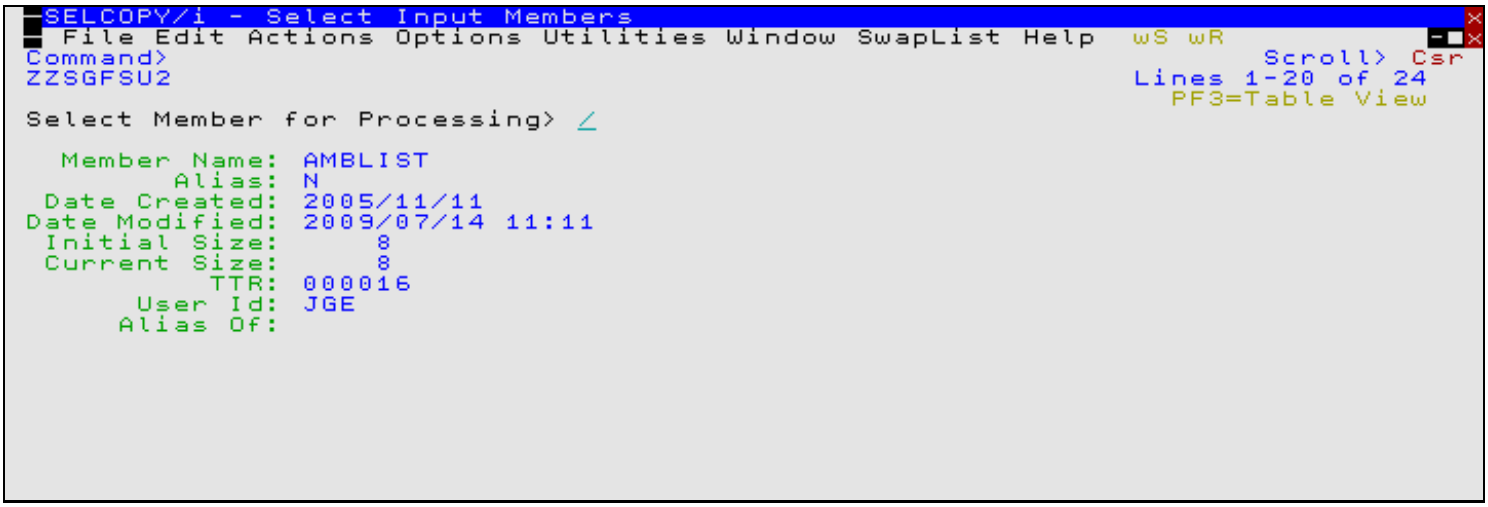


Figure 63. Select Input Members (Zoomed View).

Input Values

Fields which identify the member to be processed by the utility.

Member Mask>
An enterable field in which one or more PDS/PDSE member name masks, separated by a "," (comma) and/or one or more intervening blanks, may be entered. By default, this field contains the member mask supplied in the Member Mask field of the **Select Files to Process** panel.

See Member Mask in the description of the Multiple Fileid Masks panel for details.

Sel>
Corresponds to zoomed panel option "Select Member for Processing>". Enter "S" (or any non-blank value) in this field to select the entry for processing. If left blank, processing for the member identified by this table entry is bypassed.

Member Name:
A non-enterable field identifying the name of the library member selected by the supplied member mask.

Absolute Gen:
A non-enterable field identifying the absolute generation number of the library member generation selected by the supplied member mask.

Relative Gen:
A non-enterable field identifying the relative generation number of the library member generation selected by the supplied member mask.

Alias:
A non-enterable field indicating whether the member is an alias of another library member. Possible values are Y (Yes) or N (No).

Date Created:
For non-load libraries, a non-enterable field identifying the date on which the member was created.

Date Modified:
For non-load libraries, a non-enterable field identifying the date an time at which the member was last modified.

Initial Size:
For non-load libraries, a non-enterable field identifying the number of records in the member when it was first saved.

Current Size:
For non-load libraries, a non-enterable field identifying the number of records in the member when it was last saved.

TTR:
A non-enterable field identifying the TTR (track offset/record number) of the member data. The TTR value is displayed as a 3-byte hexadecimal value where byte 1 and 2 represent the track offset, byte 3 the record number.

User Id:
For non-load libraries, a non-enterable field identifying the userid of the user who created the member.

Alias Of:
For load libraries, a non-enterable field identifying the member name for which the member is an ALIAS.

Select Input Members - FIND

The **Select Input Members - FIND** panel view (ZZSGFSU9) is displayed when primary command FIND is executed in an FSU Select Input Members list and may be used to condense the list of displayed members. Only those members that contain at least one record which satisfies the find operation are included.

Following execution of the find operation, the FIND panel remains open to allow further find/condense operations on the list of members.

This panel is an interactive panel window (window class WINWIPO0).

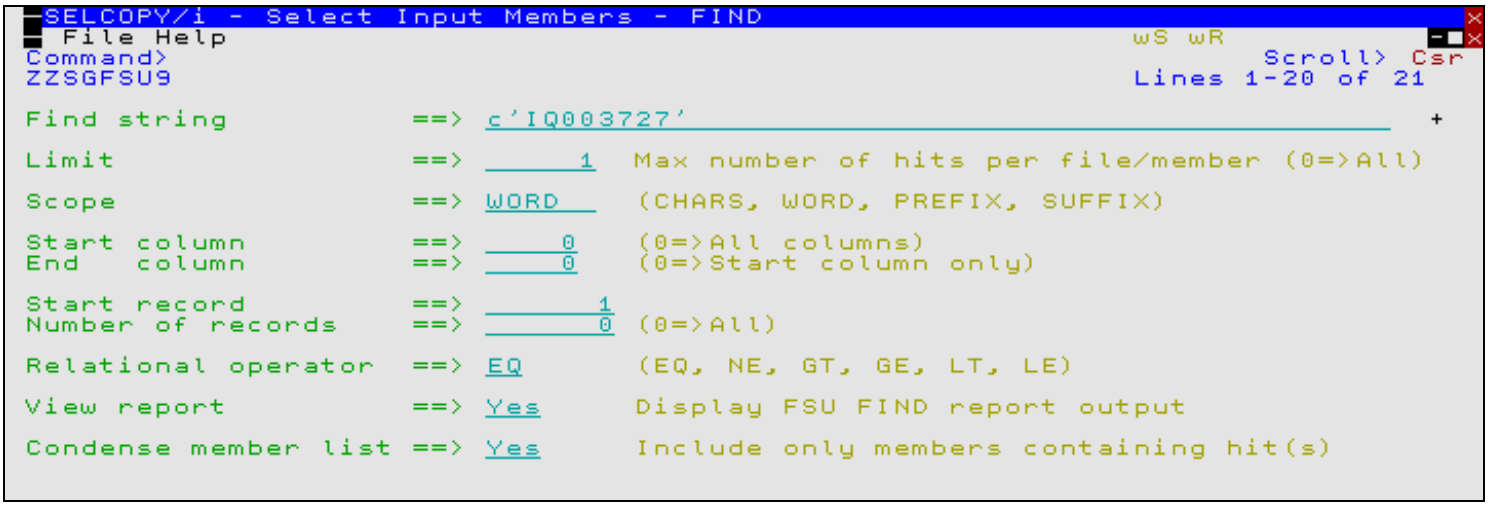


Figure 64. Find/Condense Input Members.

Execute panel primary command BACK (assigned to <F3> by default) to close the panel and return to the parent member list panel view. To reset the list of members following a condense operation, simply re-enter a value in the Member Mask field.

Input Values

Find string ==>

Specify the search value. This may be one of the following:

1. An unquoted numeric value.
2. A quoted character string (case insensitive search). e.g. 'John Doe' is the same as 'JOHN DOE'.
3. A quoted character string prefixed with "C" (case sensitive search). e.g. C'John Doe'.
4. A quoted hexadecimal string prefixed with "X". e.g. X'00'.
5. A quoted picture string prefixed with "P". e.g. P'USER###'.
   See the FIND command for details on supported picture string special characters.

Limit ==>

The maximum number of hits (records hit) to report per library member.

Specify a value of **zero (0)** to report all hits.

Efficiency gains may be achieved by specifying **Limit=1** where it is required only to establish whether or not each member contains a particular string, since once the first hit is reported further processing for that member will be bypassed.

Scope ==>

Enter restrictions to be applied to the location of the search string within record data as follow:

CHARS	The search value may be found anywhere within the specified column bounds of the input records. i.e. No restriction.
WORD	The search value may only be found if it is complete word which falls within the specified column bounds of the input records. i.e. the value must either be preceded by a non-alphanumeric character or begin at the start column bound, and either precede a non-alphanumeric character or finish in the end column bound.
PREFIX	The search value may only be found if it occurs at the start of a word and falls within the specified column bounds of the input records. i.e. the value must either be preceded by a non-alphanumeric character or begin at the start column bound, and be followed by an alphanumeric character.
SUFFIX	The search value may only be found if it occurs at the end of a word and falls within the specified column bounds of the input records. i.e. the value must be preceded by an alphanumeric character and must either precede a non-alphanumeric character or finish in the end column bound.

These options correspond to SDE FIND parameters PREFIX, SUFFIX and WORD.

Start column ==>
Specifies the start (or only) record column from which the scan for the search string will begin. Record data in positions or fields that occur before this start column value is not searched. A value of 0 (zero) implies all columns are included.

This field corresponds to SDE FIND parameter *pos1*.

End column ==>
Specifies the end record data column beyond which no part of the search value may be found. Only record data between the start and end columns will be searched.

If a 0 (zero) end column is specified then the end column is the start column number plus the length of the search value minus 1. i.e. the search string will only be found if it begins in the start column.

This field corresponds to SDE FIND parameter *pos2*.

Start record ==>
Specifies the first record number in every member at which processing will start. Records occurring sequentially before the start record will be bypassed.

This field corresponds to the FSU parameter STARTREC *recno*.

Number of records ==>
Specifies the maximum number of records within each member for which FIND processing may occur. If a 0 (zero) value is specified, then records are selected beginning at start record and ending at the last record in the file.

This field corresponds to the FSU parameter FOR.

Relational operator ==>
Identify the relational operator used when comparing the record data against the search value.

If testing for a non-equality relationship using a character string search value, then the EBCDIC values assigned to characters in the search string and record data determine the relationship. (e.g. GT, LT)

View report ==>
Select "Yes" or "No" to determine whether a File Search utility report will be displayed for the FIND operation.

Condense member list ==>
Select "Yes" or "No" to determine whether the FIND operation will condense the displayed member list.

Primary Commands

The following primary commands are supported by selected File Selection sub-panels. If issued from a panel in which the command is not valid, the message "ZZSP102E Primary command not valid in the current context" is displayed.

FIND

```
>>---- FIND ------------------------------------------------------------><
```

Applicable only in the table view of the Select Input Members sub-panel, FIND will open the Select Input Members FIND sub-panel.

RUN

```
>>--+- RUN ----------------------+-----------------------------------><
    +- EXECSYNtax ---------------+
```

Applicable only in the Select Input Members FIND sub-panel, RUN will verify input fields and execute the FIND utility. This is the default action on pressing <Enter>.

SELECT

```
>>---- SELect ----------------------------------------------------------><
```

SELECT will open a sub-panel that is applicable to the current panel view. Each of the panel views listed below identify the sub-panel opened on executing SELECT. In all other panel views, SELECT is invalid.

- **Multiple Fileid Masks - Single Row (Zoomed) View**
  Opens the Select Files to Process sub-panel for the fileid mask in the zoomed table row.
- **Select Files to Process - Single Row (Zoomed) View**
  Opens the Select Input Members sub-panel for the library DSN in the zoomed table row.

SELECT is assigned to <F5> by default.

Generate Formatted Record Expression Panels

Major OR Sub-Expressions

The **Major OR Sub-Expressions** panel view (ZZSGEXP0) is displayed when an option is taken to generate an SDE expression for formatted record fields belonging to a specified record-type definition.

This panel is an interactive panel window (window class WINWIPO0) and includes an embedded table.

Standard table edit primary and line commands must be used to insert, delete or exclude table rows as appropriate. The table entries each identify an SDE expression. In the generated expression, these entries will occur in the order specified and will be separated by the OR logical operator. e.g. For 3 expression table entries represented by A1, A2 and A3

```
(A1)  OR  (A2)  OR  (A3)
```

For each table entry, the AND Sub-Expression panel **must** be displayed to specify one or more ANDed sub-expressions.

On completing entries in this panel view, execute BACK (assigned to <F3> by default) to close the panel and return to the utility panel view.

Table View

The panel initially displays the embedded table with column names **OR**, **NOT** and **AND Sub-expressions**.

Place the cursor on the required table entry and press the <Enter> key or, if configured, double-click the left mouse button to display the AND Sub-Expressions panel.

To display a table row in single row (zoomed) format, execute primary command ZOOM (assigned to <F17> by default) with the cursor positioned on the required table row entry.

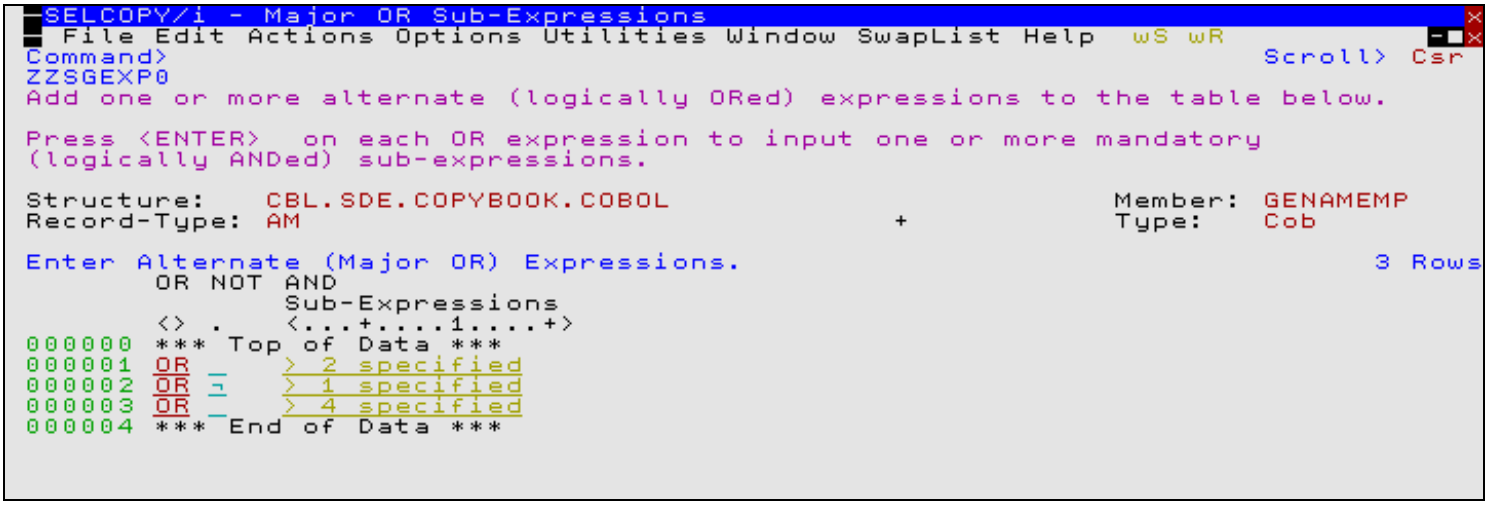


Figure 65. Major OR Sub-Expressions.

Single Row (Zoomed) View

This panel displays the table column entries for an individual table row as panel input fields.

Execute primary command SELECT (assigned to the <F5> by default) to display the AND Sub-Expressions panel from this panel view.
Execute panel primary command BACK (assigned to <F3> by default) to close the zoomed panel view and return to the table view.

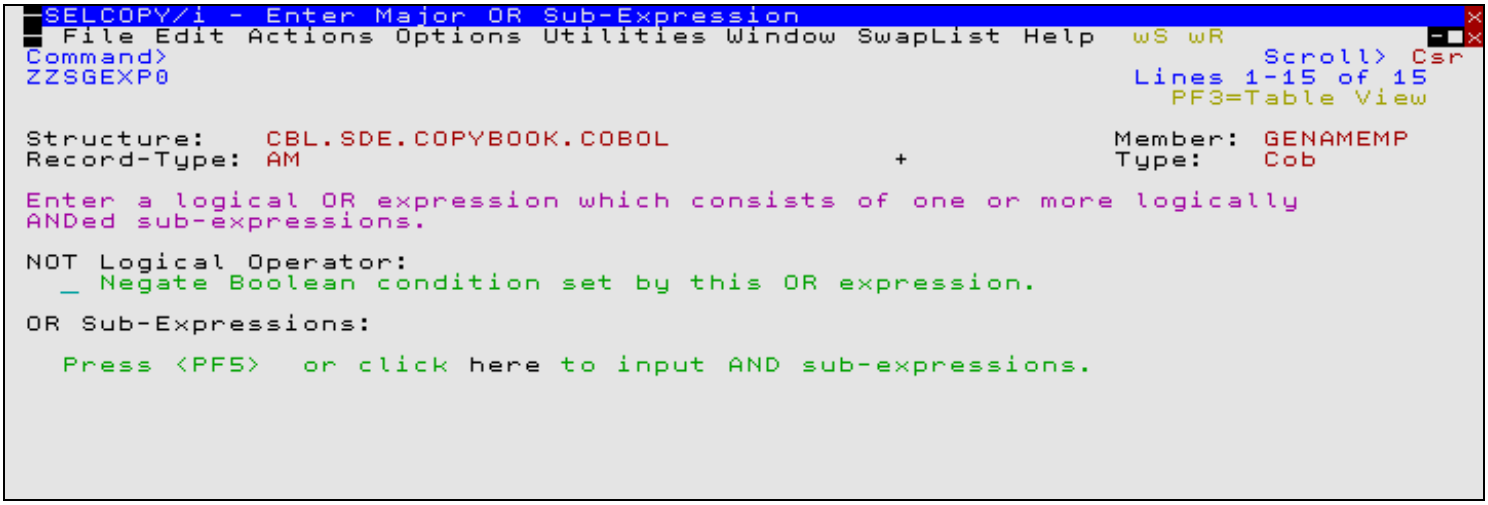


Figure 66. Enter Major OR Sub-Expression (Zoomed View).

Input Values

Fields which together identify a single SDE expression.

`Structure:`
A non-enterable field which identifies the DSN of the structure data set to be used for field name reference.

`Member:`
If Structure is a library member, a non-enterable field which identifies the member name of the structure to be used for field name reference.

`Type:`
A non-enterable field which identifies the structure source type: Cob (COBOL Copybook), Pl1 (PL1 Copybook), Ada (COBOL or PL1 ADATA file) or Sdo (SELCOPY/i Structure Definition Object).

`Record-Type:`
A non-enterable field which identifies the name of the record-type mapping, defined within the structure, to be used for field name reference.

`OR`
A non-enterable field which indicates that the expressions corresponding to each row of the table will be logically ORed together.

`NOT`
An enterable field which allows negation (logical NOT) of the result of the expression identified by the table entry. This option corresponds to the "Negate Boolean condition set by this OR expression" option input field in the zoomed view.

`AND Sub-Expressions`
Having displayed the AND Sub-Expressions panel, this field identifies the number of ANDed sub-expressions which constitute the table row expression.

AND Sub-Expressions

The **AND Sub-Expressions** panel view (ZZSGEXP1) is only displayed when an entry in the Major OR Sub-Expression panel has been selected for configuration.

This panel is an interactive panel window (window class WINWIPO0) and includes an embedded table.

Standard table edit primary and line commands must be used to insert, delete or exclude table rows as appropriate. The table entries each identify a single SDE sub-expression which together are seperated by a logical AND operator to constitute the single Major OR expression in the parent panel. e.g. For the ORed expressions in the Major OR example above:,

- A1 has two ANDed sub-expressions (represented by B1, B2).
- A2 has one sub-expression.
- A3 has four ANDed sub-expressions (represented by B3, B4, B5 and B6).

```
(B1 AND B2)  OR  (A2)  OR  (B3  AND  B4  AND  B5  AND  B6)
```

For each table entry, the Minor OR Sub-Expression panel **must** be displayed to specify one or more ORed sub-expressions.

On completing entries in this panel view, execute BACK (assigned to <F3> by default) to close the panel and return to the utility panel view.

Table View

The panel initially displays the embedded table with column names **AND**, **NOT** and **AND Sub-expressions**.

Place the cursor on the required table entry and press the <Enter> key or, if configured, double-click the left mouse button to display the Minor OR Sub-Expressions panel.

To display a table row in single row (zoomed) format, execute primary command ZOOM (assigned to <F17> by default) with the cursor positioned on the required table row entry.

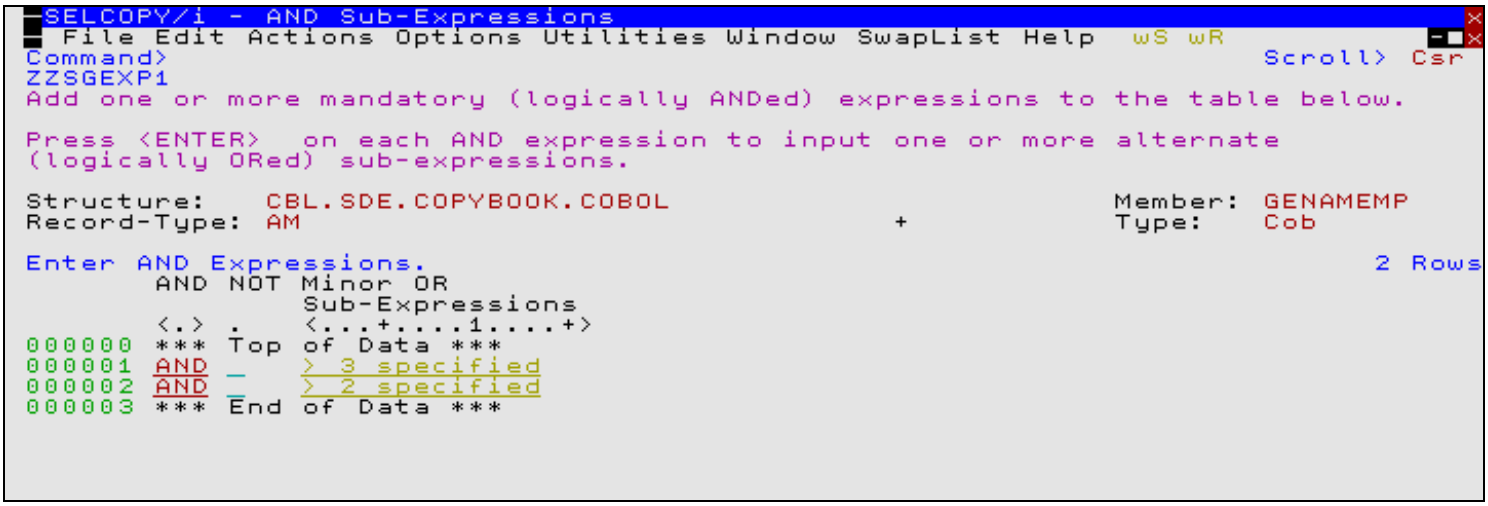


Figure 67. AND Sub-Expressions.

Single Row (Zoomed) View

This panel displays the table column entries for an individual table row as panel input fields.

Execute primary command SELECT (assigned to the <F5> by default) to display the Minor OR Sub-Expressions panel from this panel view.
Execute panel primary command BACK (assigned to <F3> by default) to close the zoomed panel view and return to the table view.

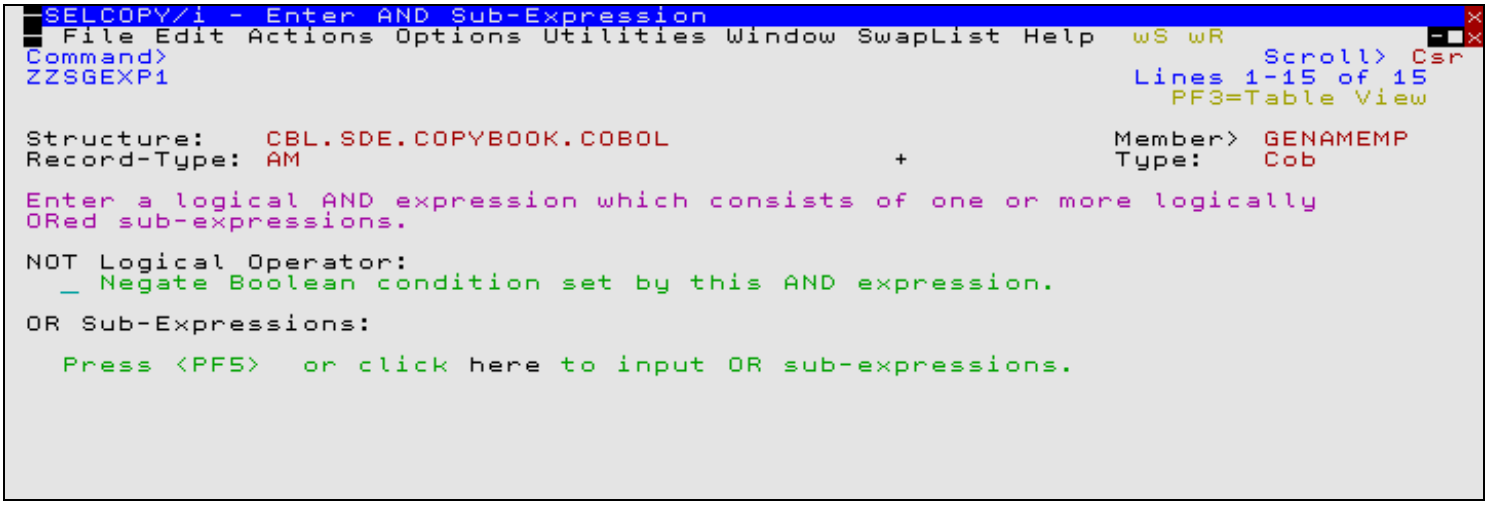


Figure 68. Enter AND Sub-Expression (Zoomed View).

Input Values

Fields which together identify a single SDE expression.

`Structure:`
A non-enterable field which identifies the DSN of the structure data set to be used for field name reference.

`Member:`
If Structure is a library member, a non-enterable field which identifies the member name of the structure to be used for field name reference.

`Type:`
A non-enterable field which identifies the structure source type: Cob (COBOL Copybook), Pl1 (PL1 Copybook), Ada (COBOL or PL1 ADATA file) or Sdo (SELCOPY/i Structure Definition Object).

`Record-Type:`
A non-enterable field which identifies the name of the record-type mapping, defined within the structure, to be used for field name reference.

`AND`
A non-enterable field which indicates that the expressions corresponding to each row of the table will be logically ANDed together.

`ANOT`
An enterable field which allows negation (logical NOT) of the result of the sub-expression identified by the table entry. This option corresponds to the "Negate Boolean condition set by this AND expression" option input field in the zoomed view.

`AND Sub-Expressions`
Having displayed the Minor OR Sub-Expressions panel, this field identifies the number of ORed sub-expressions which constitute the table row expression.

Minor OR Sub-Expressions

The **Minor OR Sub-Expressions** panel view (ZZSGEXP2) is only displayed when an entry in the AND Sub-Expression panel has been selected for configuration.

This panel is an interactive panel window (window class WINWIPO0) and includes an embedded table.

Standard table edit primary and line commands must be used to insert, delete or exclude table rows as appropriate. The table entries each identify a single SDE sub-expression which together are seperated by a logical OR operator to constitute the single AND expression in the parent panel. e.g. For the ANDed expressions in the AND example above:,

- B1 has three ORed sub-expressions (represented by C1, C2 and C3).
- B2 has two ORed sub-expressions (represented by C4 and C5).
- B3, B4, B5 and B6 each have a single sub-expressions.

```
((C1 OR C2 OR C3) AND (C4 OR C2))  OR  (A2)  OR  (B3 AND B4 AND B5 AND B6)
```

On completing entries in this panel view, execute BACK (assigned to <F3> by default) to close the panel and return to the utility panel view.

Table View

The panel initially displays the embedded table with column names **OR**, **Field Name**, **Ref#**, **Fmt**, **Pic**, **ROp** and **Value/Field**

Enter a value in the **Field Name** column to open the SDE: Select Field Name panel and select a single entry from a list of formatted record fields. Enter a test value in the **Value/Field** column which is valid for the data type of the source field.

The **Value/Field** may also contain a field reference to test the source field against the contents of this test field. To enter a field reference, first display the table row in singe row (zoomed) view to select the field option then enter a value for the Term 2 Field Name to select from the list of possible fields.

To display a table row in single row (zoomed) format, execute primary command ZOOM (assigned to <F17> by default) with the cursor positioned on the required table row entry.

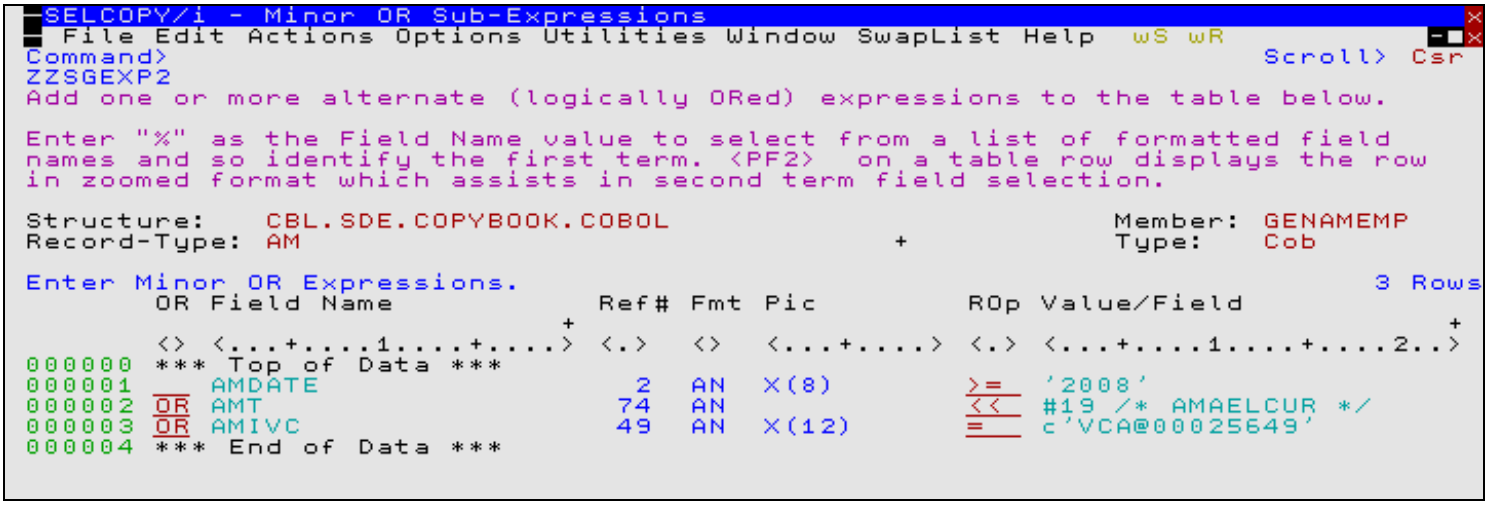


Figure 69. Minor OR Sub-Expressions.

Single Row (Zoomed) View

This panel displays the table column entries for an individual table row as panel input fields.

Execute panel primary command BACK (assigned to <F3> by default) to close the zoomed panel view and return to the table view.

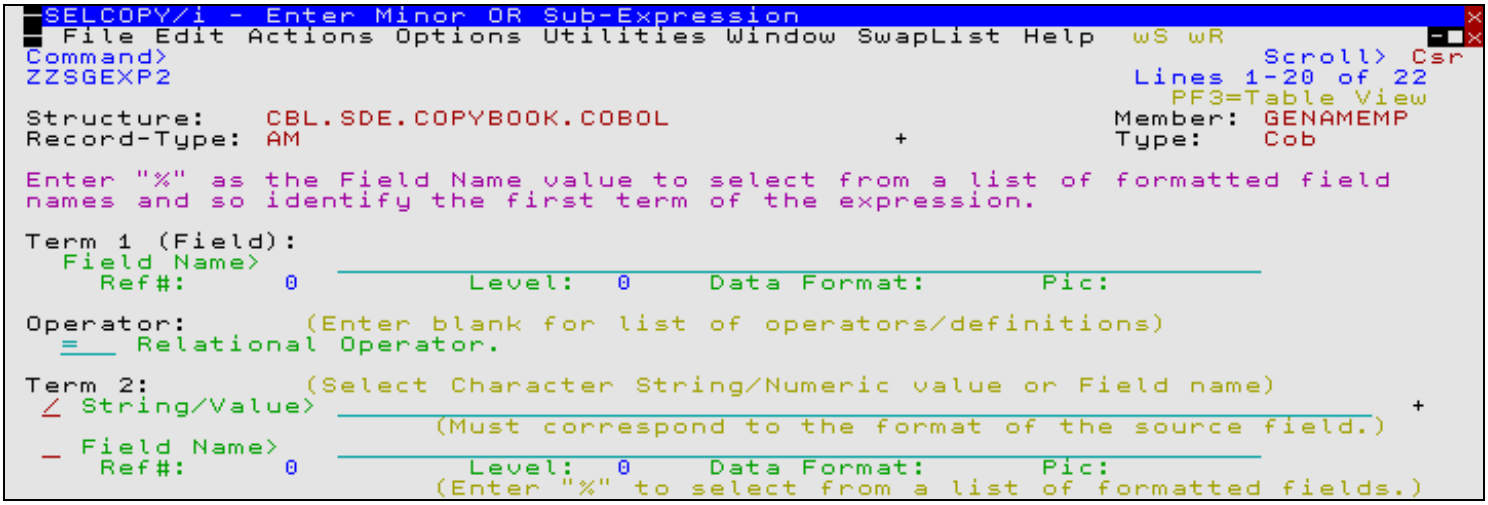


Figure 70. Enter Minor OR Sub-Expression (Zoomed View).

Input Values

Fields which together identify a single SDE expression.

`Structure:`
A non-enterable field which identifies the DSN of the structure data set to be used for field name reference.

`Member:`
If Structure is a library member, a non-enterable field which identifies the member name of the structure to be used for field name reference.

`Type:`
A non-enterable field which identifies the structure source type: Cob (COBOL Copybook), Pl1 (PL1 Copybook), Ada (COBOL or PL1 ADATA file) or Sdo (SELCOPY/i Structure Definition Object).

`Record-Type:`
A non-enterable field which identifies the name of the record-type mapping, defined within the structure, to be used for field name reference.

`OR`
A non-enterable field which indicates that the expressions corresponding to each row of the table will be logically ORed together.

`Term 1 (Field):`
Fields that identify the formatted record field that constitutes the left term (term 1) of the expression.

`Field Name / Field Name>`
An enterable field that identifies the name of a formatted record field belonging to the specified records-type.

This is a selectable value. Enter any value in this input field to select from a list of record-type fields.

`Ref# / Ref#:`
A non-enterable field that displays the field reference number of the selected field.

`Level:`
A non-enterable field that displays the group nesting level of the selected field.

`Fmt / Data Format:`
A non-enterable field that displays the data format of the selected field.

`Pic / Pic:`
Applicable only to COBOL or PL1 defined field that have an associated picture string, this is a non-enterable field that displays the picture string value of the selected field.

`ROp / Operator>`
The relational operator defining the compare operation to be performed on the two expression terms.

Enter any invalid entry to select from a list of possible operators.

`Value/Field / Term 2:`
Fields that identify the right term (term 2) of the expression. This may be a character string, numeric value or field reference which is a compatible data type with the term 1 field reference.

If a field reference is required, the table row must first be displayed in single row (zoomed) view.

In single row view, choose between 2 mutually exclusive options which identify term 2 as either a character string/numeric value or a field reference.

String Value>

Select this option to specify the term as a character string or numeric value that is compatible with the data type of the term 1 field. This may be one of the following:

1. Applicable to fields of any data type, an unquoted numeric value. For numeric fields, the numeric value and term 1 field data are converted so that they are of the same data type prior to performing an arithmetic comparison.
2. Applicable only to character fields, a quoted character string (case insensitive search). e.g. 'John Doe' is the same as 'JOHN DOE'.
3. Applicable only to character fields, a quoted character string prefixed with "C" (case sensitive search). e.g. C'John Doe'.
4. Applicable only to character fields, a quoted hexadecimal string prefixed with "X". e.g. X'00'.

Field Name>

Select this option to specify the term as a formatted record field reference which is of a data type compatible with the term 1 field.

The field name value is an enterable field that identifies the name of a formatted record field belonging to the specified records-type. This is a selectable value. Enter any value in this input field to select from a list of record-type fields.

The **Ref#**, **Level**, **Data Format** and **Pic** values, which correspond to the selected field, are displayed in non-eneterable fields as described for Term 1 above.

SDE: Select Record-Type

The **SDE: Select Record-Type** panel view (ZZSGSRT0) is displayed whenever a value in a selectable panel field is entered to select a record-type mapping from a specified structure.

This panel is an interactive panel window (window class WINWIPO0) and includes an embedded list window with all the features supported by this window class.

All fields displayed in each list entry are non-enterable and describe the field's characteristics. See the SDE DISPLAY RECTYPES command output for an overview of the displayed fields.

Enter "S" in the command prefix area of the required record-type definition to select it. Alternatively, place the cursor on the required list entry and press the <Enter> key or, if configured, double-click the left mouse button to select the required record-type definition.

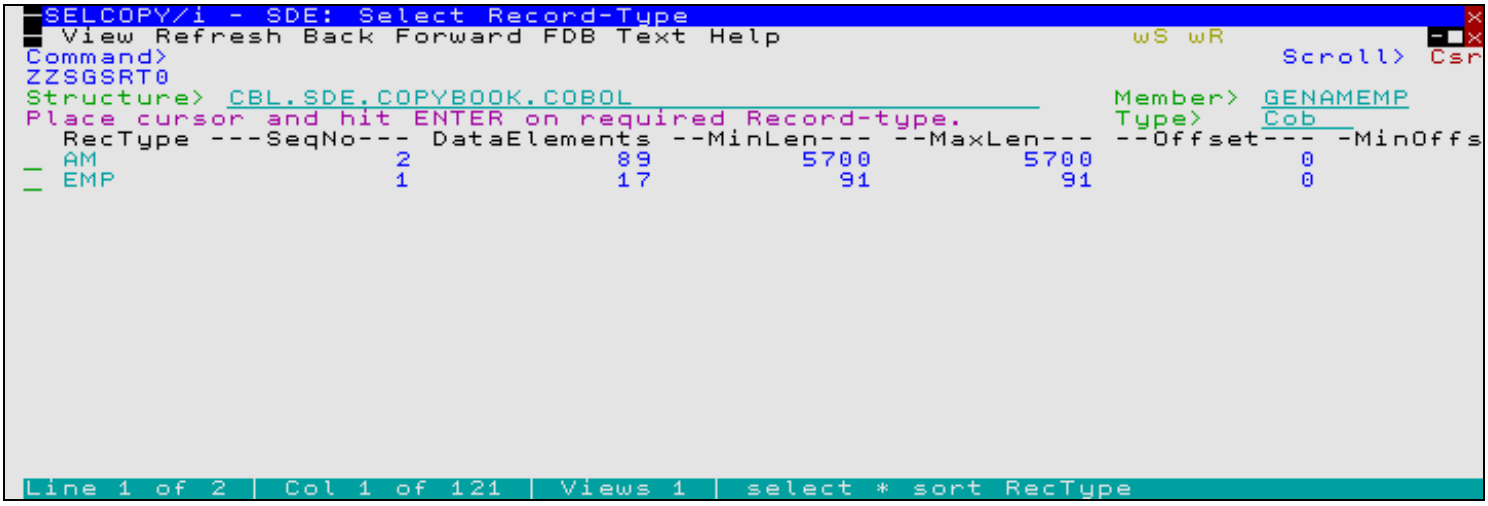


Figure 71. SDE: Select Record-Type.

SDE: Select Field Name

The **SDE: Select Field Name** panel view (ZZSGSFN0) is displayed whenever a value in a selectable panel field is entered to select a formatted record field from a structure record-type mapping.

This panel is an interactive panel window (window class WINWIPO0) and includes an embedded list window with all the features supported by this window class.

All fields displayed in each list entry are non-enterable and describe the field's characteristics. See the SDE DISPLAY STRUCTURE command output for an overview of the displayed fields.

Enter "S" in the command prefix area of the required field definition to select it. Alternatively, place the cursor on the required list entry and press the <Enter> key or, if configured, double-click the left mouse button to select the required field definition.

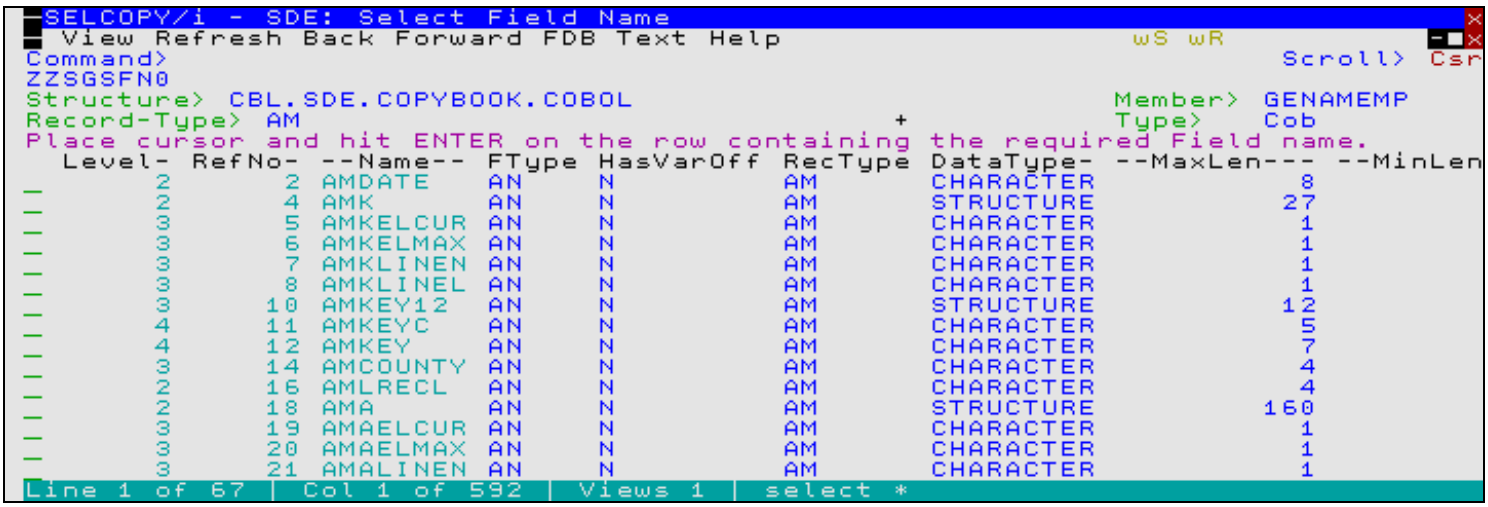


Figure 72. SDE: Select Field Name.

Primary Commands

The following primary commands are supported by selected Generate Formatted Record Expression sub-panels. If issued from a panel in which the command is not valid, the message "ZZSP102E Primary command not valid in the current context" is displayed.

SELECT

```
>>---- SELect ------------------------------------------------------><
```

SELECT will open a sub-panel that is applicable to the current panel view. Each of the panel views listed below identify the sub-panel opened on executing SELECT. In all other panel views, SELECT is un invalid.

- **Major OR Sub-Expressions - Single Row (Zoomed) View**
  Opens the AND Sub-Expressions sub-panel for a logically ORed expression that is to comprise one or more logically ANDed sub-expressions.
- **AND Sub-Expressions - Single Row (Zoomed) View**
  Opens the Minor OR Sub-Expressions sub-panel for a logically ANDed expression that is to comprise one or more logically ORed sub-expressions.

SELECT is assigned to <F5> by default.

Multiple FIND & CHANGE Specification Panels

Multiple Find Commands (unformatted)

The **Multiple Find Commands** panel view for FIND operations on unformatted record data (ZZSGFSU4) is displayed when the option is taken to Specify Multiple FIND commands from the FSU (unformatted): Search Records using the FIND Command panel view of the File Search/Update/Copy/Remap (FSU) utility panels.
This panel is an interactive panel window (window class WINWIPO0) and includes an embedded table. Standard table edit primary and line commands must be used to insert, delete or exclude table rows as appropriate.

The table entries each identify an edit FIND operation. In the generated FSU command, these entries will occur in the order specified and will all be separated by either an AND or an OR logical operator. AND/OR logical operators indicate that either **all** FIND operations or **at least one** of the FIND operations respectively must be true in order for the record to be selected.

On completing entries in this panel view, execute BACK (assigned to <F3> by default) to close the panel and return to the utility panel view.

Table View

The panel initially displays the embedded table with column names **Rop**, **Find**, **Value**, **Scope**, **Left Bnd** and **Right Bnd**.
To display a table row in single row (zoomed) format, execute primary command ZOOM (assigned to <F17> by default) with the cursor positioned on the required table row entry.

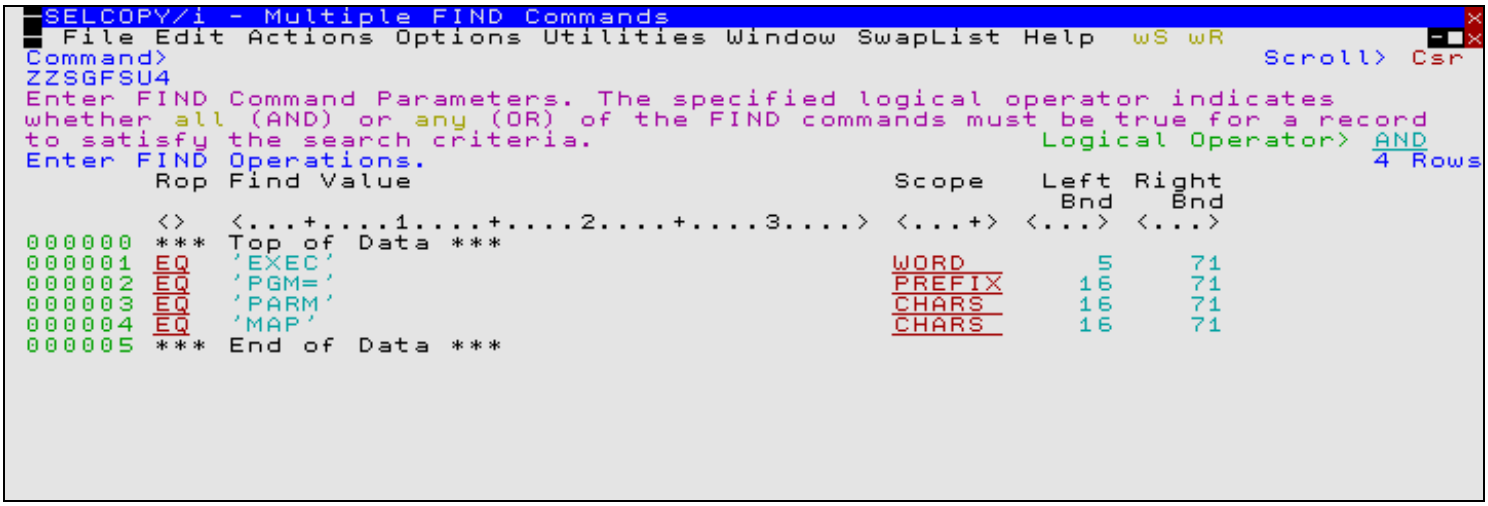


Figure 73. SELCOPY/i - Multiple FIND Commands.

Single Row (Zoomed) View

This panel displays the table column entries for an individual table row as panel input fields. Execute panel primary command BACK (assigned to <F3> by default) to close the zoomed panel view and return to the table view.

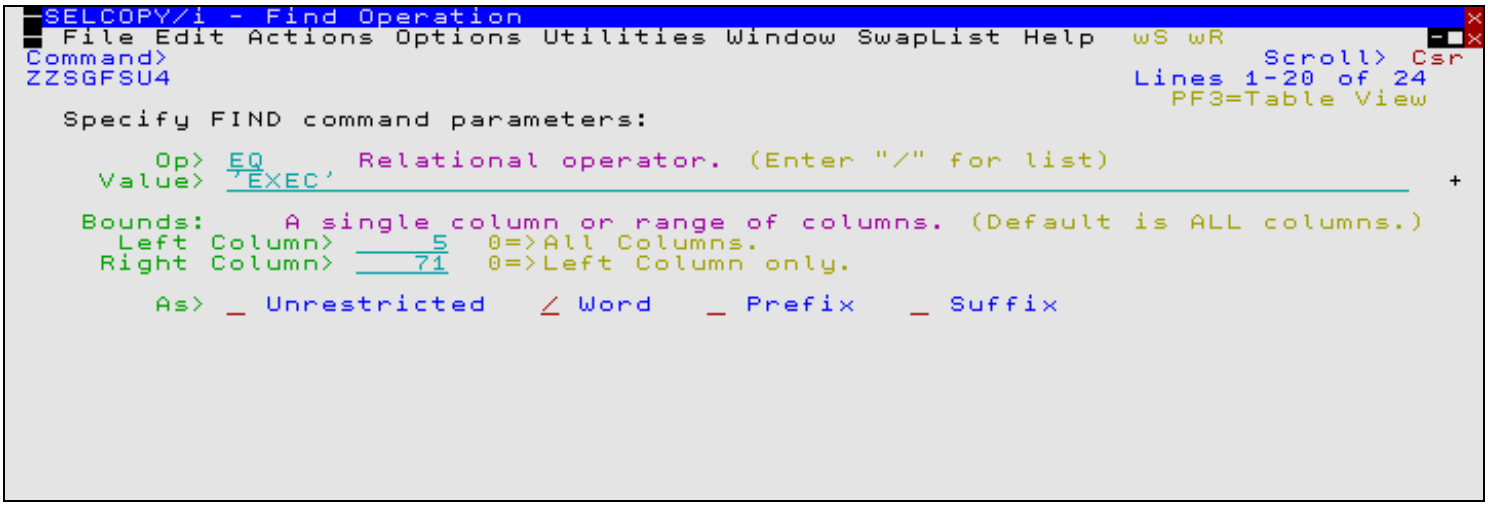


Figure 74. SELCOPY/i - Find Operation (Zoomed View).

Input Values

Logical Operator>
Specifies the logical operator (AND or OR) used to separate each of the specified FIND operations.

Op>
Identify the relational operator used when comparing the record data against the FIND search value.

If testing for a non-equality relationship using a character string search value, then the EBCDIC values assigned to characters in the search string and record data determine the relationship. (e.g. GT, LT)

Enter "/" to display a list of valid entries and a brief description.

Find Value / Value>
Specify the search value. This may be one of the following:

1. An unquoted numeric value.
2. A quoted character string (case insensitive search). e.g. 'John Doe' is the same as 'JOHN DOE'.
3. A quoted character string prefixed with "C" (case sensitive search). e.g. C'John Doe'.
4. A quoted hexadecimal string prefixed with "X". e.g. X'00'.
5. A quoted picture string prefixed with "P". e.g. P'USER###'.
   See the FIND command for details on supported picture string special characters.

Left/Right Bnd / Bounds:
Specify the record columns between which the search will occur. If the left bound value is 0 (zero), the search will include all columns within the length of the input records.

Left Column>
Specifies the start (or only) record column from which the scan for the search string will begin. Record data in positions or fields that occur before the left bound column value is not searched.

This field corresponds to SDE FIND parameter *pos1*.

Right Column>
Specifies the end record data column beyond which no part of the search value may be found. Only record data between the left and right bound columns will be searched.

If a 0 (zero) right bound column is specified then the right bound column is the left bound column number plus the length of the search value minus 1. i.e. the search string will only be found if it begins in the left bound column.

This field corresponds to SDE FIND parameter *pos2*.

Scope / As>
This field identifies restrictions to be applied to the position of the search string within record data.

In table view, enter "/" in this field to select from a list of possible values: CHAR (Unrestricted), WORD, PREFIX or SUFFIX.

In single row (zoomed) view, enter "/" in the appropriate radio button field.

Unrestricted	The search value may be found anywhere within the specified column bounds of the input records.
Word	The search value may only be found if it is complete word which falls within the specified column bounds of the input records. i.e. the value must either be preceded by a non-alphanumeric character or begin at the start column bound, and either precede a non-alphanumeric character or finish in the end column bound.
Prefix	The search value may only be found if it occurs at the start of a word and falls within the specified column bounds of the input records. i.e. the value must either be preceded by a non-alphanumeric character or begin at the start column bound, and be followed by an alphanumeric character.
Suffix	The search value may only be found if it occurs at the end of a word and falls within the specified column bounds of the input records. i.e. the value must be preceded by an alphanumeric character and must either precede a non-alphanumeric character or finish in the end column bound.

These option fields correspond to SDE FIND parameters PREFIX, SUFFIX and WORD.

Multiple Find Commands (formatted)

The **Multiple Find Commands** panel view for FIND operations on formatted record data (ZZSGFSU8) is displayed when the option is taken to Specify Multiple FIND commands from the FSU (formatted): Search Records using the FIND Command panel view of the File Search/Update/Copy/Remap (FSU) utility panels.
This panel is an interactive panel window (window class WINWIPO0) and includes an embedded table. Standard table edit primary and line commands must be used to insert, delete or exclude table rows as appropriate.

The table entries each identify an edit FIND operation. In the generated FSU command, these entries will occur in the order specified and will all be separated by either an AND or an OR logical operator. AND/OR logical operators indicate that either **all** FIND operations or **at least one** of the FIND operations respectively must be true in order for the record to be selected.

On completing entries in this panel view, execute BACK (assigned to <F3> by default) to close the panel and return to the utility panel view.

Table View

The panel initially displays the embedded table with column names **Rop**, **Find**, **Value**, **Scope**, **Left Column** and **Right Column**.

If a FIND operation is to be restricted to an individual column or range of columns then enter a value in the **Left Column** and **Right Column** fields as appropriate to open the SDE: Select Field Name panel and select a single entry from a list of formatted record fields. To display a table row in single row (zoomed) format, execute primary command ZOOM (assigned to <F17> by default) with the cursor positioned on the required table row entry.

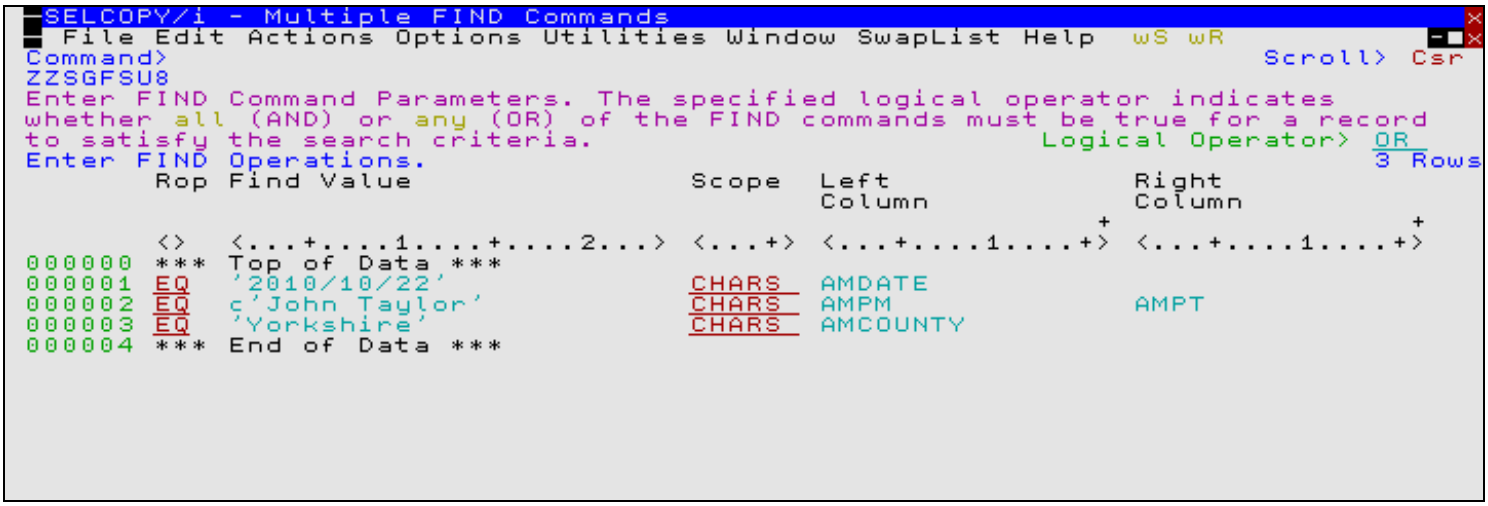


Figure 75. SELCOPY/i - Multiple (formatted) FIND Commands.

Single Row (Zoomed) View

This panel displays the table column entries for an individual table row as panel input fields. Execute panel primary command BACK (assigned to <F3> by default) to close the zoomed panel view and return to the table view.

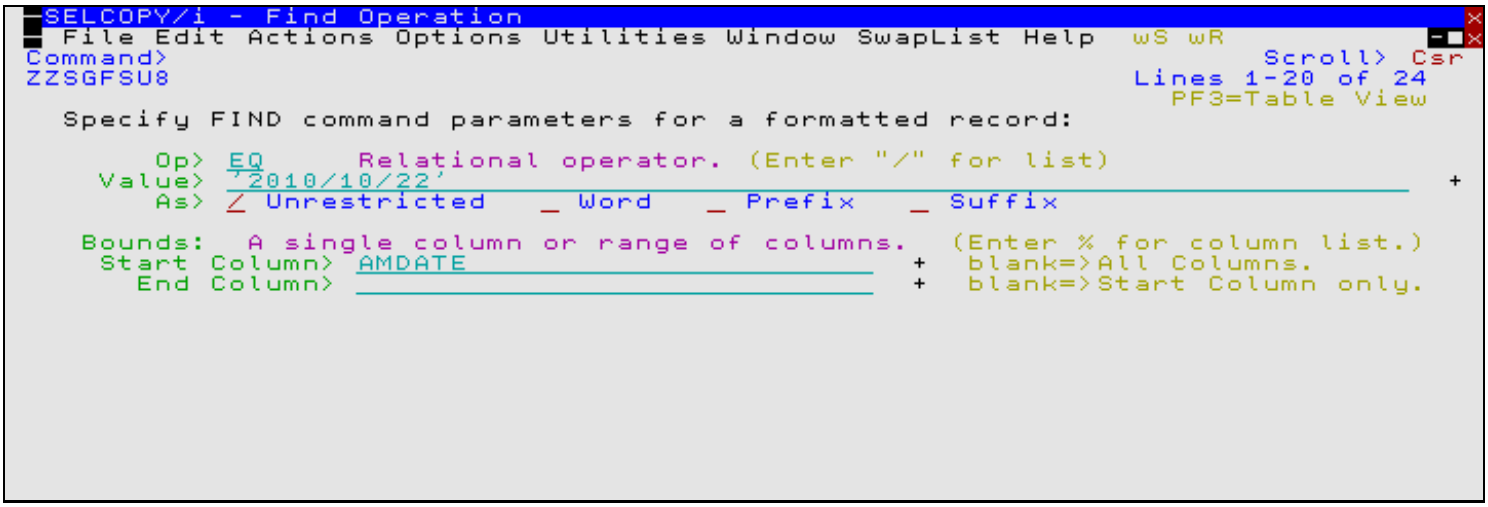


Figure 76. SELCOPY/i - (formatted) Find Operation (Zoomed View).

Input Values

Logical Operator>
Specifies the logical operator (AND or OR) used to separate each of the specified FIND operations.

ROp / Op>
Identify the relational operator used when comparing the field data against the search value.

If testing for a non-equality relationship using a character string search value, then the EBCDIC values assigned to characters in the search string and record data determine the relationship. (e.g. GT, LT)

Enter "/" to display a list of valid entries and a brief description.

Find Value / Value>
Specify the search value. This may be one of the following:

1. Applicable to fields of any data type, an unquoted numeric value. For numeric fields, the numeric search value and source field data are converted so that they are of the same data type prior to performing an arithmetic comparison.
2. Applicable only to character fields, a quoted character string (case insensitive search). e.g. 'John Doe' is the same as 'JOHN DOE'.
3. Applicable only to character fields, a quoted character string prefixed with "C" (case sensitive search). e.g. C'John Doe'.
4. Applicable only to character fields, a quoted hexadecimal string prefixed with "X". e.g. X'00'.
5. Applicable only to character fields, a quoted picture string prefixed with "P". e.g. P'USER###'.
   See the FIND command for details on supported picture string special characters.

Left/Right Column / Bounds:
Optionally specify the name record columns (fields) between which the search will occur. If the left bound value is blank, the search will include all columns within the length of the input records.

Left/Start Column
Specifies the start (or only) record field from which the scan for the search value will begin. Record data in fields that occur before this start column value is not searched.

This field corresponds to SDE FIND parameter *field_col/field_col1*.

Right/End Column
Specifies the end record field. Only record data between the start and end columns will be searched.

If no end column is specified then only the start column field is searched.

This field corresponds to SDE FIND parameter *filed_col2*.

As>
Applicable only to character fields (data-type "AN"), enter "/" in the appropriate radio button field to apply restrictions to the position of the search string within the field as follow:

Unrestricted	The search value may be found anywhere within the character field.
Word	The search value may only be found if it is complete word within the character field. i.e. the value must either be preceded by a non-alphanumeric character or begin at the first character of the field, and either precede a non-alphanumeric character or finish in the last character of the field.
Prefix	The search value may only be found if it occurs at the start of a word within the character field. i.e. the value must either be preceded by a non-alphanumeric character or begin at the first character of the field, and be followed by an alphanumeric character.
Suffix	The search value may only be found if it occurs at the end of a word within the character field. i.e. the value must be preceded by an alphanumeric character and must either precede a non-alphanumeric character or finish in the last character of the field.

These option fields correspond to SDE FIND parameters PREFIX, SUFFIX and WORD.

Multiple Change Commands (unformatted)

The **Multiple Change Commands** panel view for CHANGE operations on unformatted record data (ZZSGFSU6) is displayed when the option is taken to Specify Multiple CHANGE commands from the FSU (unformatted): Change record data using the CHANGE command panel view of the File Search/Update/Copy/Remap (FSU) utility panels.

This panel is an interactive panel window (window class WINWIPO0) and includes an embedded table.

Standard table edit primary and line commands must be used to insert, delete or exclude table rows as appropriate.

The table entries each identify an edit CHANGE operation. In the generated FSU command, these entries will occur in the order specified and will all be separated by either an AND or an OR logical operator. AND/OR logical operators indicate that either **all** CHANGE operations will be actioned or only the **first successfully executed** CHANGE operation will be actioned respectively.

On completing entries in this panel view, execute BACK (assigned to <F3> by default) to close the panel and return to the utility panel view.

Table View

The panel initially displays the embedded table with column names **Rop**, **FROM:**, **TO:**, **Scope**, **Occur**, **Src**, **Left Bnd** and **Right Bnd**.

To display a table row in single row (zoomed) format, execute primary command ZOOM (assigned to <F17> by default) with the cursor positioned on the required table row entry.

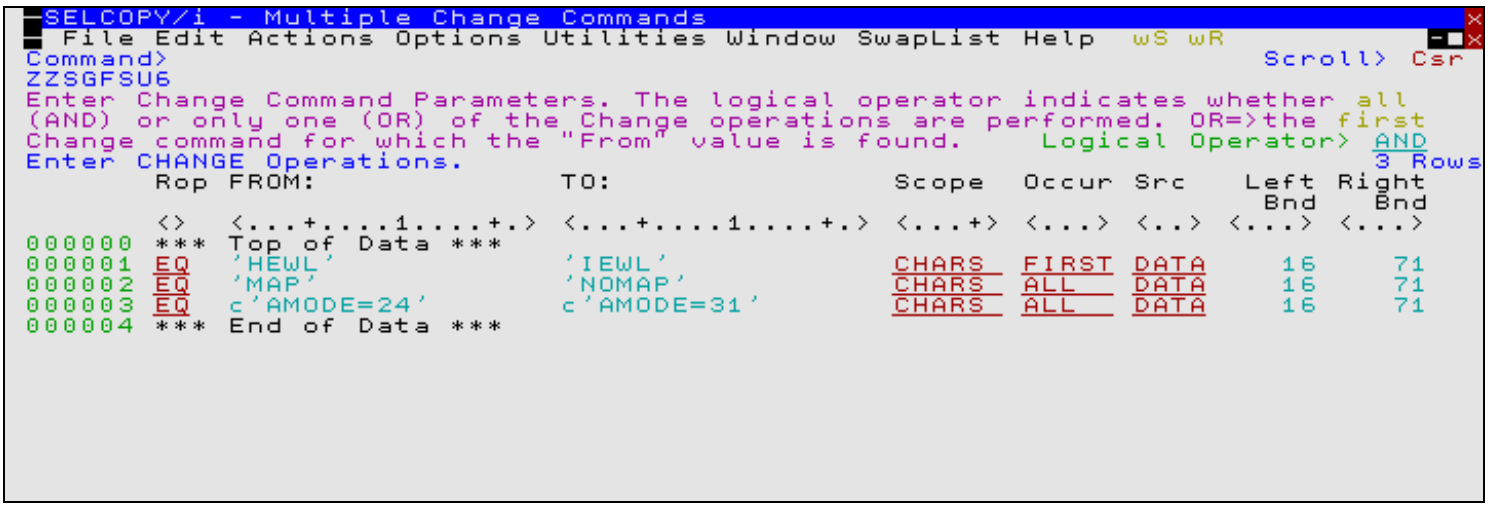


Figure 77. SELCOPY/i - Multiple Change Commands.

Single Row (Zoomed) View

This panel displays the table column entries for an individual table row as panel input fields. Execute panel primary command BACK (assigned to <F3> by default) to close the zoomed panel view and return to the table view.

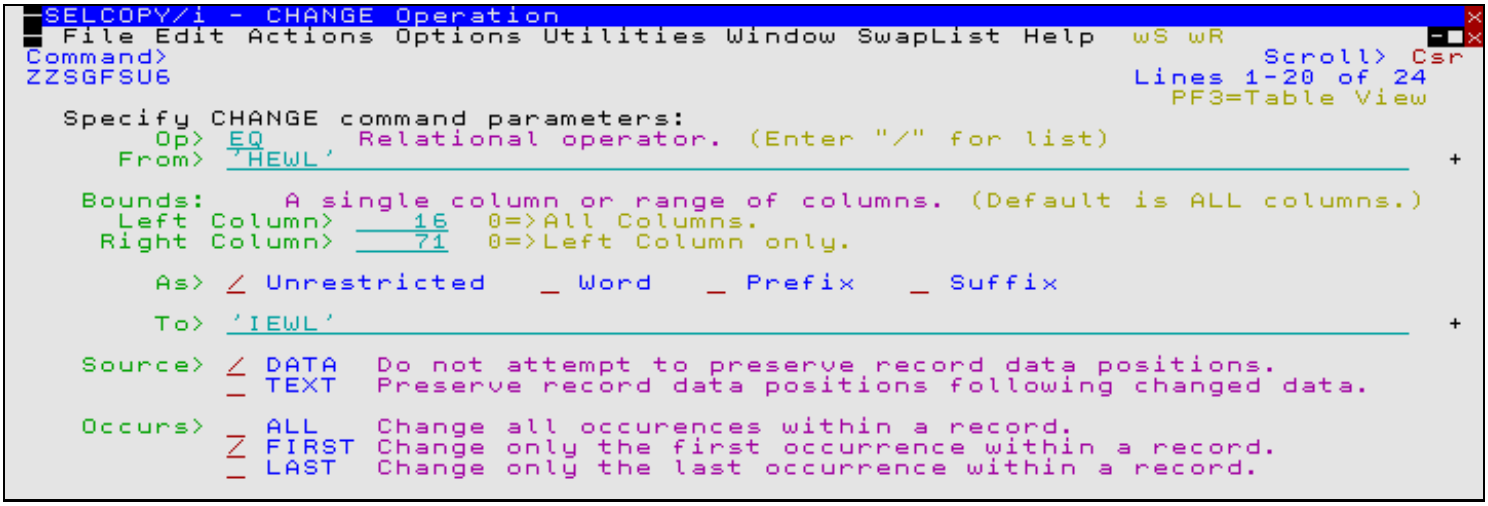


Figure 78. SELCOPY/i - Change Operation (Zoomed View).

Input Values

Logical Operator>
Specifies the logical operator (AND or OR) used to separate each of the specified CHANGE operations.

Rop / Op>
Identify the relational operator used when comparing the record data against the CHANGE operation search value.

If testing for a non-equality relationship using a character string search value, then the EBCDIC values assigned to characters in the search string and record data determine the relationship. (e.g. GT, LT)

Enter "/" to display a list of valid entries and a brief description.

FROM: / From>
Specify the CHANGE operation search value. This may be one of the following:

1. An unquoted numeric value which will be processed as a character string.
2. A quoted character string (case insensitive search). e.g. 'John Doe' is the same as 'JOHN DOE'.
3. A quoted character string prefixed with "C" (case sensitive search). e.g. C'John Doe'.
4. A quoted hexadecimal string prefixed with "X". e.g. X'00'.
5. A quoted picture string prefixed with "P". e.g. P'USER###'.
   See the CHANGE command for details on supported search value picture string special characters.

TO: / To>
Specify the CHANGE operation replace value. This may be specified in any of the formats described for the search value above.

For a replace value specified as a quoted picture string, see the CHANGE command for details on supported replace value picture string special characters.

Left/Right Bnd / Bounds:
columns between which both the search and replace values must occur in order to perform a successful CHANGE operation. If the left bound value is 0 (zero), the search will include all columns within the length of the input records.

If an in-place update is to be performed, the replace string must not result in a change to the record length. However, if records are to be copied the replace string may extend the length of the record to a value not greater than the defined maximum record length.

Left Column
Specifies the left (or only) record column from which the scan for the CHANGE operation search string will begin. Record data in positions or fields that occur before this column value is excluded from the CHANGE operation.

This field corresponds to SDE CHANGE parameter *pos1*.

Right Column
Specifies the end record data column beyond which no part of the search value may be found and no part of the replace string may occupy. Only record data between the left and right bound columns will be searched and potentially replaced.

If a 0 (zero) right bound column is specified then the right bound column is the left bound column number plus the length of the search value minus 1. i.e. the search value will only be found if it begins in the left bound column.

This field corresponds to SDE CHANGE parameter *pos2*.

Scope / As>
This field identifies restrictions to be applied to the position of the CHANGE search string within record data.

In table view, enter "/" in this field to select from a list of possible values: CHAR (Unrestricted), WORD, PREFIX or SUFFIX.

In single row (zoomed) view, enter "/" in the appropriate radio button field.

Unrestricted	The search value may be found anywhere within the specified column bounds of the input records.
Word	The search value may only be found if it is a complete word which falls within the specified column bounds of the input records. i.e. the value must either be preceded by a non-alphanumeric character or begin at the start column bound, and either precede a non-alphanumeric character or finish in the end column bound.
Prefix	The search value may only be found if it occurs at the start of a word and falls within the specified column bounds of the input records. i.e. the value must either be preceded by a non-alphanumeric character or begin at the start column bound, and be followed by an alphanumeric character.
Suffix	The search value may only be found if it occurs at the end of a word and falls within the specified column bounds of the input records. i.e. the value must be preceded by an alphanumeric character and must either precede a non-alphanumeric character or finish in the end column bound.

These option fields correspond to SDE CHANGE parameters PREFIX, SUFFIX and WORD.

Src / Source>
Identifies the format of the record source character data:

DATA
Indicates that records are to be treated as data so that inserting the replace value performs no special treatment of any multiple, consecutive blanks that occur to the right of the replaced data.

If the length of the replace value is greater than that of the search value, all data to the right of the replaced data will be shifted right.

If the length of the replace value is less than that of the search value, all data to the right of the replaced data will be shifted left.

TEXT
Indicates that records are to be treated as formatted character text so that, in an effort to maintain the record positions of non-blank text to the right of the changed text, when inserting the replacement text, consecutive blanks that occur to the right of this text are absorbed or added as approriate.

If the length of the replace value is greater than that of the search value, consecutive balnks to the right of the replaced text will be absorbed before non-blank text is shifted right. Note that blank delimitation of non-blank text is preserved.

If the length of the replace value is less than that of the search value, then blanks will be inserted immediately before the first blank character occurring to the right of the replaced text. The number of blanks inserted will be equal to the difference between the lengths of the search and replace values.

Occur / Occurs>
Identifies the occurrence of the search value within the input record to which the CHANGE operation will apply:

ALL
Attempt to change all occurrences of the search value found within the input record.

FIRST
Attempt to change only the first occurrence of the search value found within the input record.

LAST
Attempt to change only the last occurrence of the search value found within the input record.

Multiple Change Commands (formatted)

The **Multiple Change Commands** panel view for CHANGE operations on unformatted record data (ZZSGFSU7) is displayed when the option is taken to Specify Multiple CHANGE commands from the FSU (formatted): Change record data using the CHANGE command panel view of the File Search/Update/Copy/Remap (FSU) utility panels.
This panel is an interactive panel window (window class WINWIPO0) and includes an embedded table. Standard table edit primary and line commands must be used to insert, delete or exclude table rows as appropriate.

The table entries each identify an edit CHANGE operation. In the generated FSU command, these entries will occur in the order specified and will all be separated by either an AND or an OR logical operator. AND/OR logical operators indicate that either **all** CHANGE operations will be actioned or only the **first successfully executed** CHANGE operation will be actioned respectively.

On completing entries in this panel view, execute BACK (assigned to <F3> by default) to close the panel and return to the utility panel view.

Table View

The panel initially displays the embedded table with column names **Rop**, **FROM:**, **TO:**, **Scope**, **Occur**, **Src**, **Left Column** and **Right Column**.

If a CHANGE operation is to be restricted to an individual column or range of columns then enter a value in the **Left Column** and **Right Column** fields as appropriate to open the SDE: Select Field Name panel and select a single entry from a list of formatted record fields. To display a table row in single row (zoomed) format, execute primary command ZOOM (assigned to <F17> by default) with the cursor positioned on the required table row entry.

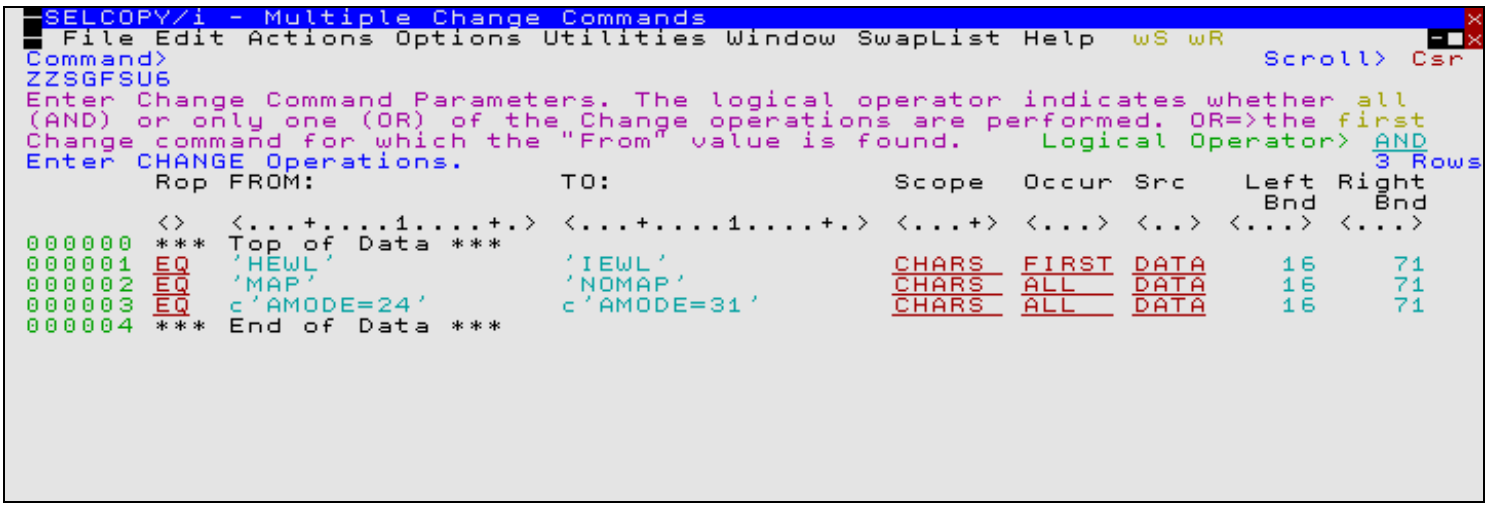


Figure 79. SELCOPY/i - Multiple FIND Commands.

Single Row (Zoomed) View

This panel displays the table column entries for an individual table row as panel input fields. Execute panel primary command BACK (assigned to <F3> by default) to close the zoomed panel view and return to the table view.

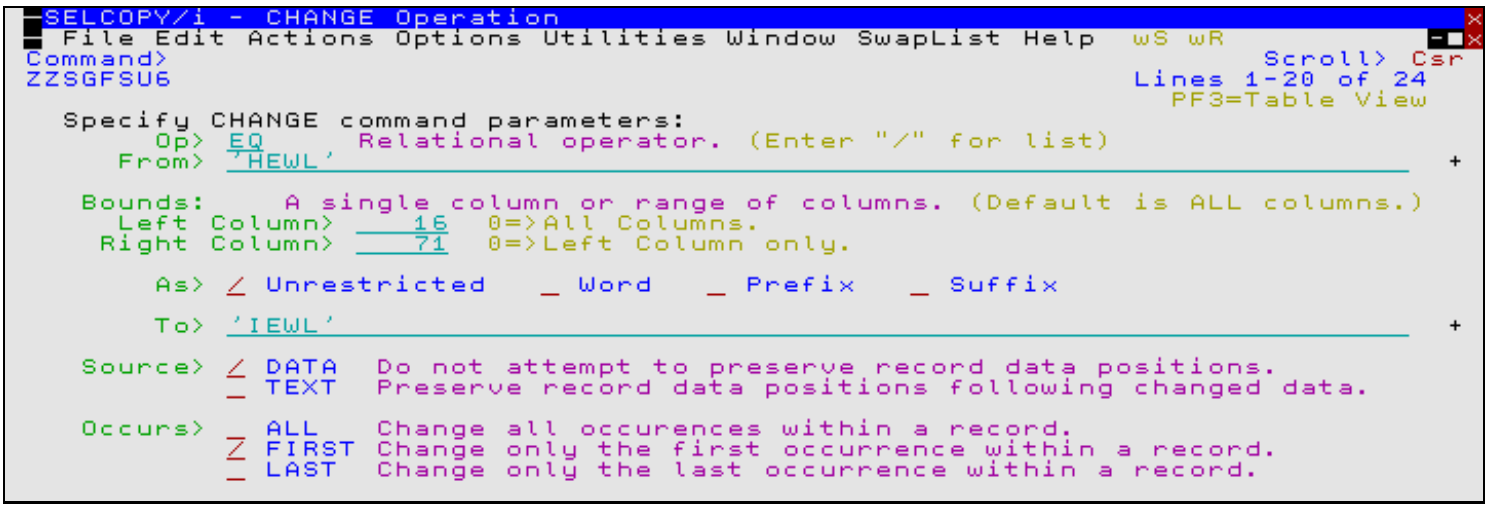


Figure 80. SELCOPY/i - Find Operation (Zoomed View).

Input Values

Logical Operator>
Specifies the logical operator (AND or OR) used to separate each of the specified CHANGE operations.

Rop / Op>
Identify the relational operator used when comparing the field data against the CHANGE operation search value.

If testing for a non-equality relationship using a character string search value, then the EBCDIC values assigned to characters in the search string and record data determine the relationship. (e.g. GT, LT)

Enter "/" to display a list of valid entries and a brief description.

FROM: / From>
Specify the CHANGE operation search value. This may be one of the following:

1. Applicable to fields of any data type, an unquoted numeric value. For numeric fields, the numeric search value and source field data are converted so that they are of the same data type prior to performing an arithmetic comparison.
2. Applicable only to character fields, a quoted character string (case insensitive search). e.g. 'John Doe' is the same as 'JOHN DOE'.
3. Applicable only to character fields, a quoted character string prefixed with "C" (case sensitive search). e.g. C'John Doe'.
4. Applicable only to character fields, a quoted hexadecimal string prefixed with "X". e.g. X'00'.
5. Applicable only to character fields, a quoted picture string prefixed with "P". e.g. P'USER###'.
   See the CHANGE command for details on supported search value picture string special characters.

TO: / To>
Specify the CHANGE operation replace value. This may be specified in any of the formats described for the search value above. For numeric fields, the replace value will be converted into the appropriate source data format.

For a replace value specified as a quoted picture string, see the CHANGE command for details on supported replace value picture string special characters.

Left/Right Column / Bounds:
Optionally specify the name record columns (fields) between which the CHANGE operation will occur. If the left bound value is blank, the search will include all columns within the length of the input records.

Left/Start Column
Specifies the start (or only) record field from which the scan for the CHANGE operation search value will begin. Record data in fields that occur before this start column value is not included.

This field corresponds to SDE CHANGE parameter *field_col/field_col1*.

Right/End Column
Specifies the end record field. Only record data between the start and end columns will be searched and potentially replaced.

If no end column is specified then only the start column field is searched.

This field corresponds to SDE CHANGE parameter *filed_col2*.

Scope / As>
Applicable only to character fields (data-type "AN"), this field identifies restrictions to be applied to the position of the CHANGE search value within the formatted record field data.

In table view, enter "/" in this field to select from a list of possible values: CHAR (Unrestricted), WORD, PREFIX or SUFFIX.

In single row (zoomed) view, enter "/" in the appropriate radio button field.

Unrestricted	The search value may be found anywhere within the character field.
Word	The search value may only be found if it is complete word within the character field. i.e. the value must either be preceded by a non-alphanumeric character or begin at the first character of the field, and either precede a non-alphanumeric character or finish in the last character of the field.
Prefix	The search value may only be found if it occurs at the start of a word within the character field. i.e. the value must either be preceded by a non-alphanumeric character or begin at the first character of the field, and be followed by an alphanumeric character.
Suffix	The search value may only be found if it occurs at the end of a word within the character field. i.e. the value must be preceded by an alphanumeric character and must either precede a non-alphanumeric character or finish in the last character of the field.

These option fields correspond to SDE CHANGE parameters PREFIX, SUFFIX and WORD.

Src / Source>
For character fields only, these options identify the format of the source character data:

DATA
: Indicates that character fields are to be treated as data so that inserting the replace value performs no special treatment of any multiple, consecutive blanks that occur to the right of the replaced data.

  If the length of the replace value is greater than that of the search value, all data to the right of the replaced data will be shifted right.

  If the length of the replace value is less than that of the search value, all data to the right of the replaced data will be shifted left.

TEXT
: Indicates that character fields are to be treated as formatted character text so that, in an effort to maintain the position of non-blank text which appear to the right of the changed text, when inserting the replacement text, consecutive blanks that occur to the right of this text are absorbed or added as approriate.

  If the length of the replace value is greater than that of the search value, consecutive balnks to the right of the replaced text will be absorbed before non-blank text is shifted right. Note that blank delimitation of non-blank text is preserved.

  If the length of the replace value is less than that of the search value, then blanks will be inserted immediately before the first blank character occurring to the right of the replaced text. The number of blanks inserted will be equal to the difference between the lengths of the search and replace values.

Occur / Occurs>
: Identifies the occurrence of the search value within the input record to which the CHANGE operation will apply:

ALL
: Attempt to change all occurrences of the search value found within the input record fields.

FIRST
: Attempt to change only the first occurrence of the search value found within the input record fields

LAST
: Attempt to change only the last occurrence of the search value found within the input record fields

File Search/Update/Copy/Remap Output

Report Format

The report generated by the file Search/Update/Copy/Remap utility is a structured data file designed to be browsed (and optionally printed) using a structure definition object (SDO) within a SELCOPY/i session.

The associated SDO is automatically generated when the utility is executed. The SDO dataset name is always the DSN of the report with a suffix of "**.SDO**".

If a report output file DSN has been specified by the user, then the data set should be already allocated large enough to receive the report data. A non-existant report datasetset will be created with default space allocation of 1 cylinder and LRECL=32756, BLKSIZE=23760.

If not specified, the default DSN for the report and SDO data sets when run interactively is "*prefix*.FSU.Dyyyyddd.Thhmmss" and "*prefix*.FSU.Dyyyyddd.Thhmmss.SDO" respectively. This is so that multiple executions of FSU may be run without exiting from the report output of any previous run.

In batch (where the default action is to print the report to SDEPRINT), the default DSN for the report and SDO data sets is "*prefix*.SELCOPYI.FSU.REPORT" and "*prefix*.SELCOPYI.FSU.REPORT.SDO" respectively.

The high level qualifier, *prefix*, is the value assigned to **System.UserDSNPrefix** in the SELCOPY/i User INI file.

If the job stream JCL is generated via the File Search/Update/Copy/Remap Panel, then DD statements with DISP=NEW will be generated for any explcitly specified report file DSN (and SDO structure) data sets, if they do not already exist.

If the utility is run in the **foreground** of a SELCOPY/i session, then the output report file is generated in storage only and is displayed and updated automatically as the utility executes. The display is refreshed every second allowing the user to view the progress and, if necessary, interrupt the execution using the Attention key.

When closing an in-storage report, the user will be prompted to save the report file and accompanying SDO structure if either of the following are true:

- File update processing has occurred with the immediate UPDATE option set. Following an UPDATE, it is strongly recommended that the user save these files, so providing an audit trail and, if required, the necessary input to the File Update Undo facility which reverses changes to updated records.
- A READ or UPDATE I/O error occurs before the run has completed. This ensures that the audit trail exists even if control is not returned from the system routine.

In all other circumstances, the user will not be prompted to save a permanent copy of the in-storage report unless the "FSUEND" command is executed to close the report window. If the user chooses to save the report, the report file will be saved as a VSAM ESDS data set by default and the accompanying SDO as a physical sequential data set.

Prior to executing FSUEND to save a foreground report, the "SET FILEID" and/or "SET DSORG" commands may be issued to override the DSN (with optional library member name) and file organisation of the saved report dataset.

Following execution of the utility (in batch or in the foreground) the saved report file may subsequently be browsed from your SELCOPY/i session using any of the following methods:

- Issue the command `FSUOUT` *`report_file_name`*.
- Issue the line-command `FO` against the report DSN in a data set list or VTOC list window, or against the report member in a Library list window.

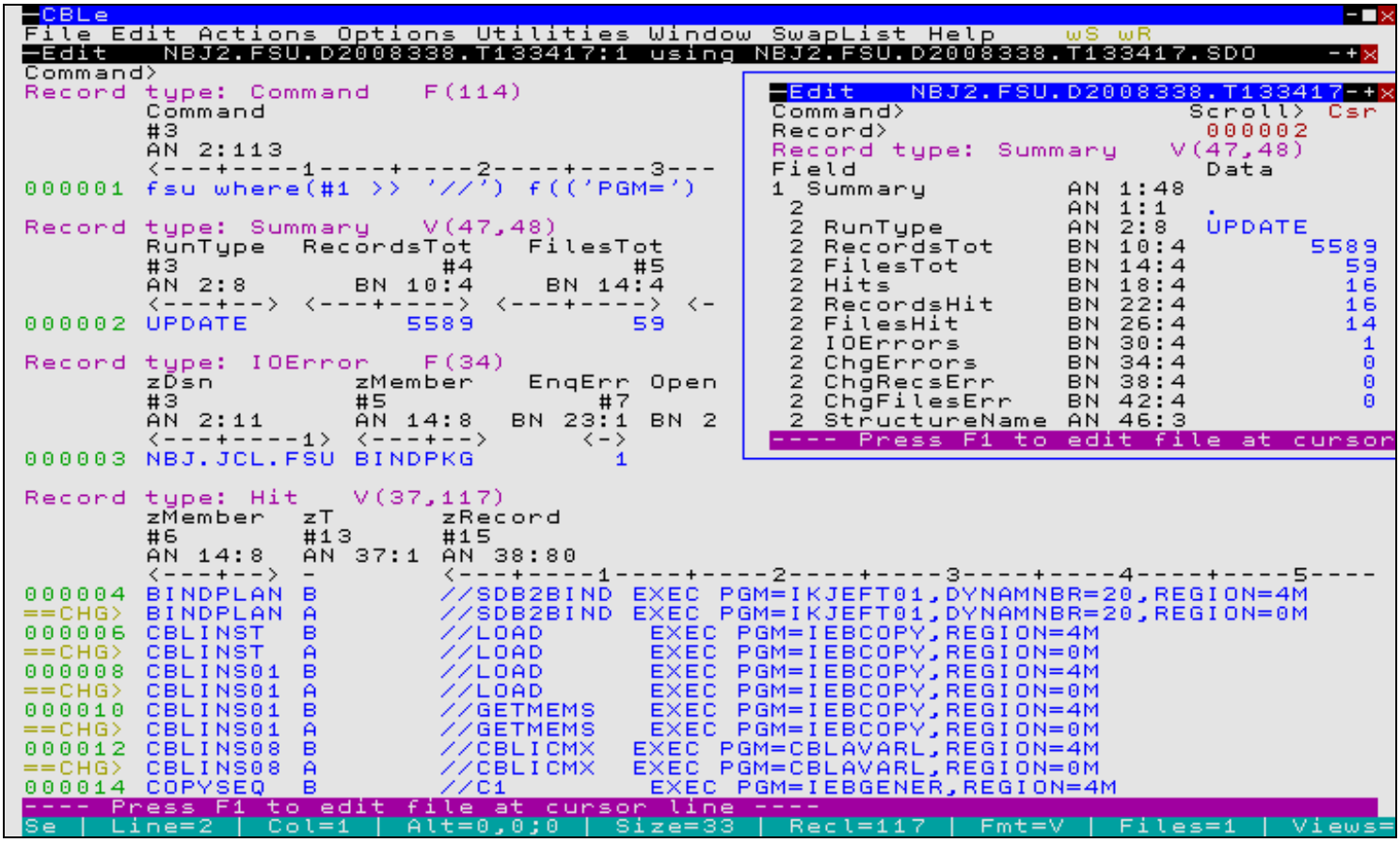


Figure 81. File Search/Update/Copy/Remap Output Output.

In the FSU output displayed in Figure 81., the user has hit the PF17 (Shift-F5) key (ZoomW) on the Summary record, to display the record's fields in single view, and also executed the following SDE SELECT command to restrict the fields displayed in records of record type Hit:

```
SELECT   zMember, zT, zRecord    FROM Hit
```

The FSU report output consists of 5 record types: 1 Command record, 1 Summary record, 0 or more Hit records, 0 or more IOError records and 0 or more Alias records.

Record Type: Command

Contains information of the FSU command stream use to execute the File Search/Update/Copy/Remap utility.

`Timestamp`
: The date and local time at which the FSU command was executed to generate this report.

`Command`
: A character field containing the FSU command executed (directly or via the File Search/Update/Copy/Remap Panel). The Command record is located at the top of the output report.

Record Type: Summary

Contains statistical fields providing totals for the File Search/Update/Copy/Remap Output execution as follow:

`RunType`
: Describes the type of execution of the File Search/Update/Copy/Remap Output utility and so governs the format of the Hit records.

Operation	RunType
Unformatted/Formatted Search	FIND
Unformatted/Formatted Update	UPDATE or NOUPDATE
Unformatted/Formatted Copy or Remap	COPY

For Update operations, RunType is "NOUPDATE" if update has been suppressed by the FSU command NOUPDATE

(default) parameter or "Immediate UPDATE" has been deactivated in the utility panel.

`RecordsTot`
The total number of input records successfully read from all selected files.

`FilesTot`
The total number of files that match the supplied fileid mask(s).

`Hits`
For RunType "FIND", this is the total number of occurrences of the FIND search string(s) found within the input records that satisfy the logical combination of FIND conditions. e.g. `FIND ( ('A') AND ('B') )` will increment the Hit total for every occurrence of 'A' and 'B' in records that contain both 'A' and 'B'.

For RunType "NOUPDATE" and "UPDATE", this is the total number of occurrences of the CHANGE search string(s) found within the selected input records (i.e. input records that satisfy the VIEW, WHERE and/or FIND criteria.)

`RecordsHit`
For RunType "FIND", this is the total number of input records that satisfy the supplied WHERE clause and/or FIND search string criteria.

For RunType "NOUPDATE" and "UPDATE", this is the total number of selected input records that satisfy the CHANGE search string criteria.

RecordsHit corresponds with the number of input records that are of the record type "Hit Records" and so are displayed in the report output.

`FilesHit`
The total number of files that contains at least one record that includes a hit.

`RemapErrors`
For Formatted Remap operations only, the total number of files for which an error has occurred on attempting to remap source fields of one data type to target fields of a different data type.

`IOErrors`
The total number of files for which I/O errors that have occurred during execution. An IOError record is displayed for each

`ChgErrors`
The total number of CHANGE errors. i.e. the total number of occurrences of a CHANGE search string, within all selected records, that cannot be updated with the CHANGE replace string.
For RunType "FIND", this value is aways 0 (zero).

`ChgRecsErr`
The total number of selected input records for which a CHANGE error has occurred.
For RunType "FIND", this value is aways 0 (zero).

`ChgFilesErr`
The total number of files that contain at least one record for which a CHANGE error has occurred.
For RunType "FIND", this value is aways 0 (zero).

`StructureName`
The DSN and member name of the structure (SDO) specified on the USING field/parameter for Formatted File Search/Update/Copy/Remap Output.
For Unformatted File Search/Update/Copy/Remap Output, this field value is always blank.

Record Type: Hit

The format of the Hit records depend on the Summary record "RunType" field, as follows:

1. For RunType "FIND" or "COPY" where search criteria have been specified, this field displays every record that satisfies all the specified search criteria. i.e. the VIEW record type (for Formatted File Search/Update/Copy/Remap Output), the WHERE clause and the FIND search string(s). Note that no CHANGE operation has been specified.

2. For RunType "NOUPDATE", "UPDATE" or "COPY" where a CHANGE operation has been specified, this field displays a pair of records for every input record that satisfies the CHANGE arguments. The first record of the pair displays the original, unaltered record data, the second displays the record data after the CHANGE operation(s) have been executed.

   Depending on whether the CHANGE operation(s) are successful, the prefix area of the line displaying the updated record will contain the line flag "==CHG>" or, if an error has occurred, "==ERR>".

   For update operations with NOUPDATE in effect, this allows the user the opportunity to check the changed data before re-running the utility with UPDATE to action the changes.

The Hit record type contains 2 group fields, "z" and "zRecord", where "z" includes information fields relating to the record, and "zRecord" includes all the record data field(s). The Hit record type has been designed this way so that the user can suppress or include all fields within either field group by specifying the group field name as the argument of a SELECT command. e.g. `SELECT zRecord` will display only the field data and suppress the information fields.

The Hit record type field structures are as follow:

`z`

The structure including all the information fields.

`zFileId`

A sub-structure containing just the zDsn and optional zMember fields.

`zDsn`

The DSN (or HFS path name) containing the reported record.

`zMember`

The PDS/PDSE member containing the reported record.
This field is only present if at least one PDS/PDSE data set is included by the input fileid mask(s). For non-PDS/PDSE data sets, this field contains blanks.

`zGenA`

The absolute generation number of the PDSE member generation containing the reported record.
This field is only present if at least one PDSE member generation mask is included in the input fileid mask(s). Blanks are displayed where member generations are not applicable.

`zGenR`

The relative generation number of the PDSE member generation containing the reported record.
This field is only present if at least one PDSE member generation mask is included in the input fileid mask(s). Blanks are displayed where member generations are not applicable.

`zRecNo`

The record number of the record within the data set.

`zHitNo`

The hit count number of the record within the data set. The "zHitNo" field is incremented by one for each new record within the data set that satisfies the search criteria for the particular RunType. The "zHitNo" count is reset to zero for each new input data set. Use `WHERE zHitNo=1` to display a list of all data sets (and library members) containing at least one hit.

`zLrecl`

The record length of the record within the data set.

`zHits`

The total number of occurrences of the search string(s) within the record.

For all RunType "FIND"/"COPY" Hit records or RunType "NOUPDATE"/"UPDATE"/"COPY" Hit records with "zT" field flag set to "B", this is the number of FIND search string occurrences.

For all RunType "NOUPDATE"/"UPDATE"/"COPY" Hit records with "zT" field flag set to "A", this is the number of CHANGE search string occurrences.

`zErrs`

For RunType "NOUPDATE", "UPDATE" and "COPY" involving a CHANGE operation, the total number of occurrences of a CHANGE search string within the record that cannot be updated with the CHANGE replace string.

For RunType "FIND", this field is omitted.

`zT`

Included only for RunType "NOUPDATE", "UPDATE" and "COPY" involving a CHANGE operation, this field displays the record image flag which may be one of the following:

B	Indicates that the record data that follows represents the record data **Before** the CHANGE operation(s) are applied.
A	Indicates that the record data that follows represents the record data **After** the CHANGE operation(s) are applied. Note that the record data will be unchanged if the values in the fields "zHits" and "zErrs" are equal.

For RunType "FIND", this field is omitted unless "CONTEXT n" is specified in which case the flag will contain one of the following:

H	Indicates a Hit record, which is hilighted in green.
L	Indicates a Leading context record.
T	Indicates a Trailing context record.

`zRecord`

The structure including all the record data fields.

For Unformatted File Search/Update/Copy/Remap Output, the "zRecord" field contains the unexpanded record data as a single character field of length equal to the record length.

For Formatted File Search/Update/Copy/Remap Output, the "zRecord" field contains the expanded record data mapped with the field names defined by the record type (RTO).

Record Type: IOError

Contains information relating to an I/O error that has occurred when opening, reading or updating the file. IOError records are located amongst the Hit records, as I/O errors are encountered.

`zDsn`
: The DSN (or HFS path name) for which the I/O error occurred.

`zMember`
: The PDS/PDSE member for which the I/O error occurred.
This field is only present if the at least one PDS/PDSE data set is included by the input fileid mask(s). For non-PDS/PDSE data sets, this field contains blanks.

`EnqErr`
: 1 if the error occurred when attempting to obtain an exclusive SPFEDIT ENQ for UPDATE on the file, otherwise 0.

`OpenErr`
: 1 if the error occurred when attempting to open the file, otherwise 0.

`ReadErr`
: 1 if the I/O error occurred when attempting to read a block of data from the file, otherwise 0.

`UpdErr`
: 1 if the I/O error occurred when attempting to re-write (update-in-place) a record to the file, otherwise 0.

`OutErr`
: 1 if the I/O error occurred when attempting to write a block of data to the OUTPUT file, otherwise 0.

`RecordsRead`
: The number of records successfully read before the I/O error occurred.

`RecordsUpd`
: The number of records successfully updated before the I/O error occurred.

Record Type: Alias

Contains information relating to aliases of library members. An Alias report record is generated for every member and member alias that contains a hit for WHERE/FIND search criteria or a CHANGE operation.

When processing PDS/PDSE library members, member records may be searched or changed having been accessed via the original member name or a member alias name. Once the member records have been processed, they will not be processed again via another alias name or their member name.

`zDsn`
: The DSN of the PDS/PDSE library.

`zMember`
: The PDS/PDSE member or member alias name.

`zAliasOf`
: The PDS/PDSE member name for which this library entry is an alias.

`zHitRef`
: The member or alias name for which associated "Hit" report records contain the search or change results applicable to the alias or member name identified by this "Alias" report record.

Record Type: Record

Contains a single, variable length character field "Record" displaying any error messages that have been generated by the File Search/Update/Copy/Remap utility.

Function Keys

<PF1>	Display context sensitive help.
<PF2>	Display the report record in a new window in single format (vertical) view. In single format view, use **<PF10>**/**<PF11>** to display the previous/next report record respectively.
<F16>	Display the SDE Edit/Browse utility menu. This includes show and hide of report records based on their type, and alter the display of report record fields.

<PF6>	Applicable to report records of record type Hit or IOError only, <PF1> edits the file(s), referenced by "zDSN" and "zMember" fields in the focus report record, and scrolls directly to the reported record. If the input records were formatted using an SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file to map the record data fields, then SDE EDIT is performed using the existing or generated SDO structure. For unformatted records, the SELCOPY/i text editor is used to edit the records.

File Update Undo

Overview

The File Update Undo utility (FSUUNDO) allows the user to restore updated records from any accidental or erroneous execution of the File Search/Update/Copy/Remap utility (FSU) where update of record data has occured.

When FSU is executed to change and immediately UPDATE data set records, the original record data, before execution of the change operation(s), is recorded in the FSU output report data set. FSUUNDO uses these report records as part of its processing and so will only operate successfully if the FSU report data set exists.

Therefore, it is strongly recommended that, when prompted on exit from the report data set, users elect to save the report and its accompanying SDE structure (SDO), for audit purposes and also for subsequent execution of FSUUNDO if required.

FSU report output reflecting FIND or NOUPDATE run types need not be saved. If used as input to FSUUNDO, FSU report output of run type FIND will return an error.

FSUUNDO generates a SELCOPY control statements that performs the following:

1. Input the FSU report records.
2. For each "Hit" record "Before" and "After" pair reported in the FSU output, identify the DSN, PDSE(E) member name (if applicable) and record number at which the updated record may be found.
3. Sequentially read records from the data set or PDS(E) member until the required record number is found.
4. Verify that the input record matches the record "After" data reported in the FSU output.
5. Optionally UPDATE the data set record with the record "Before" data reported in the FSU output, thus restoring the record to its original status.
6. Generate an entry in the File Update Undo report.
7. Repeat all steps until all "Hit" record pairs in the FSU output report have been processed.

Error checking is also performed for conditions which include record not found or containing unexpected data. These conditions are reported in the FSUUNDO output report.

This SELCOPY job may be executed in the Foreground (TSO) or displayed as a JCL job, suitable for submission to batch. Using either method, options exist to generate an Expanded or Terse report, an optional Diagnostic SELCOPY execution report and, most importantly, options to Verify or Update changed records.

It is strongly recommended that users execute FSUUNDO with options VERIFY (the default) and EXTENDED, and then review the FSUUNDO output report before executing FSUUNDO again with option UPDATE.

Beware that record data that has been changed between the time of execution of the original File Search/Update/Copy/Remap job and this execution of FSUUNDO, will not be updated but will be reported as an error. Processing will continue with input of the next "Hit" record pair. Records that have already been restored as a result of a previous execution of FSUUNDO UPDATE, will be reported as a match and no error returned.

If SELCOPY ends with a return code other than 0 (zero - successful execution, no error conditions) or 112 (errors condition(s) detected), then re-run with option DIAGNOSE to establish the cause of the SELCOPY error.

The most likely cause of an unxpected return code will be if a selection (run) time error (RC=44) has occured. Usually caused by an OPEN error for an input data set (e.g. if an exclusive ENQ already exists for the data set.) In this event, SELCOPY processing ends immediately and all data sets opened by SELCOPY are automatically closed.

File Update Undo Panel

The File Update Utility Undo panel view may be started by executing command FSUUNDO from the command line of any window.

Field options may be selected or de-selected by entering a non-blank or blank character respectively in the option field.

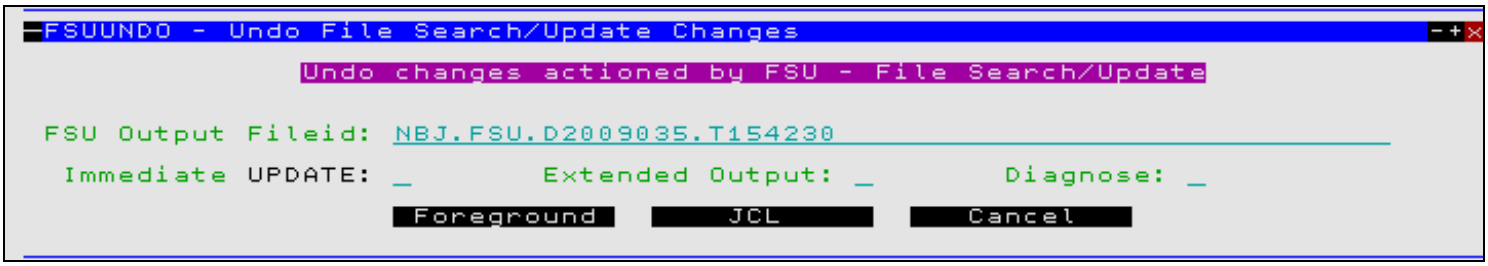


Figure 82. FSUUNDO - File Update Utility Undo Panel.

File Update Undo Output

Report Format

Report output is generated on every execution of the File Update Undo utility (FSUUNDO).

If FSUUNDO is executed with parameter BACKGROUND (JCL) to generate JCL output, then the output report is written to SYSPRINT when the job is submitted. By default, SYSPRINT is allocated to SYSOUT=*. Furthermore, if DIAGNOSE parameter was specified, then the SYSPRINT output will also contain diagnostic information for the SELCOPY run before and after the printed report output.

If FSUUNDO is executed with parameter FOREGROUND to execute in TSO, then an output report data set is opened in a CBLe edit view and report records are inserted. The DSN of this data set is equal to the FSU report DSN with the additional low level qualifier "UNDOV" for VERIFY reports, or "UNDO" for UPDATE reports. e.g.
`NBJ2.DEV.FSU.D2008346.T162607.UNDO`

If this data set already exists, then the report records will be appended to the existing report data and the edit display positioned at the start of the latest report output. On exit of this data set the user will be prompted to save and, if necessary allocate, the data set with suitable space attributes.

Report Fields

Dataset
The up to 44 character DSN of the data set or PDS/PDSE processed.

Member
For PDS/PDSE libraries only, the name of the member being processed.

RecordNumber
The record number at which an FSUUNDO error has occurred.
For EXTENDED output, this field also contains the record numbers of records that have been successfully updated.

Message Text
Message indicating success or failure to locate and update records referenced by the FSU report. All possible messages are as follow:

`= = File Updated = =`
`= = Member Updated = =`
One or more records within the reported data set or PDS/PDSE member were successfully updated.

For EXTENDED output only, this message is repeated for each record that has been successfully updated. Also, up to 100 bytes of the record data before and after the update is printed on the report lines that follow.

`## Member not found ##`
The member name within the PDS/PDSE library referenced by the FSU report line, no longer exists. The member has been deleted or renamed.
Return Code 112 is set and the error count incremented by one for each missing member.

`## Record not found ##`
: The record number of the data set or member record referenced by the FSU report line, no longer exists. This message is repeated for each missing record within a data set or member.

  For EXTENDED output only, up to 100 bytes of the expected record data is also printed on the report lines that follow.
  Return Code 112 is set and the error count is incremented by one for each missing record.

`### Data Mismatch ###  --WARNING--`
: The record data at the required record number within the data set or member, does not match the record data as it was following the FSU update. (i.e. the data has been altered since the FSU update was executed.) This message is repeated for each mis-matching record within a data set or member.

  For both EXTENDED and TERSE output, up to 100 bytes of the found record data, followed by up to 100 bytes of the expected record data, is also printed on the report lines that follow.
  Return Code 112 is set and the error count is incremented by one for each missing record.

`<<< Already Undone >>>`
: This message is generated for EXTENDED report output only. TERSE report output does not report instances where no record update is required.

  The record data at the required record number within the data set or member, matches the record data as it was prior to the FSU update. (i.e. the record has already been restored.)

  This message is repeated for each already restored record within a data set or member and the records already undone count is incremented by one. Up to 100 bytes of the record data is also printed on the following report lines.

Report Data

For EXPANDED output or where a data mismatch error has occurred, record data is reported in the FSUUNDO output.

The data lines are preceded by a brief description of the record. Where two contrasting records are to be displayed, the description line also reports the position within the record data of the first difference found. Furthermore, if this difference occurs beyond the record description and within the first 100 bytes, a marker ">*<" is positioned above the mis-matching character.

The record data follows the description line for a length equal to the lesser of the record length value and 100. The data is displayed in both character and vertical hexadecimal notation. (Equivalent to SELCOPY's TYPE=B print output.)

A scale line is written following the record data.

```
Record Found:  (1st Diff at Pos    049)        >*<
EDTFCLR0 0720 MEMFLSTE >>>>>>>>>>AZZElement     Test                          3 117ADA10 117BFAC8 00000..
CCECCDDF4FFFF4DCDCDEEC46666666666CEEC98989A44444E8AA444444444444444444444444F4FFFCCCFF4FFFCCCCF4FFFFF
5436339000720045463235OEEEEEEEEE19953545530000035230000000000000000000000030117141100117261380000000

Record Expected:
EDTFCLR0 0720 MEMFLSTE >>>>>>>>>>AZZElement                                   3 117ADA10 117BFAC8 00000..
CCECCDDF4FFFF4DCDCDEEC46666666666CEEC98989A444444444444444444444444444444F4FFFCCCFF4FFFCCCCF4FFFFF
5436339000720045463235OEEEEEEEEE19953545530000000000000000000000000000000030117141100117261380000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8....,....9....,...10
```

Figure 82. Record Data Output.

Summary Block

The summary block reports the totals of:

1. Records undone
2. Records already undone
3. Files updated
4. Errors

Where FSUUNDO errors have occured, the Errors total has the additional eye-catcher "###" following.

Sample Terse Report Output

```
------------------------------------ Start of FSUUNDO Report ----------------------------------------


  Output From:  FSUUNDO  NBJ2.DEV.FSU.D2008357.T132545  FOREGROUND  VERIFY  TERSE


  ** Verify Only - No data sets will be updated. **



------------------------------------------- -------- ------------         ----------------   --------
Dataset                                     Member   RecordNumber         2008/12/22 13:27   PAGE   1
------------------------------------------- -------- ------------         ----------------   --------
NBJ.JCL.FSU                                 CBLINST                 = = Member Updated = =
NBJ.JCL.FSU                                 CBLINS01                = = Member Updated = =
NBJ.JCL.FSU                                 CBLINS01           188  ## Record not found ##
NBJ.JCL.FSU                                 EQAWIVP1                = = Member Updated = =
NBJ.JCL.FSU                                 IEBCOPY1            51   ### Data Mismatch ###  --WARNING--

Record Found:  (1st Diff at Pos    046)     >*<
//GETMEMS   EXEC PGM=IEBCOPY,REGION=0M       ## Get ###
66CCEDCDE444CECC4DCD7CCCCDDE6DCCCDD7FD4444444774C8A4777444444444444444444444444444
11753454200057530774E9523678B957965E040000000BB07530BBB00000000000000000000000000

Record Expected:
//GETMEMS   EXEC PGM=IEBCOPY,REGION=0M
66CCEDCDE444CECC4DCD7CCCCDDE6DCCCDD7FD444444444444444444444444444444444444444444444
11753454200057530774E9523678B957965E040000000000000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8

NBJ.JCL.FSU                                 IEBCOP02                = = Member Updated = =

     ---------------------------------------
     Records undone:                    4
     Records already undone:            0
     Files   updated:                   5
     Errors:                            2###
     ---------------------------------------

  ** Verify Only - No data sets have been updated. **


  *** Warning:  SELCOPY execution ended with RC=112 ***

  One or more data sets have been altered between the time of execution of
  the original File Search/Update job and this execution of FSUUNDO.
```

Figure 82. FSUNDO TERSE Output with VERIFY.

Sample EXTENDED Output

```
------------------------------------ Start of FSUUNDO Report ---------------------------------------


  Output From:  FSUUNDO  NBJ2.DEV.FSU.D2008357.T132545  FOREGROUND  UPDATE  EXTENDED



------------------------------------------- -------- ------------         ----------------   --------
Dataset                                     Member   RecordNumber         2008/12/22 13:35   PAGE   1
------------------------------------------- -------- ------------         ----------------   --------
NBJ.JCL.FSU                                 CBLINST           42  = = Member Updated = =

Record Old:  (Changed data starts at Pos    037)
//LOAD      EXEC PGM=IEBCOPY,REGION=4M
66DDCC444444CECC4DCD7CCCCDDE6DCCCDD7FD44444444444444444444444444444444444444444444
113614000000057530774E9523678B957965E44000000000000000000000000000000000000000000

Record New:
//LOAD      EXEC PGM=IEBCOPY,REGION=0M
66DDCC444444CECC4DCD7CCCCDDE6DCCCDD7FD44444444444444444444444444444444444444444444
113614000000057530774E9523678B957965E04000000000000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8



------------------------------------------- -------- ------------         ----------------   --------
Dataset                                     Member   RecordNumber         2008/12/22 13:35   PAGE   1
------------------------------------------- -------- ------------         ----------------   --------
NBJ.JCL.FSU                                 CBLINS01         119  = = Member Updated = =

Record Old:  (Changed data starts at Pos    037)
//LOAD      EXEC PGM=IEBCOPY,REGION=4M
66DDCC444444CECC4DCD7CCCCDDE6DCCCDD7FD44444444444444444444444444444444444444444444
113614000000057530774E9523678B957965E44000000000000000000000000000000000000000000

Record New:
//LOAD      EXEC PGM=IEBCOPY,REGION=0M
66DDCC444444CECC4DCD7CCCCDDE6DCCCDD7FD44444444444444444444444444444444444444444444
113614000000057530774E9523678B957965E04000000000000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8

NBJ.JCL.FSU                                 CBLINS01         188  ## Record not found ##

Record Expected:
//GETMEMS   EXEC PGM=IEBCOPY,REGION=4M
66CCEDCDE444CECC4DCD7CCCCDDE6DCCCDD7FD44444444444444444444444444444444444444444444
117534542000057530774E9523678B957965E44000000000000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8



------------------------------------------- -------- ------------         ----------------   --------
Dataset                                     Member   RecordNumber         2008/12/22 13:35   PAGE   1
------------------------------------------- -------- ------------         ----------------   --------
NBJ.JCL.FSU                                 EQAWIVP1         124  = = Member Updated = =

Record Old:  (Changed data starts at Pos    046)
//COMPILE EXEC PGM=IGYCRCTL,PARM=(''),REGION=4M
66CDDDCDC4CECC4DCD7CCECDCED6DCDD747756DCCCDD7FD44444444444444444444444444444444444
113647935057530774E97839333B7194EDDDDB957965E44000000000000000000000000000000000

Record New:
//COMPILE EXEC PGM=IGYCRCTL,PARM=(''),REGION=0M
66CDDDCDC4CECC4DCD7CCECDCED6DCDD747756DCCCDD7FD44444444444444444444444444444444444
113647935057530774E97839333B7194EDDDDB957965E04000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8

NBJ.JCL.FSU                                 EQAWIVP1         203  = = Member Updated = =

Record Old:  (Changed data starts at Pos    043)
//LKED    EXEC PGM=HEWL,PARM='MAP',REGION=4M
66DDCC4444CECC4DCD7CCED6DCDD77DCD76DCCCDD7FD44444444444444444444444444444444444444
113254000057530774E8563B7194ED417DB957965E44000000000000000000000000000000000000

Record New:
//LKED    EXEC PGM=HEWL,PARM='MAP',REGION=0M
66DDCC4444CECC4DCD7CCED6DCDD77DCD76DCCCDD7FD44444444444444444444444444444444444444
113254000057530774E8563B7194ED417DB957965E04000000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8



------------------------------------------- -------- ------------         ----------------   --------
Dataset                                     Member   RecordNumber         2008/12/22 13:35   PAGE   1
------------------------------------------- -------- ------------         ----------------   --------
NBJ.JCL.FSU                                 IEBCOPY1          51   ### Data Mismatch ###  --WARNING--
```

```
Record Found:  (1st Diff at Pos    046)     >*<
//GETMEMS   EXEC PGM=IEBCOPY,REGION=4M       ## Get ###
66CCEDCDE444CECC4DCD7CCCCDDE6DCCCDD7FD4444444774C8A477744444444444444444444444444
11753454200057530774E9523678B957965E440000000BB07530BBB00000000000000000000000000

Record Expected:
//GETMEMS   EXEC PGM=IEBCOPY,REGION=4M
66CCEDCDE444CECC4DCD7CCCCDDE6DCCCDD7FD444444444444444444444444444444444444444444
11753454200057530774E9523678B957965E440000000000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8



---------------------------------------- -------- ------------         ----------------   --------
Dataset                                   Member   RecordNumber         2008/12/22 13:35   PAGE   1
---------------------------------------- -------- ------------         ----------------   --------
NBJ.JCL.FSU                               IEBCOP02           7  = = Member Updated = =

Record Old:  (Changed data starts at Pos    037)
//UNLOAD    EXEC PGM=IEBCOPY,REGION=4M
66EDDDCC4444CECC4DCD7CCCCDDE6DCCCDD7FD444444444444444444444444444444444444444444
11453614000057530774E9523678B957965E440000000000000000000000000000000000000000

Record New:
//UNLOAD    EXEC PGM=IEBCOPY,REGION=0M
66EDDDCC4444CECC4DCD7CCCCDDE6DCCCDD7FD444444444444444444444444444444444444444444
11453614000057530774E9523678B957965E040000000000000000000000000000000000000000
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8


     ---------------------------------------
     Records undone:                     5
     Records already undone:             0
     Files   updated:                    4
     Errors:                             2###
     ---------------------------------------


  *** Warning:  SELCOPY execution ended with RC=112 ***

  One or more data sets have been altered between the time of execution of
  the original File Search/Update job and this execution of FSUUNDO.
```

Figure 82. FSUNDO EXTENDED Output with UPDATE.

Compare Files/Libraries Menu (=7)

The Compare Files/Libraries Menu panel (ZZSGCOMP) is an interactive panel window opened on selection of option 7. in the SELCOPY/i Primary option menu.

SELCOPY/i data object compare features can be accessed via this panel.

Options

1 Compare Files	**COMPF**	Compare Files
2 Compare Libraries	**COMPL**	Compare Libraries

Compare Files (=7.1)

Overview

The Compare Files utility (COMPFILE) provides a set of both basic and extended features that allow the user to compare records in **NEW** and **OLD** versions of a file.

Basic features include:

- Specify the start record.
- Restrict the number of records compared.
- Restrict the number of differences to be reported.
- Restrict the comparison to a specific area of the file records.
- Strip trailing characters prior to record compare.

Extended features include:

- All basic feature options but with separate specifications for the NEW and OLD files where sensible.
- Apply a **structure** (copybook) overlay to map records, and optionally restrict the comparison to specified record-types and/or named fields. This is known as a **formatted compare**.
- Control how re-synchronisation of record pairs should occur following detection of an inserted or deleted record.
- For formatted or unformatted compare, specify **key** segments (at the record-type level) that allow the utility to identify **synchronised pairs** of records.
- Formatted compare supports application of different structures to the NEW and OLD files, with comparison restricted to only those fields that exist in both structures. This allows comparison of NEW and OLD file records where corresponding fields are at different locations within the records and maybe of different data-type or length.

Following Compare Files online execution, by default report output is generated in a structured format suitable for presentation to the user in a Data Edit window view. For batch execution the default is to produce a formatted text report.

Source File Types

COMPFILE can process records from any combination of the following file types:

- Cataloged or uncataloged sequential (including multi-volume) datasets.
- Partitioned dataset (PDS/PDSE) members.
- Partitioned PDSE version 2 dataset member generations.
- GDG datasets.
- VSAM (KSDS, ESDS, RRDS, VRDS).
- Unix (HFS/ZFS/NFS) Files.
- DB2 Result Tables

Output Report

The report generated by the compare files utility is available in two formats controlled by the LIST=FMT|TEXT option.

LIST=FMT is the default when **COMPFILE** is executed within the **SELCOPYi online environment**. Using this option the report generated is a **structured data file**. This is designed to be (automatically) browsed (not printed) from within a SELCOPYi online session, using a structure definition (SDO) which is also generated automatically during execution of the compare.

LIST=TEXT is the default when **COMPFILE** is executed from JCL as a batch job. Using this option the report generated is a more traditional formatted text document, designed to be printed if necessary.

Following execution of the compare utility, records are flagged as being **matched** or as having been **inserted**, **deleted** or **changed**.

Matched
Records that exist in both the NEW and OLD files forming a *synchronised record pair* for which the compared data is unchanged (matches).

Inserted
Records that previously did not exist in the old file and so are considered to have been inserted into the NEW file.

Deleted
Records that no longer exist in the NEW file and so are considered to have been deleted from the OLD file.

Changed
Records that exist in both files forming a *synchronised record pair* in which the compared data has been changed (i.e does not match).

Determination of synchronised record pairs is achieved by the compare file utility using record synchronisation techniques.

In order to improve readability, the report of consecutive records flagged as having been deleted are grouped together, and similarly for records flagged as having been inserted.

See Compare Files Output for a detailed description of the generated output report.

Unformatted Compare

Unformatted compare is the most commonly used format for **text** files containing unformatted records.

By definition, unformatted compare operates on records without application of a structure (SDO) or COBOL/PL1 copybook to format record data. i.e. each record is treated as a single character string.

Basic Unformatted Compare
Basic unformatted compare specifically relates to unformatted compare where selected options apply to **both** files involved in the compare operation. These options are:

- ◊ The compare data start position within the record.
- ◊ The compare data length.
- ◊ The trailing character to be stripped before comparing the data.
- ◊ The first record to be compared. (Nominated by record number, key or RBA.)
- ◊ The number of records to be compared.

Furthermore, record synchronisation technique employed is restricted to 1-TO-1 or read-ahead with a read-ahead limit of 100 records and read-ahead matching record count of 1.

The Compare File utility panel view and input field options relating to basic unformatted compare are described in *"Basic Unformatted Compare Panel"*.

Extended Unformatted Compare
Extended unformatted compare allows specification of the same options as basic unformatted compare but with potentially different values for each of the two files in the compare operation. In addition to this, extended unformatted compare allows specification of the following:

- ◊ Record synchronisation techniques Sorted Keyed and Unsorted Keyed which involves specification of key segments.
- ◊ For read-ahead record synchronisation, non-default values for limit and matching record count. Also the option to allow synchronisation on blank records.
- ◊ The option to perform case-insensitive compare.
- ◊ Report output options to exclude display of changed, inserted and/or deleted records. Also allows specification of a non-default report file DSN.
- ◊ Output file DSNs into which to copy records flagged as being matched, changed, inserted and/or deleted. A separate data set name may be specified for NEW and OLD file records that are attributed these flags.

The Compare File utility panel views and input field options relating to extended unformatted compare are described in *"Extended Unformatted Compare Panel"*.

Formatted Compare

More advanced than unformatted compare, formatted compare is invoked where an SDE structure (SDO), COBOL or PL1 copybook overlay is specified to map record data fields for use in the compare files operation.

Records are treated as comprising a number of data fields of pre-determined lengths and of various data types. Each field within the record may be referenced independently (by field name or field reference number) allowing the user to be more descriminate when selecting records, and fields to be compared.

If a COBOL copybook, PL1 include file or an ADATA file generated from a COBOL or PL1 compilation is specified, then this file will be used to generate a temporary SDO before proceeding with record formatting. Note that a non-temporary SDO may be generated from the COBOL/PL1/ADATA file using the SDE command, CREATE STRUCTURE.

Each input record is assigned a record type (RTO) defined in the specified or generated SDO and the field definitions defined by that RTO are used to map the data within the record. SDE determines the record type to be assigned to each record based on any USE WHEN conditions saved in the SDO and the individual record's length. See *"Record Type Assignment"* in the *"SELCOPYi Structured Data Editor (SDE)"* publication.

Formatted compare may be selected via the Compare File utility panel by first selecting **Extended options** from the Compare Files Basic Options view. Compare File utility panel views and input field options relating to formatted compare are described in *"Formatted Compare Panel"*.

Hierarchical Compare

Hierarchical compare is not selected explicitly but is implied when both of the following conditions are true:

1. Formatted compare is used incorporating records assigned to different record types in the SDO.
2. KEY synchronisation is performed with key segments specified as formatted record field names or field reference numbers.

The compare files command, COMPFILE, generated by the dialog panel or entered manually by the user, specifies synchronisation key fields for one or more record types in the specified SDO. The order in which these record types occur in the COMPFILE command also define the levels of record type hierarchy. i.e. The record type synchronisation key definition occurring first identifies the level-1 (highest level) record type, the second definition identifies the level-2 (level-1 child) record type, etc.

Record types with no synchronisation key are the lowest level in the record type hierarchy, i.e. rated lower than any record type that has been defined with a synchronisation key.

Hierarchical compare is sensitive to the level of record type assigned to a record. All records that immediately follow the current record which are assigned record types lower in the record type hierarchy than that of the current record, are treated as being descendants of the current record. These records are grouped with the current record so that record synchronisation does not exceed the bounds of the current hierarchical record group.

This type of compare ensures that only record pairs that belong to the same hierarchical parent record pair can be synchronised.

For details on the synchronisation criteria and the synchronisation process, see *"Hierarchical Key Synchronisation"*.

Record Synchronisation

The process of comparing records requires that a pair of records, one from NEW file and one from the OLD file, are first synchronised before the data compare can take place.

Terminology

Before learning the different methods of synchronisation, a user should become familiar with the following terms:

Record Pair
Refers to one record from the NEW file and one from the OLD file.

Synchronised Record Pair
A record pair for which the records satisfy synchronisation criteria and so are eligible to be compared.

See the different record synchronisation techniques which identify synchronisation criteria.

Current Record Pair
Identifies the record pair for which synchronisation will occur.

The current record pair is usually the next record pair read following processing of a synchronised record pair. Unless End-of-Data is encountered, re-synchronisation processing will invariably change one of the records selected for the current record pair.

Compare Data
The compare data is identified as belonging to those areas of the record for which the compare operation will be actioned. Record data that falls outside these areas is not compared.

Unformatted compare data areas are specified via the Compare File panel "`Compare Position/Length:`" fields or the COMPFILE command STARTCOL/COMPARELEN parameters (or their derivatives).
Formatted and Hierarchical compare data areas are specified via the Compare File sub-panel *"Select Field Names to Compare"* or the COMPFILE command SELECT *field_col* parameter.

By default, compare data is all data in the record.

Keyed Record

Applicable only to file compare with KEY synchronisation, a keyed record is a record for which one or more compare files synchronisation key field segments have been defined.

For unformatted or formatted compare where key segments are specified using fixed field positions and lengths, this applies to all records in the NEW and OLD files.

For formatted compare where key segments are specified using formatted record field names, this applies only to records in the NEW and OLD files that are assigned record types for which a key has been defined.

Unkeyed Record

Applicable to compare of formatted or unformatted files, unkeyed records are records not defined a compare files synchronisation key field.

For hierarchical compare, an unkeyed record is a record assigned a record type for which no synchronisation key has been defined. Unkeyed records are of record types which are lowest level in the record type hierarchy.

Overview

Following input of a record pair, the compare files utility must use the rules defined by the specified synchronisation technique to determine whether the current record pair is a synchronised record pair.

If the current record pair is not a synchronised record pair, then record synchronisation is performed for one or both of the records in the current record pair. With the exception of 1-TO-1 synchronisation, this involves reading additional records from the NEW and/or OLD files in order to synchronise with a record in the current record pair.

If a synchronised record pair is established, then the compare of record data is performed so that the records in the record pair are flagged as being either **matched** or **changed**. Following the compare, the next record pair is read sequentially from the NEW and OLD files and the synchronisation process begins again.

Records in the NEW file that are been skipped as a result of record synchronisation are flagged as having been **inserted**. Similarly, skipped records in the OLD file are flagged as having been **deleted**.

The possible record synchronisation techniques are described below.

1-TO-1 Synchronisation

All current record pairs are synchronised record pairs so no attempt will ever be made to synchronise records. Mismatching record pairs are flagged as a record change and only additional records in the NEW or OLD files will be flagged as being inserted or deleted respectively.

This also applies to formatted compare so even records of different record types are considered to be a synchronised record pair.

This technique corresponds to the COMPFILE command parameter `SYNC 1TO1`.

Read-Ahead Synchronisation

Read-Ahead synchronisation is suitable where the NEW and OLD files are predominantly comprised of equal records, although some may have been updated, inserted and/or deleted.

Read-ahead synchronisation criteria requires that the compare data within the records that constitute the current record pair must match.
Note that, for formatted compare, read-ahead synchronisation will consider only record pairs of the same record type as being potential synchronised record pairs.

If the current record pair is not a synchronised record pair, then records are read sequentially from one or both files in order to identify a record that constitutes a synchronised record pair with one of the records in the current record pair.

Read-ahead synchronisation for unformatted or formatted compare proceeds as follows:

1. From the OLD file, a specified number of records are read as defined by the read-ahead limit (default 100) until one is found that matches the NEW file record in the current record pair. If found, this record becomes the OLD file record in the current record pair.

   The read-ahead match value (default 1) identifies the total number of consecutive matching record pairs that must exist, starting at the current record pair, before the current record pair can be considered a potential synchronised record pair.

   By default, matching pairs of blank records are **not** included in the read-ahead match count of consecutive matching record pairs. This avoids erroneously synchronising on blank records within inserted or deleted blocks of records. If blank records are to be included in the read-ahead synchronisation, then this may be set in the compare files panels or by specifying parameter SYNCONBLANK on the COMPFILE line command.

   If the read-ahead number of consecutive matching pairs is not satisfied, then the read-ahead synchronisation process continues for the records in the OLD file until End-of-Data or the read-ahead limit is reached.

2. Employing the same conditions, the read-ahead process is repeated for records in the NEW file in order to identify a potential synchronisation record pair with the OLD file record in the original current record pair.

3. If only one possible synchronised record pair is identified, then this becomes the new synchronised record pair.

   If read-ahead in the OLD and NEW files identify different potential synchronised record pairs, then synchronisation will occur at the record pair with the fewer number of records between it and the current record pair. If this number of intervening records is equal for both record pairs, then synchronisation occurs for the synchronised record pair found in the read-ahead of the NEW file.

4. If no potential synchronised record pairs are identified, then the current record pair constitutes a record mismatch corresponding to a NEW file record insert and an OLD file record delete. The compare operation continues at the next record pair read sequentially from both files.

This synchronisation technique corresponds to the COMPFILE command parameter `SYNC READAHEAD(ralimit ramatch)`.

Key Synchronisation

Key synchronisation is suitable where an exact match on only data in specified key segments within the records of a record pair is necessary to identify it as a synchronised record pair. The data contained in the key segments need not be unique and records containing these key fields need not have been sorted into ascending order based on data in the key segments.

If the files contain records of different record type having synchronisation key segments that are not common to all record types, then Hierarchical Key Synchronisation must be used.

Key segments in these records are collectively known as the synchronisation key and are defined when invoking the compare files facility. Segments of the synchronisation key need not necessarily be included within the compare data.

For unformatted compare, the synchronisation key segments must be specified as key field positions and lengths.

For formatted compare that involves records of different record types but with each record containing the same synchronisation key (e.g. KSDS records), the synchronisation key segments should be specified as a key field positions and lengths. In doing so, the compare files utility will ignore the applied record structure when performing key synchronisation.

Formatted compare involving only records of the same record type may specify the synchronisation key segments using the formatted record field names or field references.
Formatted compare involving records of different record types with synchronisation key segments specified using formatted record field names or field references forces hierarchical key synchronisation.

Key Synchronisation criteria requires that data must match in all key segments defined within the records comprising the current record pair.

Unsorted Key Synchronisation

Where the keyed records **are not** sorted by the synchronisation key in ascending order, unsorted key synchronisation should be used.

Unsorted key synchronisation employs a read-ahead policy to synchronise record pairs, including use of a read-ahead limit and read-ahead matching record count.

Unsorted key synchronisation criteria is similar to read-ahead synchronisation except that the areas of matching data within records comprising the current record pair, are defined by the synchronisation key as opposed to the compare data.

Unsorted Key synchronisation proceeds as described for read-ahead synchronisation above.

This synchronisation technique corresponds to the COMPFILE command parameters `SYNC UNSORTED KEY READAHEAD(ralimit ramatch)`.

Sorted Key Synchronisation

Where the keyed records **are** sorted by the synchronisation key in ascending order, sorted key synchronisation should be used.

Sorted Key synchronisation proceeds as follows:

1. The synchronisation key segments of the current record pair are checked to determine whether synchronisation criteria are satisfied. If so, the compare data in this synchronised record pair is compared and flagged as being matched or having been changed as appropriate.

2. If not a synchronised pair, then records that follow the current record pair are read sequentially from the file containing the record with the lower synchronisation key value. Reading stops when End-of-Data is encountered or a record is found that has synchronisation key data that matches, or is greater than, the synchronisation key data of the other record in the current record pair.

3. If End-of-Data is encountered then all records in both files, starting at the records in the current record pair up to the End-of-Data, are flagged as being inserts or deletes as appropriate.

4. If a record with a matching or greater synchronisation key is found then this record becomes the new record in the current record pair and all records between it and the original current record are flagged as having been

either inserted or deleted as appropriate.
Note that the record in the original current record pair with the higher synchronisation key, remains in the current record pair.

The synchronisation process is repeated for the new current record pair.

This synchronisation technique corresponds to the COMPFILE command parameters `SYNC KEY`.

Hierarchical Key Synchronisation

Hierarchical key synchronisation is performed automatically for Hierarchical Compare.

Hierarchical key synchronisation is suitable where a hierarchy exists between formatted records or record segments of different record types in the same file. e.g. A file may be arranged in a hierarchy of records or record segments detailing ORDERS, ORDER ITEMS and ITEM PARTS so that ORDERS base record segments are followed by a number of ORDER ITEMS record segments followed by a number of ITEM PARTS record segments.

Note that, if no hierarchical relationship exists between formatted records assigned different keyed record types within the same file, then a separate non-hierarchical compare files operation should be performed instead, one each for records assigned the same record type.

The record hierarchy is maintained by key fields defined to at least one of the record types. Key fields in these records are collectively known as the synchronisation key and are defined when invoking the compare files facility. Fields defined in the synchronisation key need not necessarily be included within the compare data.

The record type synchronisation key hierarchy is established by the order in which synchronisation keys are specified for each record type. The first record type to be defined a synchronisation key is attributed the highest level (level-1) entry in the hierarchy, the next key definition is attributed the level-2 entry, etc. Record types with no defined synchronisation key are equally attributed the lowest level entry in the synchronisation key hierarchy.

Data contained in the synchronisation key need not be unique and records containing a synchronisation key do not have to be sorted into ascending order based on data in the synchronisation key fields.

Hierarchical Key Synchronisation criteria requires that the following conditions are true in order identify the current record pair as a synchronised record pair:

- Keyed or unkeyed records that comprise the current record pair must be of the same record type.
- Keyed records that comprise the current record pair must have matching data in all synchronisation key fields.
- Unkeyed records that comprise the current record pair must have matching compare data and satisfy the read-ahead matching record count. (True for both sorted and unsorted hierarchical key synchronisation.)

The defining feature of both unsorted and sorted hierarchical key synchronisation processing, is that input of records from the NEW and/or OLD files stops when a keyed record is read which is rated higher in the synchronisation key hierarchy than the record being synchronised in the current record pair.
This ensures that records cannot be synchronised with records of the same record type but belonging to a different parent record type.

This rule applies equally to records comprising the current record pair. i.e. synchronisation will not be attempted for a record within the current record pair if that record is rated lower in the synchronisation key hierarchy than the other record in the current record pair.

Unsorted Hierarchical Key Synchronisation

Unsorted hierarchical key synchronisation processing is the same as for unsorted key synchronisation of formatted records but with the following additional conditions:

1. The read ahead of records from the OLD and NEW files is restricted, not only by the defined read-ahead limit (keyed records only) and End-of-Data conditions, but also by input of a record which is rated higher in the synchronisation key hierarchy than that of the record being synchronised.

2. On attempting synchronisation of a **keyed** record, the read-ahead matching record count value is ignored. This is because a matching keyed record pair may often be followed by different child record types.

3. Synchronisation of **unkeyed** records is described in *"Hierarchical Key Synchronisation of Unkeyed Records"* below.

4. If no potential synchronised record pairs are identified, then additional processing occurs as described in *"Unsynchronised Hierarchical Record Pair"* below.

This synchronisation technique corresponds to the COMPFILE command parameter `SYNC UNSORTED KEY READAHEAD(`*`ralimit ramatch`*`)`.

Sorted Hierarchical Key Synchronisation

Sorted hierarchical key synchronisation may only be used if **all** keyed records are sorted in ascending order by their synchronisation key data.

The sorted hierarchical key synchronisation processing is the same as for sorted key synchronisation of formatted records but with the following additional conditions:

1. If the current record pair is comprised of records assigned record types of different levels in the synchronisation key hierarchy, then synchronisation occurs for the record with the key rated higher in the synchronisation key hierarchy.

   The record with the lower rated key and all records up to the next synchronised record pair will be flagged as being inserted or deleted as appropriate.

2. Input of records from the file with the lower synchronisation key is restricted, not only by the End-of-Data condition, but also by input of a record which is rated higher in the synchronisation key hierarchy than that of the record being synchronised.

3. Synchronisation of **unkeyed** records performs read-ahead synchronisation as described in *"Hierarchical Key Synchronisation of Unkeyed Records"* below.

4. If no potential synchronised record pairs are identified, then additional processing occurs as described in *"Unsynchronised Hierarchical Record Pair"* below.

This synchronisation technique corresponds to the COMPFILE command parameters `SYNC KEY`.

Hierarchical Key Synchronisation of Unkeyed Records

Hierarchical compare (both sorted or unsorted key synchronisation) supports records tha are assigned a record type defined without a synchronisation key. For key synchronisation purposes, these unkeyed records are considered to be lower in the synchronisation key hierarchy than any keyed record.

Synchronisation processing for unkeyed record pairs is identical for both sorted and unsorted hierarchical key synchronisation and will only be attempted if both records of the current record pair are unkeyed. This is because synchronisation of an unkeyed record is not attempted if the other record in the current record pair is keyed.

Whether or not records comprising the unkeyed record pair is assigned the same record type, standard read-ahead synchronisation processing is performed with the following exceptions:

1. The read-ahead limit is ignored for both sorted and unsorted hierarchical synchronisation. The records read will be limited by the End-of-Data condition and input of a keyed record.

2. Unlike standard sorted key synchronisation, sorted hierarchical key synchronisation supports a read-ahead matching record count in order to comply with read-ahead synchronisation of unkeyed records.

Unsynchronised Hierarchical Record Pair

For **sorted** hierarchical key synchronisation only, encountering End-of-Data before a synchronised record pair can be established will simply flag the records in the current record pair up to the End-of-Data for both files, as having been inserted or deleted as appropriate.

If, for any other reason, sorted or unsorted hierarchical key synchronisation identifies no potential synchronised record pairings for records in the current record pair, then different processing occurs depending on the records that comprise the current record pair:

Unkeyed Record Pair

If both records are unkeyed then this constitutes a record mismatch corresponding to a NEW file record insert and an OLD file record delete. The compare operation continues at the next record pair read sequentially from both files.

Keyed Record - Record Type Different

If at least one of the records is keyed and the records are of **different** record types, then processing proceeds as follows:

1. Records are read from the **same** file as the record assigned a record type with the lower level synchronisation key until one of the following is encountered.

   - The End-of-Data condition.
   - For **unsorted** hierarchical key synchronisation only, the read-ahead limit.
   - A record of the same record type or one which is rated higher in the synchronisation key hierarchy.

2. If End-of-Data is encountered then all records in both files, starting at the records in the current record pair up to the End-of-Data, are flagged as having been inserted or deleted as appropriate.

3. For **unsorted** hierarchical key synchronisation only, if the read-ahead limit is reached then the compare file operation terminates with with error ZZSD410E.

4. If a record of the same record type or one with a higher level synchronisation key is found then this record becomes the new record in the current record pair. All records between this record and the original current record are flagged as having been either inserted or deleted as appropriate.
   Note that the record in the current record pair with the higher level key, remains in the current record pair.

   If the new current record pair is not a synchronised record pair, then hierarchical key syncronisation processing occurs.

Keyed Record - Record Type Same

If the records are of the **same keyed** record type, then processing proceeds slightly differently for sorted and unsorted hierarchical key synchronisation.

For **unsorted** hierarchical key synchronisation:

1. Records are read from **both** files until either End-of-Data or the read-ahead limit is encountered, or a record of the same record type or one with a higher level synchronisation key is read.
2. If End-of-Data is encountered in **either** of the files, then all records in both files, starting at the records in the current record pair up to the End-of-Data, are flagged as having been inserted or deleted as appropriate.
3. If the read-ahead limit is reached then the compare file operation terminates with with error ZZSD410E.
4. If a record of the same record type or one with a higher level synchronisation key is found in both files then these records become the new current record pair. All records between these records and the records in the original current record pair are flagged as having been either inserted or deleted as appropriate.

   If the new current record pair is not a synchronised record pair, then unsorted hierarchical key syncronisation processing occurs.

For **sorted** hierarchical key synchronisation:

1. Records are read from the file with the higher level synchronisation key until either End-of-Data is encountered or a record is read that is assigned a record type with a higher level synchronisation key.
2. If End-of-Data is encountered, then all records in both files, starting at the records in the current record pair up to the End-of-Data, are flagged as having been inserted or deleted as appropriate.
3. If a record with a higher level synchronisation key is found then this record becomes the new record in the current record pair. All records between this record and the original record in the current record pair are flagged as having been either inserted or deleted as appropriate.
   Note that the record in the current record pair with the lower level synchronisation key, remains a record of the current record pair.

   The new current record pair is not a synchronised record pair so sorted key syncronisation processing occurs.

Compare Files Panels

See Compare DB2 Tables for a description of the panels specific to the compare of 2 DB2 database result tables.

The Compare Files utility panel views (ZZSGCFO0) and their sub-panels are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select 'Compare Files' from the Utilities menu.
- Execute the command COMPFile with no parameters from the command line of any window.
- Execute the prefix command "**CF**" from a file List type window. The resulting Compare Files panel window will treat the corresponding list entry as the NEW file.

By default, field entries are populated with arguments and options that were entered the last time the Compare Files Utility panels were used.

Most field entries are optional and need to be activated by entering "/" in the preceding field.

Basic Unformatted Compare Panel

Compare Files: Basic Options

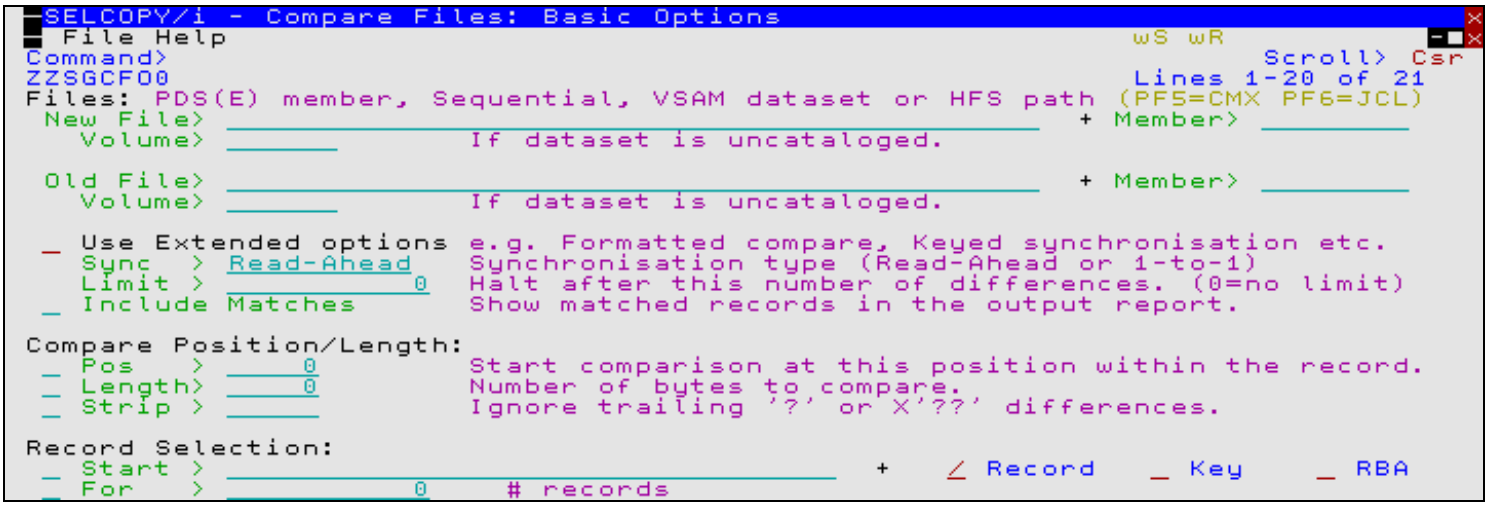


Figure 83. SELCOPY/i - Compare Files: Basic Options.

The **Compare Files: Basic Options** panel view is the first displayed when the Compare Files utility is started interactively.

It is anticipated that most file compare requirements will be satisfied by this panel view. Therefore, having typed entries in the required panel fields, simply pressing the <Enter> key or, if configured, double-clicking the left mouse button will will action the file compare in the foreground.

Alternatively, the user can:

- Press **PF5** to generate COMPFILE command line syntax.
- Press **PF6** to generate a COMPFILE batch job.

If this panel view does not satisfy the user's compare file requisites, then the Extended Options option field should be selected.

Features of basic unformatted compare are described in *"Basic Unformatted Compare"*.

New/Old File>
Member>
Volume>

Identifies the NEW and OLD files to be compared. Dataset names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

Sync>

Select either **Default** or **1-to-1**.

If "Default" is chosen and both files are VSAM KSDS with equal defined key lengths, then keyed synhronisation is employed, with the key locations automatically assigned from the KSDS file definitions. For all other file combinations, "Default" employs **Read-Ahead** synhronisation.

The **Compare Files: Basic Options** panel only offers these record synchronisation techniques. For a greater choice (including Unsorted Key and Sorted Key synchronisation), select the Use Extended Options option.

A default employing **Read-Ahead** uses default read ahead limit value of 100 records, matching record count of 1 record and bypasses synchronisation on blank records.

This field corresponds to COMPFILE parameter SYNC.

Limit>

Use this option in order terminate the compare process as soon as the specified number of record mismatches has been encountered.

Specifying zero or blank indicates that no limit is placed, and therefore the whole of each file (or record selection range) is processed.

This field corresponds to the COMPFILE parameter LIMIT.
Default is 0 (no limit).

Context>
A selection list will be displayed if the option entered is unrecognised or left blank. Choose from one of the following options:

Partial
Show 10 lines before and after each difference in order to provide additional context without displaying all matching records. Each difference "block" will be followed by 2 "gap" records designed to provide visual separation.

This option corresponds to the COMPFILE parameters "CONTEXT 10" and "GAP 2".

For full control over the number of context and gap records select Use Extended Options below.

Full
All corresponding records from both NEW and OLD files that match are to be included in the output report file.

This option corresponds to the COMPFILE parameter "INCMATCHED".

None
No matching records are to be displayed.

List>
A selection list will be displayed if the option entered is unrecognised or left blank. Choose from one of the following options:

FMT
The report generated is a **structured data file** designed to be (automatically) browsed (not printed) from within a SELCOPYi online session, using a structure definition (SDO) which is also generated automatically during execution of the compare.

TEXT
The report generated is a more traditional formatted plain text document, designed to be printed if necessary.

Default
FMT if running online, **TEXT** if running from JCL in batch.

Pos>
Record data from both the NEW and OLD files will be compared from this position onwards, for the specified length, or to the end of the record if no length is specified.

This field corresponds to the COMPFILE parameter STARTCOL.
Default is 1.

Length>
Record data from both the NEW and OLD files will be compared for this length.

This field corresponds to the COMPFILE parameter COMPARELEN.
Default is the length from the start position, specified in the **Pos>** field, to the end of the record.

Strip>
Record data from both the NEW and OLD files will have any trailing occurrences of the specified character stripped before comparison is actioned.

This option is particularly useful when **fixed length records** of different length are being compared, or when fixed length records are being compared with **variable length records**.

The single character may be specified as a literal **x** or **'x'**, which will be upper cased before stripping occurs, character string **C'x'** (character case preserved) or a hexadecimal string **X'nn'**.

This field corresponds to the COMPFILE parameter STRIP.
Default is blank (X'40').

Start>
Defines the start record for comparison in both the NEW and OLD files.

User should enter a record number, an RBA number (ESDS only), or a key string (KSDS only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

This field corresponds to the COMPFILE parameters STARTREC, STARTRBA and STARTKEY.
There is no default.

Record | Key | RBA
Identifies the type of start value as described by Start above.

For>

Specifies the maximum number of records to be compare from both files.

This field corresponds to the COMPFILE parameter FOR.
Default is 0 (all records).

Use Extended Options
Select this option if Extended Unformatted Compare , Formatted Compare or Hierarchical Compare is required.

If selected then the **Compare Type - Formatted/Unformatted ?** panel view is displayed which is a springboard into the Extended Unformatted Compare Panels or Formatted Compare Panels.

Apart from the file names which are passed to the extended compare panels, all options an field values entered in the **Compare Files: Basic Options** panel view will be ignored. The user will be presented with a sequence of panels that allow specification of the extended compare options.

Extended Options should be selected only if one or more of the following are required:

◊ **Compare Position/Length** needs to be specfied separately for the NEW and OLD files e.g. compare positions 1-10 from the NEW file with positions 101-110 from the OLD file.

◊ **Record Selection** needs to be specified separately for the NEW and OLD files e.g. compare the first 100 records from the NEW file with the second 100 records from the OLD file.

◊ A record synchronisation technique is required other than 1-TO-1 or Read-Ahead with the default limit value of 100, default matching record count value of 1 and no synchronisation on blank records. e.g. Read-Ahead with a limit of 20 and matching record count 2, Unsorted Key or Sorted Key synchronisation.

◊ Formatted Compare or Hierarchical Compare is required (both using a structure (SDO) or copy book to map record data).

◊ Records flagged as having been changed, inserted and/or deleted are to be excluded from the output report.

◊ A specific report DSN and/or output files for NEW/OLD matched/inserted/deleted/changed records are required.

Compare Type - Formatted/Unformatted ?

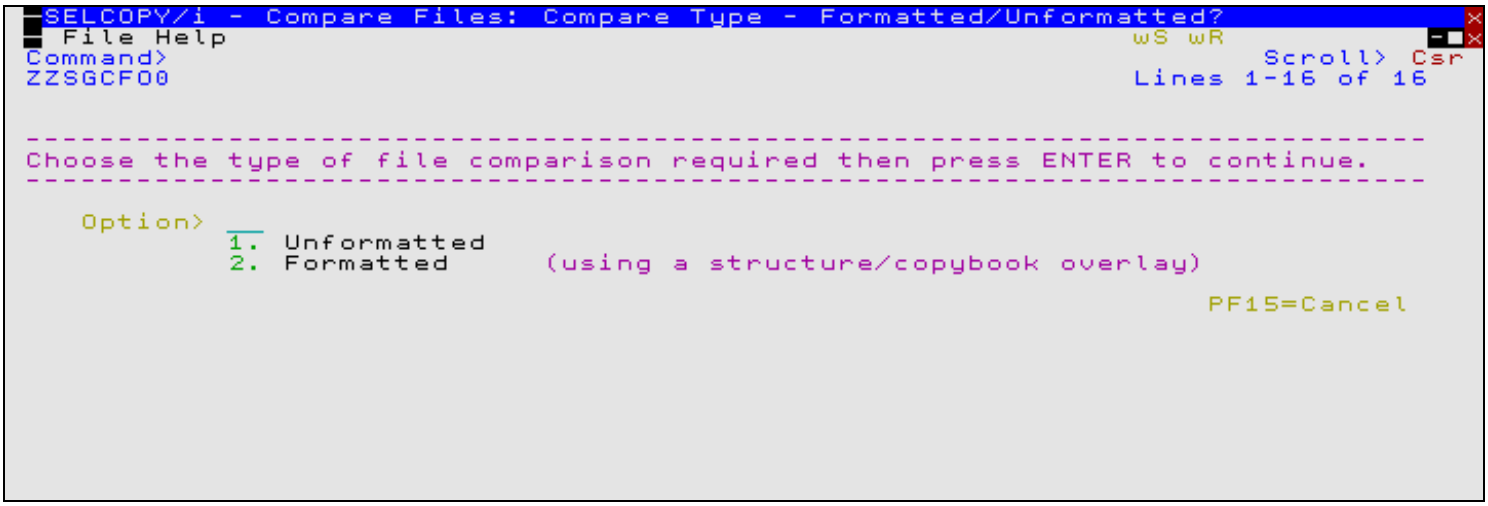


Figure 84. SELCOPY/i - Compare Files: Compare Type - Formatted/Unformatted?

Option>
Enter the number corresponding to the type of file compare required. Alternatively, position the cursor on the required type and press the <Enter> key or, if configured, double-click the left mouse button.

1. Unformatted
   Select this option for a extended unformatted compare of record data. Subsequent panels will offer only options suitable for extended unformatted compare.

2. Formatted
   Select this option for a formatted compare of structured records. Subsequent panels will require you to enter a structure (SDO) or COBOL/PL1 copybook/ADATA data set name.

   This option should also be selected if Hierarchical Compare is required.

Extended Unformatted Compare Panels

Compare Files (unformatted): New file details and options

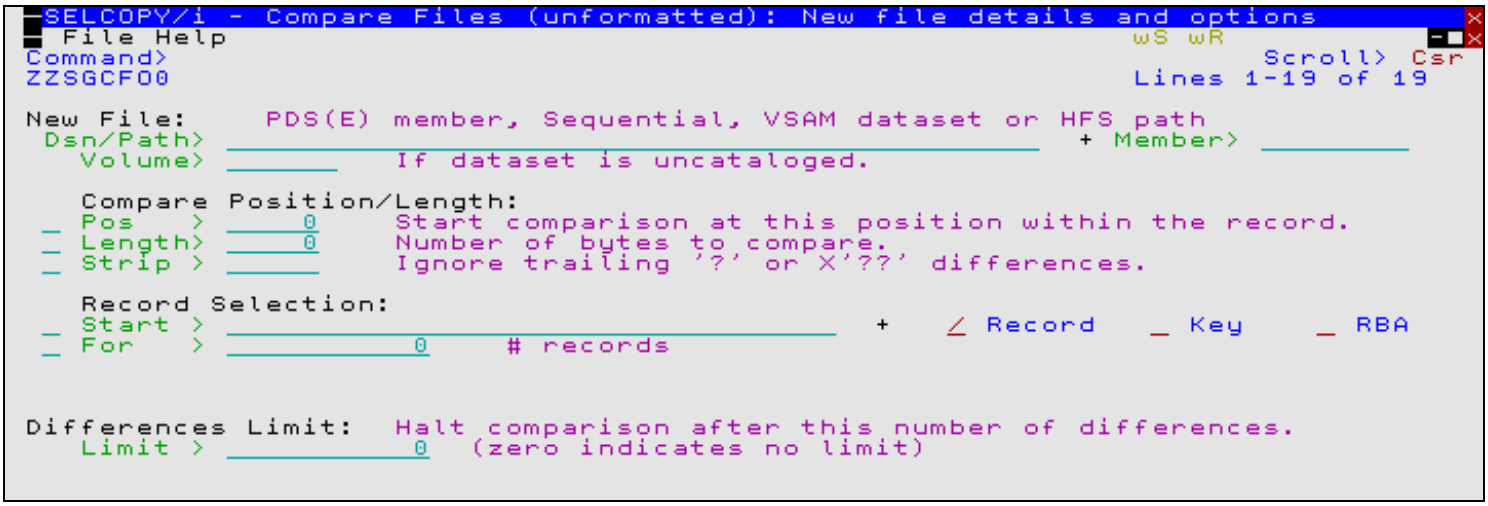


Figure 85. SELCOPY/i - Compare Files (unformatted): New file details and options.

The **Compare Files (unformatted): New file details and options** panel view is the first displayed for Extended Unformatted Compare, following selection of "Unformatted" from the Compare Type - Formatted/Unformatted ? panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will will proceed to the next Extended Unformatted Compare panel view, Compare Files (unformatted): Old file details and options.

Differences between Basic and Extended unformatted compare are described in *"Extended Unformatted Compare"*.

Dsn/Path>
Member>
Volume>

Identifies the NEW file to be compared. Dataset names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

Pos>

Record data from the NEW file will be compared from this position onwards, for the specified length, or to the end of the record if no length is specified.

This field corresponds to the COMPFILE parameter NSTARTCOL.
Default is 1.

Length>

Record data from the NEW file will be compared for this length.

If the compared record data in the NEW and OLD files are of different lengths, then a mismatch is inevitable and the record will be flagged as having been changed, inserted or deleted as appropriate to the employed synchronisation technique.

This field corresponds to the COMPFILE parameter NCOMPARELEN.
Default is the length from the start position, specified in the **Pos>** field, to the end of the record.

Strip>

Record data from the NEW file will have any trailing occurrences of the specified character stripped before comparison is actioned.

This option is particularly useful when **fixed length records** of different length are being compared, or when fixed length records are being compared with **variable length records**.

The single character may be specified as a literal **x** or **'x'**, which will be upper cased before stripping occurs, character string **C'x'** (character case preserved) or a hexadecimal string **X'nn'**.

This field corresponds to the COMPFILE parameter NSTRIP.
Default is blank (X'40').

Start>

Defines the record in the NEW file at which records will start to be compared.

User should enter a record number, an RBA number (ESDS only), or a key string (KSDS only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

This field corresponds to the COMPFILE parameters NSTARTREC, NSTARTRBA and NSTARTKEY.
There is no default.

Record | Key | RBA
Identifies the type of start value as described by Start above.

For>
Specifies the maximum number of records to be compared from the NEW file. The compare operation stops if this threshold is encountered even if the equivalent threshold for OLD file records has not been reached.

This field corresponds to the COMPFILE parameter NFOR.
Default is 0 (all records).

Limit>
Use this option in order terminate the compare process as soon as the specified number of record mismatches has been encountered.

Specifying zero or blank indicates that no limit is placed, and therefore the whole of each file (or record selection range) is processed.

This field corresponds to the COMPFILE parameter LIMIT.
Default is 0 (no limit).

Compare Files (unformatted): Old file details and options

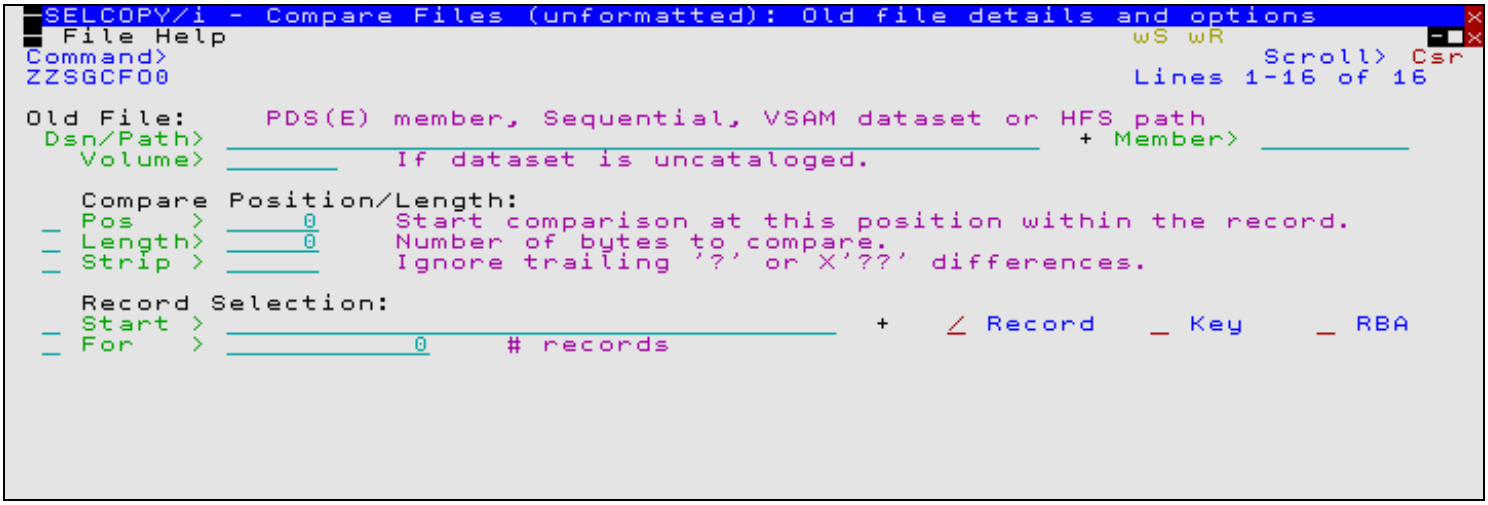


Figure 86. SELCOPY/i - Compare Files (unformatted): Old file details and options.

The **Compare Files (unformatted): Old file details and options** panel view is displayed following the Compare Files (unformatted): New file details and options panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will will proceed to the next Extended Unformatted Compare panel view, Compare Files (unformatted): Re-synchronisation options.

Dsn/Path>
Member>
Volume>
Identifies the OLD file to be compared. Dataset names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

Pos>
Record data from the OLD file will be compared from this position onwards, for the specified length, or to the end of the record if no length is specified.

This field corresponds to the COMPFILE parameter OSTARTCOL.
Default is 1.

Length>
Record data from the OLD file will be compared for this length.

If the compared record data in the NEW and OLD files are of different lengths, then a mismatch is inevitable and the record will be flagged as having been changed, inserted or deleted as appropriate to the employed synchronisation technique.

This field corresponds to the COMPFILE parameter OCOMPARELEN.
Default is the length from the start position, specified in the **Pos>** field, to the end of the record.

Strip>
Record data from the OLD file will have any trailing occurrences of the specified character stripped before comparison is actioned.

This option is particularly useful when **fixed length records** of different length are being compared, or when fixed length records are being compared with **variable length records**.

The single character may be specified as a literal **x** or **'x'**, which will be upper cased before stripping occurs, character string **C'x'** (character case preserved) or a hexadecimal string **X'nn'**.

This field corresponds to the COMPFILE parameter OSTRIP.
Default is blank (X'40').

Start>
Defines the record in the OLD file at which records will start to be compared.

User should enter a record number, an RBA number (ESDS only), or a key string (KSDS only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

This field corresponds to the COMPFILE parameters OSTARTREC, OSTARTRBA and OSTARTKEY.
There is no default.

Record | Key | RBA
Identifies the type of start value as described by Start above.

For>
Specifies the maximum number of records to be compared from the OLD file. The compare operation stops if this threshold is encountered even if the equivalent threshold for NEW file records has not been reached.

This field corresponds to the COMPFILE parameter OFOR.
Default is 0 (all records).

Compare Files (unformatted): Re-synchronisation options

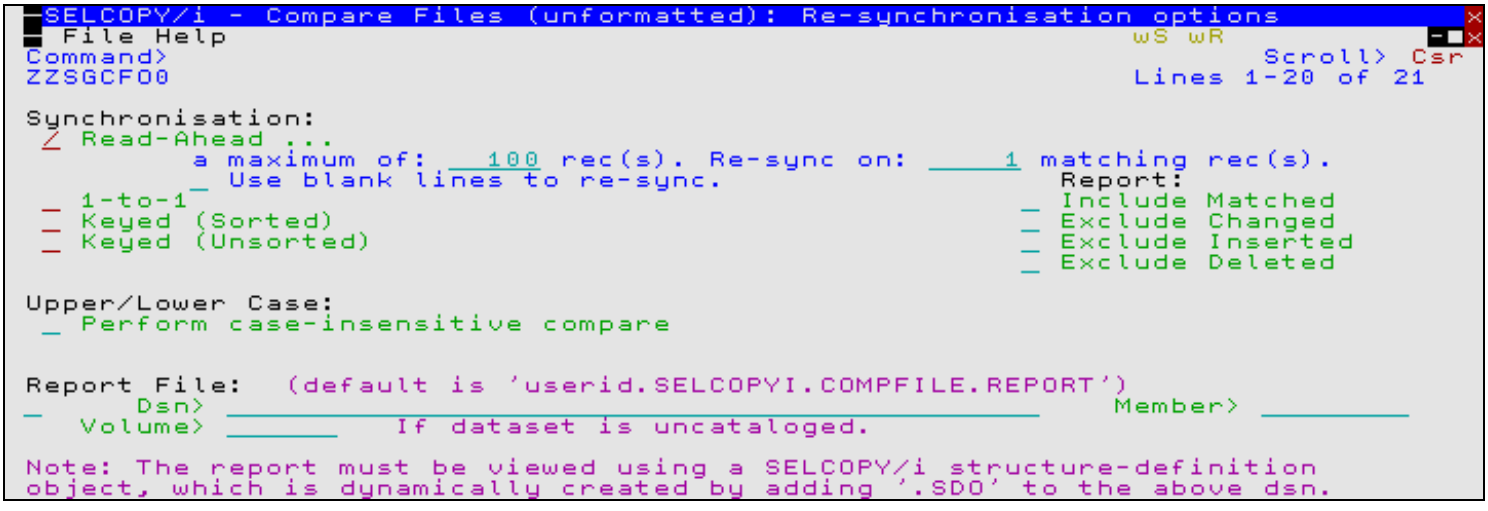


Figure 87. SELCOPY/i - Compare Files (unformatted): Re-synchronisation options.

The **Compare Files (unformatted): Re-synchronisation options** panel view is displayed following the Compare Files (unformatted): Old file details and options panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will do the following:

- If Read-Ahead or 1-to-1 synchronisation is selected, the next Extended Unformatted Compare panel view, Compare Files (unformatted): Output Files is displayed.
- If Keyed (Sorted) or Keyed (Unsorted) synchronisation is selected, then panel Compare Files (unformatted): Specify Key fields is displayed.

Read-Ahead ...
Select this option to use read-ahead record synchronisation.

Read-ahead synchronisation technique is suitable where the NEW and OLD files are predominantly comprise equal records, although some may have been changed, inserted or deleted.

When a record mismatch is detected, the compare files utility will attempt to resynchronsise the current, mismatching records by reading a specified number of records, first from the OLD file then from the NEW file, in order to find a match on a specified number of consecutive records. If successful, a synchronised record pair may be established and the compare operation continued from these records.

Records that have been skipped as a result of the read-ahead synchronisation are flagged as having been inserted or deleted as appropriate.

For a detailed description, see *"Read-Ahead Synchronisation"*.

This option corresponds to COMPFILE parameters SYNC READAHEAD.

a maximum of *RALimit* rec(s).
The maximum number of records to read-ahead in each file when attempting to establish a synchronised record pair.

For efficiency, this value should be only one more than the maximum number of expected consecutive non-matching record pairs.

This field corresponds to the number *n1* in the COMPFILE parameters SYNC READAHEAD(*n1 n2*). Default value is 100.

Re-sync on *RAMatch* matching rec(s).
The number of consecutive matching record pairs that are required in order to establish a synchronised record pair. If satisfied, the first matching record pair is identified as a synchronised record pair.

For text files such as program source, where blank comment lines are common-place, then a match on a single line is likely to produce a false synchronised record pair that results in a less accurate report. In these circumstances a higher *RAMatch* value is advised.

This field corresponds to the number *n2* in the COMPFILE parameters SYNC READAHEAD(*n1 n2*). Default value is 1.

Use blank lines to re-sync.
Since Read-ahead synchronisation is most commonly used on text type files, where matches on blank records are likely to produce false synchronised record pairs, then, by default, blank lines are ignored when encountered as part of the read-ahead synchronisation process. i.e. a matching blank line will require a further match on the next consecutive, non-blank record pair in order to qualify.

Select this option if you wish to bypass this feature.

This field corresponds to the COMPFILE READAHEAD parameter SYNCONBLANK.

1-to-1
Select this option to use 1-TO-1 record synchronisation.

For 1-TO-1 synchronisation the files are assumed to contain corresponding records, so no attempt is made to resynchronise.

For a detailed description, see *"1-TO-1 Synchronisation"*.

This option corresponds to COMPFILE parameters SYNC 1TO1.

Keyed (Sorted)
Select this option to use Sorted Key Synchronisation.

Keyed (Sorted) synchronisation type is suitable where the NEW and OLD files are sorted in ascending order based on one or more key segments within each record.

If this option is selected then a sub-panel will be opened, prompting the user to specify the required key segment(s).

An OLD and NEW file record may then be identified as a synchronised record pair when there is an exact match in all key segments of the record.

Where data mismatches occur in other parts of the records comprising the synchronised record pair, then the record is flagged as having been **changed**.

Records that are not established as being one of a synchronised record pair are reported as having been **inserted** or **deleted** as appropriate.

Sorted key synchronisation of unformatted data occurs by reading records from the file with the lower key data until a record with matching or higher key data is read. Intervening records are then treated as having been inserted or deleted.

For a detailed description, see *"Key Synchronisation"*.

This option corresponds to COMPFILE parameters SYNC KEY.

Keyed (Unsorted)
Select this option to use Unsorted Key Synchronisation.

Like the **Keyed (Sorted)** option, Keyed (Unsorted) synchronisation type is suitable where the NEW and OLD file records may be identified as a synchronised record pair by an exact match in all key segments of the record. However, records are not sorted into ascending order based on these key segments.

Synchronisation of keyed unsorted records occurs using the read-ahead method as described for Read-Ahead synchronisation above. The difference being that data need only match in the defined key segments to qualify as a potential synchronised record pair.

The associated read-ahead record limit, number of consecutive matching record pairs and blank line synchronisation may also be specified for unsorted key synchronisation using the same fields used for Read-Ahead synchronisation. Namely "a maximum of *RALimit* rec(s).", "Re-sync on *RAMatch* matching rec(s)." and "Use blank lines to re-sync."

This option corresponds to COMPFILE parameters SYNC UNSORTED KEY.

Default
If "Default" is chosen and both files are VSAM KSDS with equal defined key lengths, then keyed synhronisation is employed, with the key locations automatically assigned from the KSDS file definitions.

For all other file combinations, "Default" employs **Read-Ahead** synhronisation.

Include Matched
Select this option to include matching records in the output report file.

This option corresponds to COMPFILE parameter INCMATCHED.

Exclude Changed
Select this option to exclude changed records from the output report file.

This option corresponds to COMPFILE parameter EXCHANGED.

Exclude Inserted
Select this option to exclude inserted records from the output report file.

This option corresponds to COMPFILE parameter EXINSERTED.

Exclude Inserted
Select this option to exclude deleted records from the output report file.

This option corresponds to COMPFILE parameter EXDELETED.

Show Context
Select this option to include a specified number of (possibly matching) records immediately before and after each detected difference in order to provide context without including all matching records, which for large files is likely to be prohibitive.

Note that this option will be ignored if "Include Matched" is already selected.

nn Lines Top/Bottom
The number of context lines to be displayed before and after each difference.

This option corresponds to COMPFILE parameter "CONTEXT nn".

nn Gap Lines
The number of "Gap" records to display in order to separate each difference context block.

This option corresponds to COMPFILE parameter "GAP nn".

Perform case-insensitive compare
Select this option to perform a case insensitive compare. For unformatted compare, **all data** will be translated to upper case before comparison.

This option corresponds to COMPFILE parameter CASEINSENSITIVE (synonym CASEIGNORE).

Ignore embedded blanks
Select this option to allow differences in the number of blanks between words to be ignored.

This means that should two corresponding records mismatch, then ALL embedded blanks will be removed from both records before the comparison is retried. If there are no other differences then the records are treated as equal.

Note that this option may prove particularly CPU intensive when **READAHEAD** synchronisation option is in effect.

This option corresponds to COMPFILE parameter SPACE.

Report File:
Dsn>
Member>
Volume>

If the Report File option field is selected, then these fields identify the fileid of the file to which the compare files utility report records will be written. Dataset names must be fully qualified, quotes being unnecessary but permitted.

For "Type> FMT" (see below), the report is a structured data file designed to be browsed (not printed) using a structure definition object (SDO), which will also be generated by the compare files utility.

The associated SDO fileid is constructed simply by adding **.SDO** to the report fileid. Therefore, for the DSN of the report file is restricted to 40 bytes in length.
Report output to an HFS dataset is not currently supported.

If the report file and/or the SDO file do not already exist, then they will automatically be allocated by the compare files utility, relying on SMS ACS to select a suitable storage group of eligable DASD volumes. The report file is allocated using DCB geometry RECFM=VB, LRECL=32756, BLKSIZE=0 and a space allocation of TRACKS(150,75). The SDO is allocated using DCB geometry RECFM=VB, LRECL=16380, BLKSIZE=0 and a space allocation of TRACKS(2,2).

If this option is not specified, *fileid* defaults to "*user*.SELCOPYI.COMPFILE.REPORT" with SDO fileid "*user*.SELCOPYI.COMPFILE.REPORT.SDO".

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

These fields correspond to COMPFILE parameter REPORT.

Type> `Default | FMT | TEXT`
This field corresponds to the COMPFILE option "LIST=".

A selection list will be displayed if the option entered is unrecognised or left blank. Choose from one of the following options:

FMT
The report generated is a **structured data file** designed to be (automatically) browsed (not printed) from within a SELCOPYi online session, using a structure definition (SDO) which is also generated automatically during execution of the compare.

TEXT
The report generated is a more traditional formatted plain text document, designed to be printed if necessary.

Default
FMT if running online, **TEXT** if running from JCL in batch.

Compare Files (unformatted): Specify Key fields

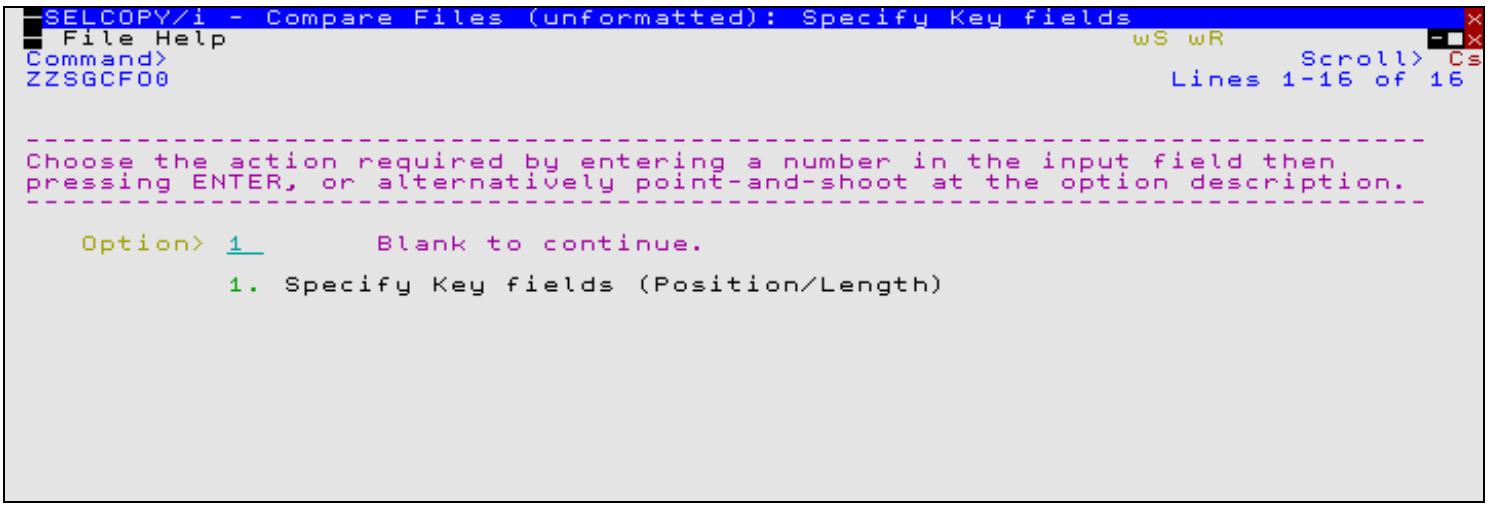


Figure 88. SELCOPY/i - Compare Files (unformatted): Specify Key fields.

The **Compare Files (unformatted): Specify Key fields** panel view is displayed following the Compare Files (unformatted): Re-synchronisation options panel view if Keyed (Sorted) or Keyed (Unsorted) synchronisation is selected.

Option>
Enter the number corresponding to the type of key field specification. Alternatively, position the cursor on the required type and press the <Enter> key or, if configured, double-click the left mouse button.

For unformatted compare, key segments may be specified by position and length only.

A blank in this field will proceed to the next Extended Unformatted Compare panel view, Compare Files (unformatted): Output Files.

1. **Specify Key fields (Position/Length)**
   A separate panel will be displayed in which the user can enter a table row entry for each required key segment.

   Each table row entry consists of the key length, and position in both NEW and OLD file records.

Compare Files: Specify Key Pos/Len

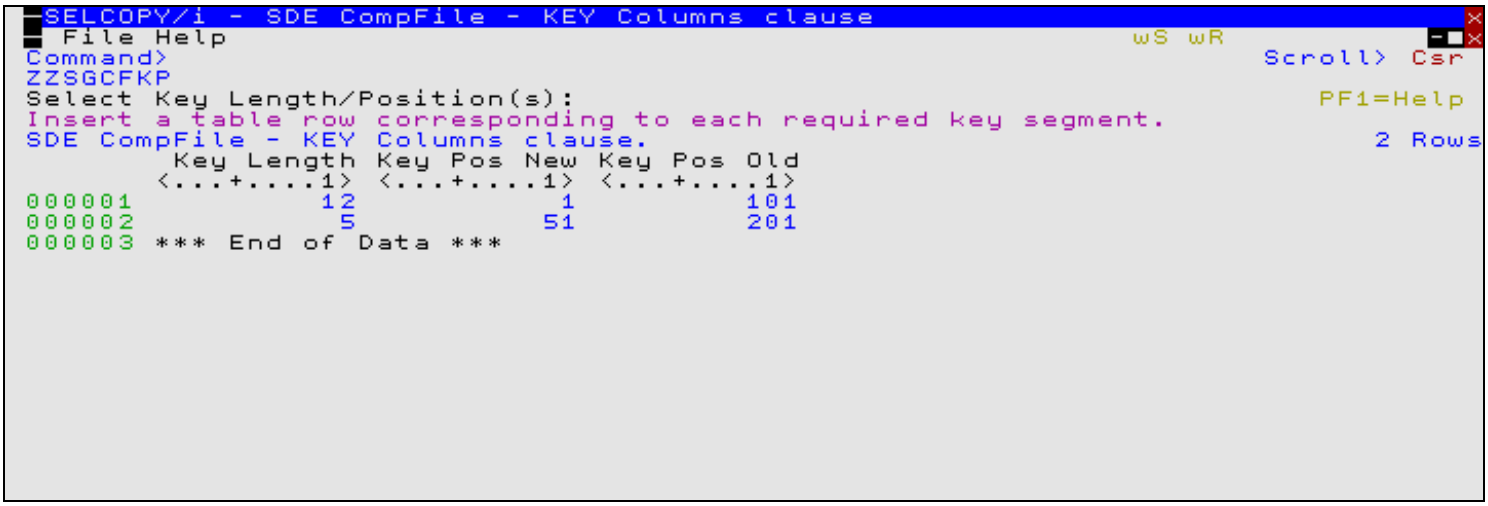


Figure 89. SELCOPY/i - Compare Files - Specify Key Pos/Len.

The **Compare Files: Specify Key Pos/Len** panel (ZZSGCFKP) is displayed following selection of option 1. from the Compare Files (unformatted): Specify Key fields panel view.

Standard SELCOPY/i table editing techniques should be used to add a table row entry for each required key segment.

Each table row entry consists of a key length and key start positions in the NEW and OLD file records. Although the key length is fixed, the key position may differ in NEW and OLD file records.

For both unformatted compare and formatted compare, key segments specified using fixed positions and lengths apply to all record types. For record type specific keys (hierarchical compare), key segments must be selected by column name.

Key segments should be entered in the order in which they constitute the key. This is particularly important when identifying a record key to be used for sorted key synchronisation. For unsorted key synchronisation, the order in which key segments have been entered is the order in which the segments will be compared when establishing a synchronised record pair. Performance may be improved if key segments containing volatile data are specified first.

Pressing <PF3> to exit the panel, will also save the table of key segments and return to the **Compare Files (unformatted): Specify Key fields** panel view.

Compare Files (unformatted): Output Files

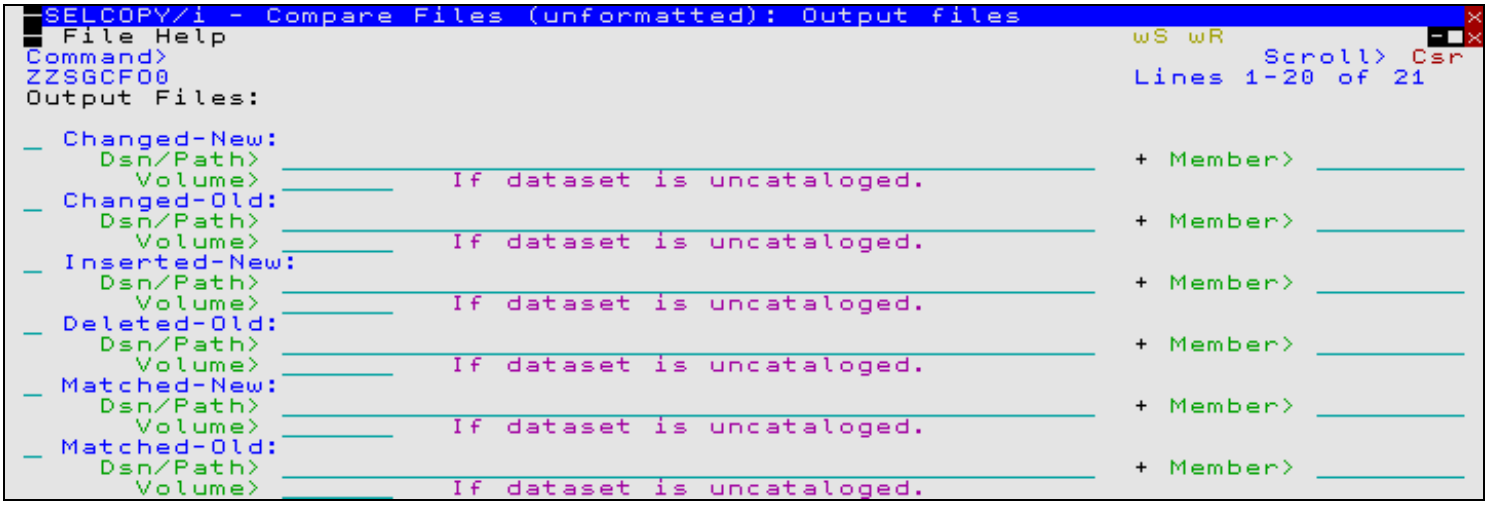


Figure 90. SELCOPY/i - Compare Files (unformatted): Output Files.

If Keyed synchronisation was specified, the **Compare Files (unformatted): Output Files** panel view is displayed following the Compare Files (unformatted): Specify Key fields panel view. Otherwise, this panel view is displayed following the Compare Files (unformatted): Re-synchronisation options panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will proceed to the next Extended Unformatted Compare panel view, Compare Files (unformatted): Options / Action.

The Output Files panel view identifies the output files to which a record from the NEW and/or OLD files are to be copied, based on its flagged status (matched, changed, inserted or deleted). The output fileid may be an HFS file path, sequential data set or PDS/PDSE library member.

Data set names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards are entered, or if a dataset is specified which is a PDS/PDSE library and the member field is left blank.

If a specified output file is non-HFS and does not already exist, then it will automatically be allocated by the compare files utility, relying on SMS ACS to select a suitable storage group of eligable DASD volumes. The data set is allocated using DCB RECFM, LRECL and BLKSIZE geometry that best matches the NEW or OLD file as appropriate.

Changed-New:
Dsn/Path>
Member>
Volume>

If the Changed-New option field is selected, then these fields identify the fileid of the file to which NEW file records, flagged as having been changed (CN), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITECN *cn_fileid*.

Changed-Old:
Dsn/Path>
Member>
Volume>

If the Changed-Old option field is selected, then these fields identify the fileid of the file to which OLD file records, flagged as having been changed (CO), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITECO *co_fileid*.

Inserted-New:
Dsn/Path>
Member>
Volume>

If the Inserted-New option field is selected, then these fields identify the fileid of the file to which NEW file records, flagged as having been inserted (I), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITEIN *in_fileid*.

Deleted-Old:
Dsn/Path>
Member>
Volume>

If the Deleted-Old option field is selected, then these fields identify the fileid of the file to which OLD file records, flagged as having been deleted (D), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITEDO *do_fileid*.

Matched-New:
Dsn/Path>
Member>
Volume>

If the Matched-New option field is selected, then these fields identify the fileid of the file to which NEW file records, flagged as being matched, are to be copied.

This fileid corresponds to the COMPFILE parameter WRITEMN *mn_fileid*.

Matched-Old:
Dsn/Path>
Member>
Volume>

If the Matched-Old option field is selected, then these fields identify the fileid of the file to which OLD file records, flagged as being matched, are to be copied.

This fileid corresponds to the COMPFILE parameter WRITEMO *mo_fileid*.

Compare Files (unformatted): Options / Action

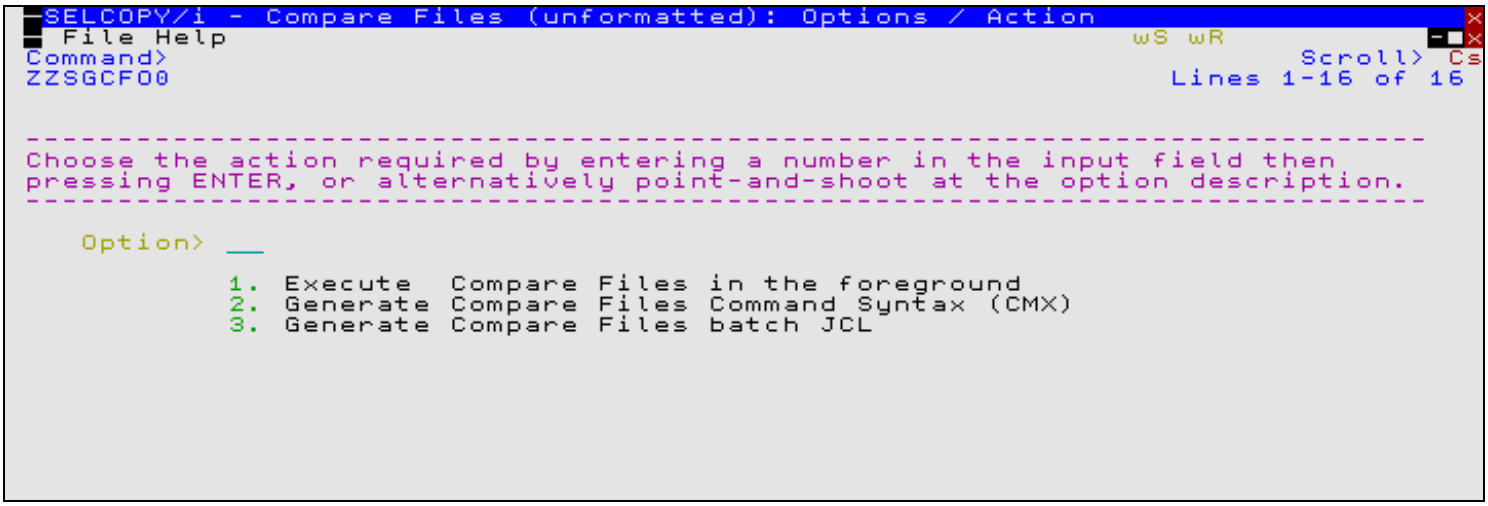


Figure 91. SELCOPY/i - Compare Files (unformatted): Options / Action.

The **Compare Files (unformatted): Options** / **Action** panel view is the last of the Extended Unformatted Compare panels and is displayed following the Compare Files (unformatted): Output Files panel view.

Option>
Enter the number corresponding to the action required. Alternatively, position the cursor on the action description and press the <Enter> key or, if configured, double-click the left mouse button.

1. **Execute Compare Files in the foreground**
   The compare utility will run from your SELCOPY/i session and the structured output report file will be automatically displayed in an SDE browse window view.

2. **Generate Compare Files Command Syntax (CMX)**
   COMPFILE command line syntax to run the compare files utility using the chosen options is generated and placed in a temporary CMX file. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

3. **Generate Compare Files batch JCL**
   Creates a JCL job stream that executes the **SDEAMAIN** program. SDEIN input comprises the COMPFILE command with parameters reflecting options specified in these panels.

   The output report, generated on execution of this batch job, may be viewed from your SELCOPY/i session by issuing the command **CFOUT *report_file_name***, or by issuing CFOUT as a prefix command against the report DSN or member name in a dataset list or library list window.

Formatted Compare Panels

The Formatted Compare panels also provide facility to specify field values necessary for Hierarchical Compare.

Compare Files (formatted): New file details and options

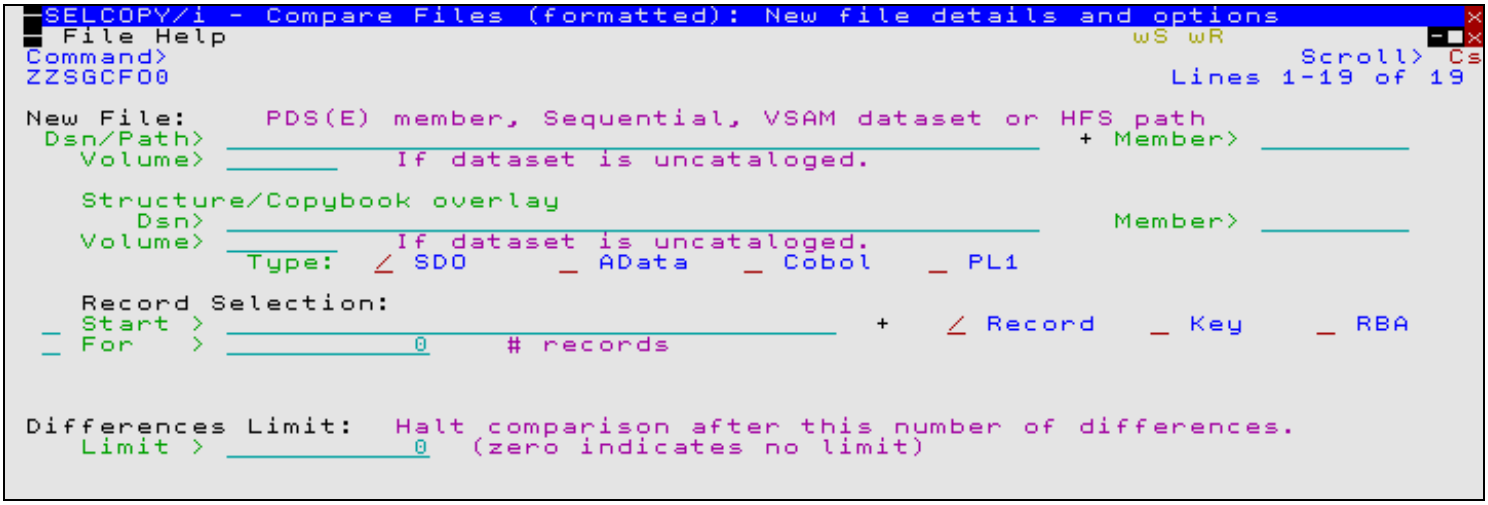


Figure 92. SELCOPYi - Compare Files (formatted): New file details and options.

The **Compare Files (formatted): New file details and options** panel view is the first displayed for Formatted/Hierarchical Compare, following selection of "Formatted" from the Compare Type - Formatted/Unformatted ? panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will will proceed to the next Formatted Compare panel view, Compare Files (formatted): Old file details and options.

Descriptions of these types of compare may be found under *"Formatted Compare"* and *"Hierarchical Compare"*.

Dsn/Path>
Member>
Volume>

Identifies the NEW file to be compared. Dataset names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

This field is mandatory.

Structure/Copybook overlay
Dsn/Path>
Member>
Volume>
Type

Specifies the structure type (SDO, COBOL or PL1 Copybook, COBOL or PL1 ADATA output) and structure name (sequential data set or PDS/PDSE library member) to be used to map record data in NEW file.

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

This field corresponds to the COMPFILE parameter USING SDO/COBOL/PL1/ADATA *new_structname*.
There is no default.

Recompile>

If *Structure/Copybook overlay* refers to a COBOL, PL1 or ADATA source file (not an SDO) then a compile step must be performed in order to turn the source copybook into SELCOPYi's own internal structure (SDO) format. A temporary SDO will be created, lasting the duration of the SELCOPYi session, making subsequent reference to the same copybook during the session much faster.

If, however, the copybook (or any of its included components) is modified during the session, then a recompile of the source will normally be required.

For performance reasons **N** (meaning no recompile should occur) is the default.

Specify **Y** to force a copybook recompile each time the process is run. Alternatively type the primary command:
SD DROP *<copybook_name>*

Start>
Defines the record in the NEW file at which records will start to be compared.

User should enter a record number, an RBA number (ESDS only), or a key string (KSDS only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

This field corresponds to the COMPFILE parameters NSTARTREC, NSTARTRBA and NSTARTKEY.
There is no default.

Record | Key | RBA
Identifies the type of start value as described by Start above.

For>
Specifies the maximum number of records to be compared from the NEW file. The compare operation stops if this threshold is encountered even if the equivalent threshold for OLD file records has not been reached.

This field corresponds to the COMPFILE parameter NFOR.
Default is 0 (all records).

Limit>
Use this option in order terminate the compare process as soon as the specified number of record mismatches has been encountered.

Specifying zero or blank indicates that no limit is placed, and therefore the whole of each file (or record selection range) is processed.

This field corresponds to the COMPFILE parameter LIMIT.
Default is 0 (no limit).

Compare Files (formatted): Old file details and options

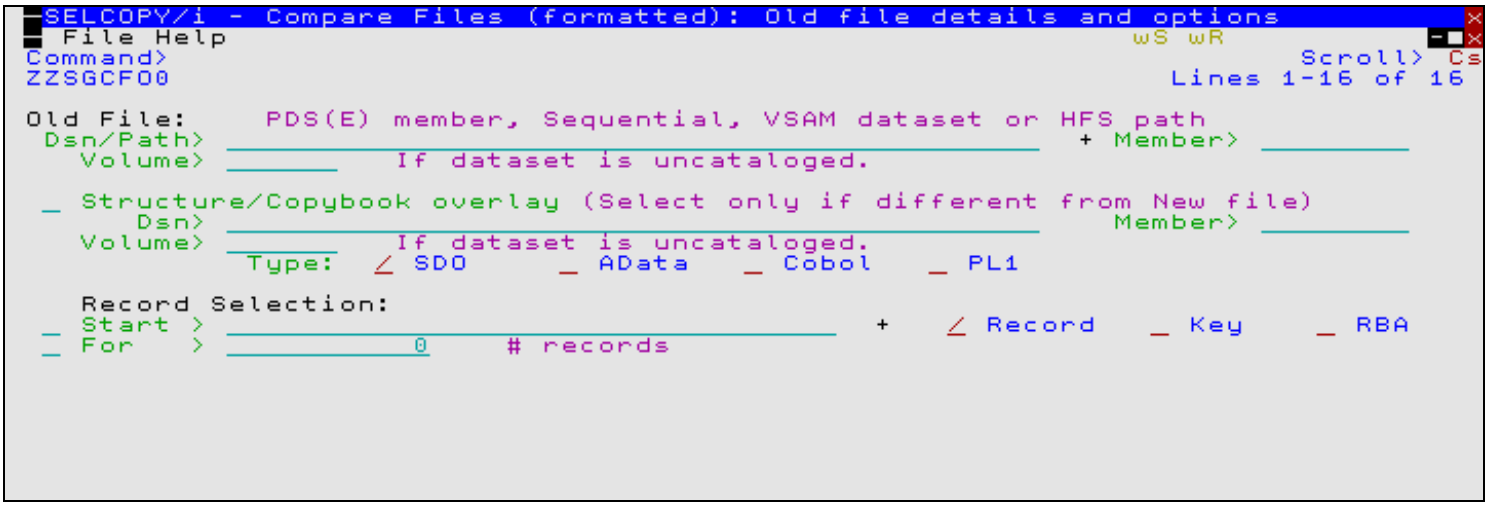


Figure 93. SELCOPYi - Compare Files (formatted): Old file details and options.

The **Compare Files (formatted): Old file details and options** panel view is displayed following the Compare Files (formatted): New file details and options panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will will proceed to the next Formatted/Hierarchical Compare panel view, Compare Files (formatted): Re-synchronisation options.

Dsn/Path>
Member>
Volume>
Identifies the OLD file to be compared. Dataset names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

This field is mandatory.

Structure/Copybook overlay
Dsn/Path>
Member>
Volume>
Type

If the Structure/Copybook overlay option field is selected, then these fields specify the structure type (SDO, COBOL or PL1 Copybook, COBOL or PL1 ADATA output) and structure name (sequential data set or PDS/PDSE library member) to be used to map record data in OLD file.

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

This field corresponds to the COMPFILE parameter USING SDO/COBOL/PL1/ADATA *old_structname*. There is no default.

These fields are required only if a structure is to be used for the OLD file which is different to that used by the NEW file.

If Record type definitions of the same name exist in both the NEW and OLD structures, then these are automatically **"matched"**.

Similarly, fields of the same name within these matched record types will automatically be "matched" and subsequently compared.

i.e. Only records assigned record types that are "matched" are eligible for compare and only record data occupying fields that are "matched" are compared.

Type the MAP primary command to interactively match-up Record-types and Fields that are **not identically named**.

Matched Fields will be compared without error even though they may be of different data types. Note that a character field, when compared with a field of numeric data type, must contain valid numeric data (potentially including exponent 'E' or 'e', sign and/or exponent sign '+' or '-', decimal point '.' and/or commas ',')

Recompile>

If *Structure/Copybook overlay* refers to a COBOL, PL1 or ADATA source file (not an SDO) then a compile step must be performed in order to turn the source copybook into SELCOPYi's own internal structure (SDO) format. A temporary SDO will be created, lasting the duration of the SELCOPYi session, making subsequent reference to the same copybook during the session much faster.

If, however, the copybook (or any of its included components) is modified during the session, then a recompile of the source will normally be required.

For performance reasons **N** (meaning no recompile should occur) is the default.

Specify **Y** to force a copybook recompile each time the process is run. Alternatively type the primary command:
SD DROP *<copybook_name>*

Start>

Defines the record in the OLD file at which records will start to be compared.

User should enter a record number, an RBA number (ESDS only), or a key string (KSDS only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

This field corresponds to the COMPFILE parameters OSTARTREC, OSTARTRBA and OSTARTKEY.
There is no default.

Record | Key | RBA

Identifies the type of start value as described by Start above.

For>

Specifies the maximum number of records to be compared from the OLD file. The compare operation stops if this threshold is encountered even if the equivalent threshold for NEW file records has not been reached.

This field corresponds to the COMPFILE parameter OFOR.
Default is 0 (all records).

Compare Files (formatted): Re-synchronisation options

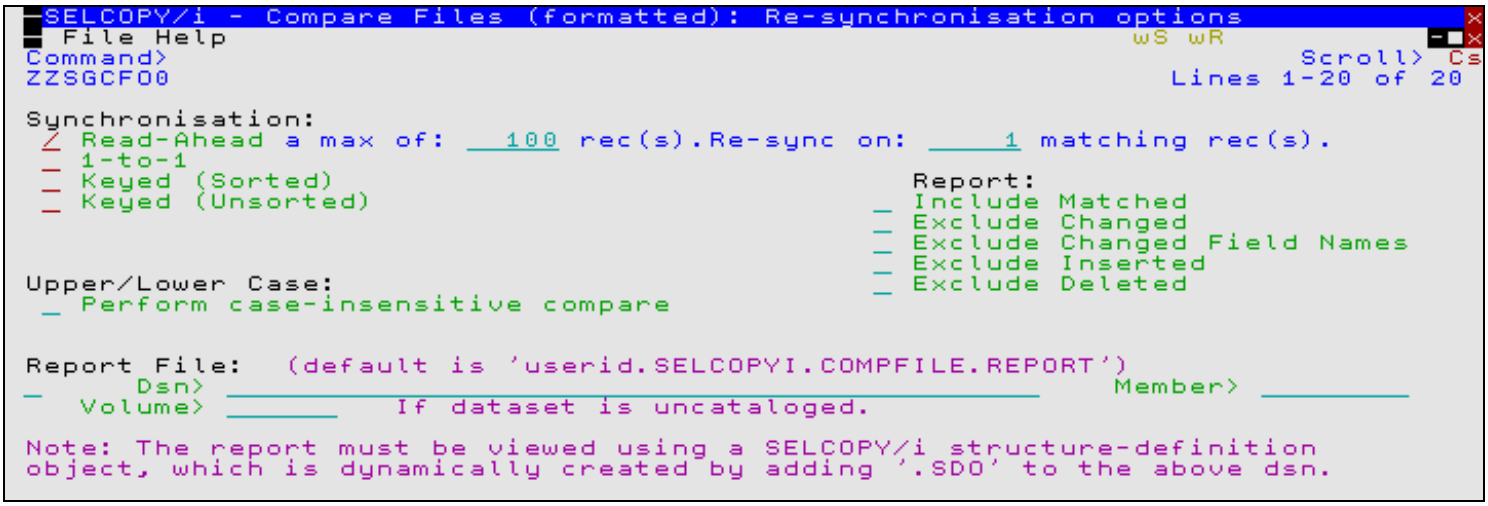


Figure 94. SELCOPYi - Compare Files (formatted): Re-synchronisation options.

The **Compare Files (formatted): Re-synchronisation options** panel view is displayed following the Compare Files (formatted): Old file details and options panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will will do the following:

- If Read-Ahead, 1-to-1 or Default synchronisation is selected, the next Formatted Compare panel view, Compare Files (formatted): Output Files is displayed.
- If Keyed (Sorted) or Keyed (Unsorted) synchronisation is selected, then panel Compare Files (formatted): Specify Key fields is displayed. Note that Keyed synchronisation is required for **Hierarchical Compare**.

`Read-Ahead ...`
Select this option to use read-ahead record synchronisation.

Read-ahead synchronisation technique is suitable where the NEW and OLD files are predominantly comprise equal records, although some may have been changed, inserted or deleted.

When a record mismatch is detected, the compare files utility will attempt to resynchronsise the current, mismatching records by reading a specified number of records, first from the OLD file then from the NEW file, in order to find a match on a specified number of consecutive records. If successful, a synchronised record pair may be established and the compare operation continued from these records.

Note that, before attempting match data in a record pair, Read-ahead synchronisation of formatted records will first verify that the record pair are of the same record type.

Records that have been skipped as a result of the read-ahead synchronisation are flagged as having been inserted or deleted as appropriate.

For a detailed description, see *"Read-Ahead Synchronisation"*.

This option corresponds to COMPFILE parameters SYNC READAHEAD.

`a maximum of RALimit rec(s).`
The maximum number of records to read-ahead in each file when attempting to establish a synchronised record pair.

For efficiency, this value should be only one more than the maximum number of expected consecutive non-matching record pairs.

This field corresponds to the number *n1* in the COMPFILE parameters SYNC READAHEAD(*n1 n2*). Default value is 100.

`Re-sync on RAMatch matching rec(s).`
The number of consecutive matching record pairs that are required in order to establish a synchronised record pair. If satisfied, the first matching record pair is identified as a synchronised record pair.

This field corresponds to the number *n2* in the COMPFILE parameters SYNC READAHEAD(*n1 n2*). Default value is 1.

`1-to-1`
Select this option to use 1-TO-1 record synchronisation.

For 1-TO-1 synchronisation the files are assumed to contain corresponding records, so no attempt is made to resynchronise.

For a detailed description, see *"1-TO-1 Synchronisation"*.

This option corresponds to COMPFILE parameters SYNC 1TO1.

Keyed (Sorted)
Select this option to use Sorted Key Synchronisation or Sorted Hierarchical Key Synchronisation.

Keyed (Sorted) synchronisation type is suitable where the NEW and OLD files are sorted based on one or more key segments within each record.

If this option is selected then a sub-panel will be opened, prompting the user to specify the required key segment(s).

An OLD and NEW file record may then be identified as a synchronised record pair when there is an exact match in all key segments of the record.

Where data mismatches occur in other parts of the records comprising the synchronised record pair, then the record is flagged as having been **changed**.

Records that are not established as being one of a synchronised record pair are reported as having been **inserted** or **deleted** as appropriate.

In general, synchronisation occurs by reading records from the file with the lower key data until a record with matching or higher key data is read. Intervening records are then treated as having been inserted or deleted.

For Sorted Hierarchical Key Synchronisation, record types that have been defined a key must be sorted in ascending sort order within its hierarchical group. e.g. records of sorted keyed record type, ALBUM, has child records of sorted keyed record type, TRACKS. Records of record type TRACKS are sorted by field Track_Number (the key field) which starts at Track_Number=1 following each new ALBUM parent record.

For a detailed description, see *"Key Synchronisation"* and *"Hierarchical Key Synchronisation"*.

This option corresponds to COMPFILE parameters SYNC KEY.

Keyed (Unsorted)
Select this option to use Unsorted Key Synchronisation and Unsorted Hierarchical Key Synchronisation.

Like the **Keyed (Sorted)** option, Keyed (Unsorted) synchronisation type is suitable where the NEW and OLD file records may be identified as a synchronised record pair by an exact match in all key segments of the record. However, records are not sorted into ascending order based on these key segments.

Synchronisation of keyed unsorted records occurs using the read-ahead method as described for Read-Ahead synchronisation above. The difference being that data need only match in the defined key segments to qualify as a potential synchronised record pair.

The associated read-ahead record limit and number of consecutive matching record pairs may also be specified for unsorted key synchronisation using the same fields used for Read-Ahead synchronisation. Namely "a maximum of *RALimit* rec(s)." and "Re-sync on *RAMatch* matching rec(s)."

This option corresponds to COMPFILE parameters SYNC UNSORTED KEY.

Default
If "Default" is chosen and both files are VSAM KSDS with equal defined key lengths, then keyed synhronisation is employed, with the key locations automatically assigned from the KSDS file definitions.

For all other file combinations, "Default" employs **Read-Ahead** synhronisation.

Include Matched
Select this option to include matching records in the output report file.

This option corresponds to COMPFILE parameter INCMATCHED.

Exclude Changed
Select this option to exclude changed records from the output report file.

For a formatted compare only, reporting a record which is flagged as having been changed will not only display the formatted record data from the NEW and OLD files, but also a number of **Field** records which identify the name of each changed field.

Note that, opting to exclude these records may result in a significant performance improvement since the process of comparing field-by-field is terminated at the first mismatch in each record.

This option corresponds to COMPFILE parameter EXCHANGED.

Exclude Changed Field Values
Select this option to indicate that field values for any changed records are not to be included in the output report.

Note that specifying this option may result in a significant performance improvement since the process of comparing field-by-field is terminated at the first mismatch in each record.

This option corresponds to COMPFILE parameter EXFIELDCHANGED.

Exclude Inserted
Select this option to exclude inserted records from the output report file.

This option corresponds to COMPFILE parameter EXINSERTED.

Exclude Inserted
Select this option to exclude deleted records from the output report file.

This option corresponds to COMPFILE parameter EXDELETED.

Include Field Values
Select this option to indicate that field values for any matched, inserted and/or deleted rows are eligible to be included in the output report.

For example, field values will be displayed for inserted rows provided that the "Exclude Inserted" option is **not selected.**

This option corresponds to COMPFILE parameter INCFIELDS.

Include all Keyed records
For sorted and unsorted KEY synchronisation only, records containing a matching key field should be displayed in the output report.

This is of primary use when performing a formatted compare of hierarchical records where records are assigned to different record types and keys fields are defined in multiple record types.

This option corresponds to COMPFILE parameter INCKEYS.

Show Context
Select this option to include a specified number of (possibly matching) records immediately before and after each detected difference in order to provide context without including all matching records, which for large files is likely to be prohibitive.

Note that this option will be ignored if "Include Matched" is already selected.

nn Lines Top/Bottom
The number of context lines to be displayed before and after each difference.

This option corresponds to COMPFILE parameter "CONTEXT nn".

nn Gap Lines
The number of "Gap" records to display in order to separate each difference context block.

This option corresponds to COMPFILE parameter "GAP nn".

Perform case-insensitive compare
Select this option to perform a case insensitive compare. For a formatted and hierarchical compare, character (AN) fields will be translated to upper case before comparison.

This option corresponds to COMPFILE parameter CASEINSENSITIVE (synonym CASEIGNORE).

Report File:
Dsn>
Member>
Volume>
If the Report File option field is selected, then these fields identify the fileid of the file to which the compare files utility report records will be written. Dataset names must be fully qualified, quotes being unnecessary but permitted.

For "Type> FMT" (see below), the report is a structured data file designed to be browsed (not printed) using a structure definition object (SDO), which will also be generated by the compare files utility.

The associated SDO fileid is constructed simply by adding **.SDO** to the report fileid. Therefore, for the DSN of the report file is restricted to 40 bytes in length.
Report output to an HFS dataset is not currently supported.

If the report file and/or the SDO file do not already exist, then they will automatically be allocated by the compare files utility, relying on SMS ACS to select a suitable storage group of eligable DASD volumes. The report file is allocated using DCB geometry RECFM=VB, LRECL=32756, BLKSIZE=0 and a space allocation of TRACKS(150,75). The SDO is allocated using DCB geometry RECFM=VB, LRECL=16380, BLKSIZE=0 and a space allocation of TRACKS(2,2).

If this option is not specified, *fileid* defaults to "*user*.SELCOPYI.COMPFILE.REPORT" with SDO fileid "*user*.SELCOPYI.COMPFILE.REPORT.SDO".

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

These fields correspond to COMPFILE parameter REPORT.

Type> `Default | FMT | TEXT`
This field corresponds to the COMPFILE option "LIST=".

A selection list will be displayed if the option entered is unrecognised or left blank. Choose from one of the following options:

FMT
The report generated is a **structured data file** designed to be (automatically) browsed (not printed) from within a SELCOPYi online session, using a structure definition (SDO) which is also generated automatically during execution of the compare.

TEXT
The report generated is a more traditional formatted plain text document, designed to be printed if necessary.

Default
FMT if running online, **TEXT** if running from JCL in batch.

Compare Files (formatted): Specify Key fields

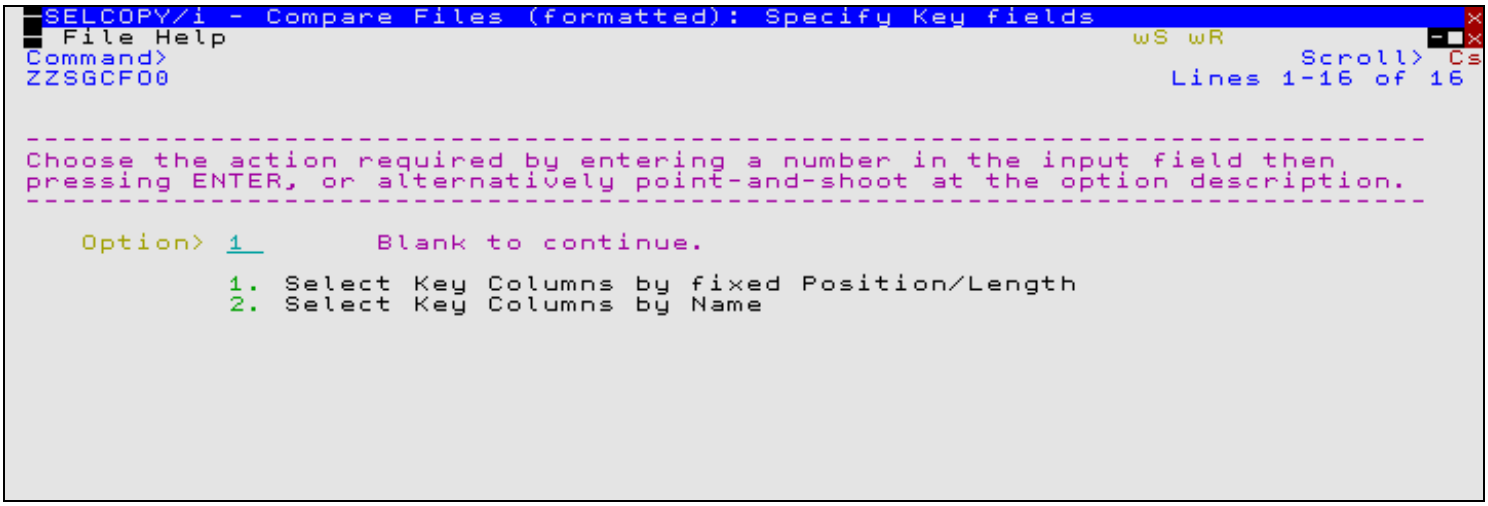


Figure 95. SELCOPYi - Compare Files (formatted): Specify Key fields.

Option>
Enter the number corresponding to the type of key field specification. Alternatively, position the cursor on the required type and press the <Enter> key or, if configured, double-click the left mouse button.

Formatted compare key segments may be specified by position and length or by formatted column name. Hierarchical compare key segments must be specified by formatted column name.

A blank in this field will proceed to the next Extended Formatted Compare panel view, Compare Files (formatted): Output Files.

1. **Select Key Columns by fixed Position/Length**
   A separate panel will be displayed in which the user can enter a table row entry for each required key segment.

   Each table row entry consists of the key length, and position in both NEW and OLD file records.

2. **Select Key Columns by Name**
   A separate panel will be displayed containing a list of record types defined by the NEW structure (SDO) or copybook. For informational purposes only, each record type is accompanied by its **USE WHEN** condition, where specified, which enables SDE to identify each particular record type.

   From this panel, select those record types to which key segments are to be defined. A panel will be opened for each selected record type allowing the user to exclude and re-order the field columns, leaving only those to be used as key segments.

Compare Files: Specify Key Pos/Len

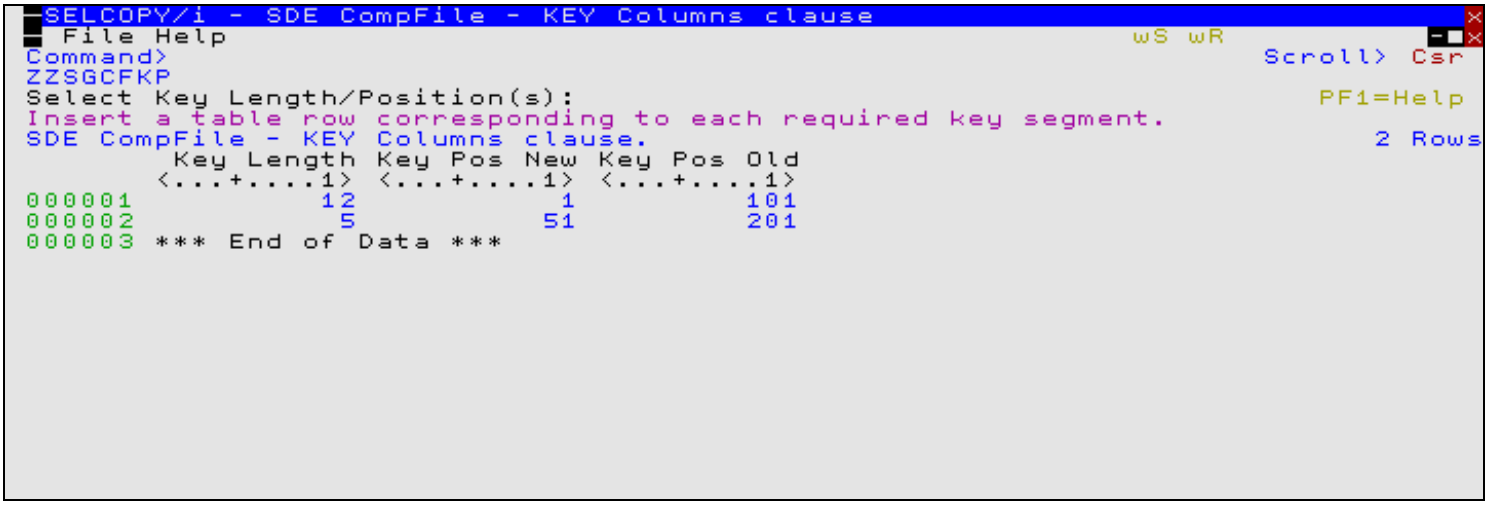


Figure 96. SELCOPYi - Compare Files - Specify Key Pos/Len.

The **Compare Files: Specify Key Pos/Len** panel (ZZSGCFKP) is displayed following selection of option 1. from the Compare Files (formatted): Specify Key fields panel view.

Standard SELCOPYi table editing techniques should be used to add a table row entry for each required key segment.

Each table row entry consists of a key length and key start positions in the NEW and OLD file records. Although the key length is fixed, the key position may differ in NEW and OLD file records.

For both unformatted compare and formatted compare, key segments specified using fixed positions and lengths apply to all record types. For record type specific keys (hierarchical compare), key segments must be selected by column name.

Key segments should be entered in the order in which they constitute the key. This is particularly important when identifying a record key to be used for sorted key synchronisation. For unsorted key synchronisation, the order in which key segments have been entered is the order in which the segments will be compared when establishing a synchronised record pair. Performance may be improved if key segments containing volatile data are specified first.

Pressing <PF3> to exit the panel, will also save the table of key segments and return to the **Compare Files (formatted): Specify Key fields** panel view.

SELCOPYi Compare Files - KEY Columns (Record Types List)

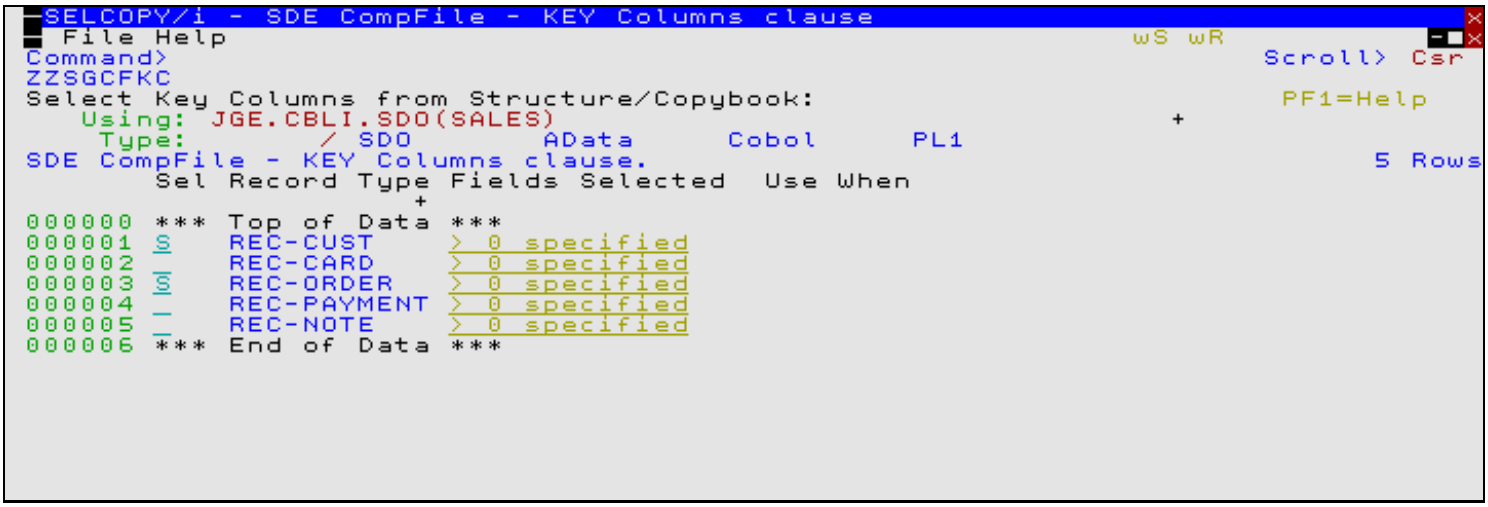


Figure 97. SELCOPYi - Compare Files - KEY Columns (Record-Types List)

The **Compare Files - Key Columns (Record Types List)** panel (ZZSGCFKC) is displayed following selection of option 2. from the Compare Files (formatted): Specify Key fields panel view.

If more than one record type is displayed in this list (i.e. the structure contains multiple record type definitions), then selecting key segment fields for any of these record types will imply Hierarchical Compare.

Select each record type for which key columns are to be specified by entering 'S' against the record type in the **Sel** column or by positioning the cursor on the required record type then either pressing the <Enter> key or, if configured, double-click the left mouse

button. To deselect the record type key field definition, remove the 'S' against its entry in the 'Sel' column.

For each selected record type, the Compare Files - Select from Field Names List panel (ZZSGCFOF) is opened displaying a list of fields comprising that record type. The list of field names should be edited so that only the required key fields are diplayed in the correct order.

On return from the selectable field list, the **Fields Selected** column will be updated to indicate the number of fields included in the key.

For Hierarchical Compare it is imperative that the list of record types are ordered so that the level 1 (highest priority) keyed record type occurs first in the list, followed by the level 2 keyed record type, etc. Use standard SELCOPYi table editing techniques to re-order the record types as required.

Pressing <PF3> to exit the panel, will also save the table of selected keyed record types and return to the **Compare Files (formatted): Specify Key fields** panel view.

SELCOPYi Compare Files - Select from Field Names List

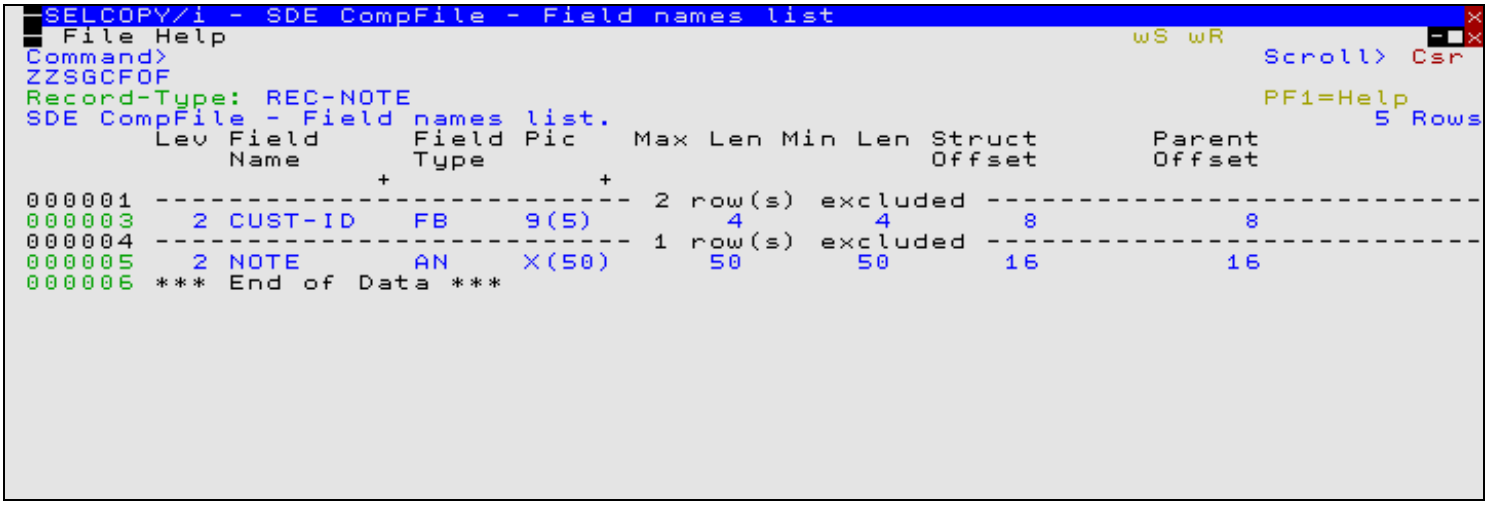


Figure 98. SELCOPYi - Compare Files - Select from Field Names List.

The **Compare Files - Select from Field Names List** panel (ZZSGCFOF) is displayed for each record type selected from the Compare Files - Key Columns (Record Types List) panel.

A list of field names, defined by the selected record type, is presented to the user as an editable table. Standard SELCOPYi table editing techniques should be used to exclude and re-order the fields so that only fields which comprise the key are displayed and in the correct order.

Only included entries are used to define the key, therefore a field name may be excluded (as opposed to deleted) in order to remove it from the key. This has the benefit that it may easily be included again later if necessary. For example, the following commands may be executed to filter (include) specific table rows:

`WHERE SelectFld >> 'ABC-' or #1 << 'DEF-'`
: Exclude all entries except those where the Field Name begins with literal "**ABC-**", or "**DEF-**". Note that the Field Name column is field reference number 1.

`LESS SelectTyp='AN'`
: Exclude all entries where the Field Type is **AN**. (Entries that were already excluded will be unaffected.)

Key field segments should occur in the order in which they constitute the key. This is particularly important when identifying a record key to be used for sorted key synchronisation. For unsorted key synchronisation, the order in which key fields have been entered is the order in which the key fields will be compared when establishing a synchronised record pair. Performance may be improved if key fields containing volatile data are specified first.

Pressing <PF3> to exit the panel, will also save the table of selected key field names and return to the **Compare Files - Key Columns (Record Types List)** panel.

Compare Files (formatted): Output Files

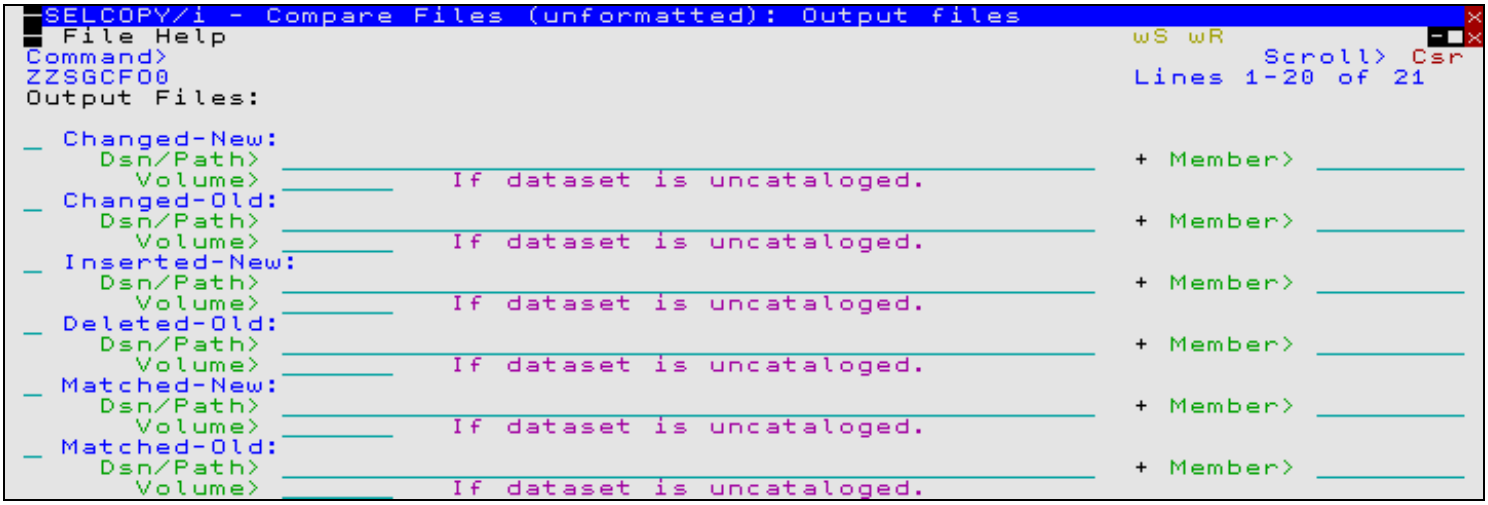


Figure 99. SELCOPYi - Compare Files (formatted): Output Files.

If Keyed synchronisation was specified, the **Compare Files (formatted): Output Files** panel view is displayed following the Compare Files (formatted): Specify Key fields panel view. Otherwise, this panel view is displayed following the Compare Files (formatted): Re-synchronisation options panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will proceed to the next Formatted/Hierarchical Compare panel view, Compare Files (formatted): Options / Action.

The Output Files panel view identifies the output files to which a record from the NEW and/or OLD files are to be copied, based on its flagged status (matched, changed, inserted or deleted). The output fileid may be an HFS file path, sequential data set or PDS/PDSE library member.

Data set names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards are entered, or if a dataset is specified which is a PDS/PDSE library and the member field is left blank.

If a specified output file is non-HFS and does not already exist, then it will automatically be allocated by the compare files utility, relying on SMS ACS to select a suitable storage group of eligable DASD volumes. The data set is allocated using DCB RECFM, LRECL and BLKSIZE geometry that best matches the NEW or OLD file as appropriate.

Changed-New:
Dsn/Path>
Member>
Volume>

If the Changed-New option field is selected, then these fields identify the fileid of the file to which NEW file records, flagged as having been changed (CN), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITECN *cn_fileid*.

Changed-Old:
Dsn/Path>
Member>
Volume>

If the Changed-Old option field is selected, then these fields identify the fileid of the file to which OLD file records, flagged as having been changed (CO), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITECO *co_fileid*.

Inserted-New:
Dsn/Path>
Member>
Volume>

If the Inserted-New option field is selected, then these fields identify the fileid of the file to which NEW file records, flagged as having been inserted (I), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITEIN *in_fileid*.

Deleted-Old:
Dsn/Path>
Member>
Volume>

If the Deleted-Old option field is selected, then these fields identify the fileid of the file to which OLD file records, flagged as having been deleted (D), are to be copied.

This fileid corresponds to the COMPFILE parameter WRITEDO *do_fileid*.

Matched-New:
Dsn/Path>
Member>
Volume>

If the Matched-New option field is selected, then these fields identify the fileid of the file to which NEW file records, flagged as being matched, are to be copied.

This fileid corresponds to the COMPFILE parameter WRITEMN *mn_fileid*.

Matched-Old:
Dsn/Path>
Member>
Volume>

If the Matched-Old option field is selected, then these fields identify the fileid of the file to which OLD file records, flagged as being matched, are to be copied.

This fileid corresponds to the COMPFILE parameter WRITEMO *mo_fileid*.

Compare Files (formatted): Options / Action

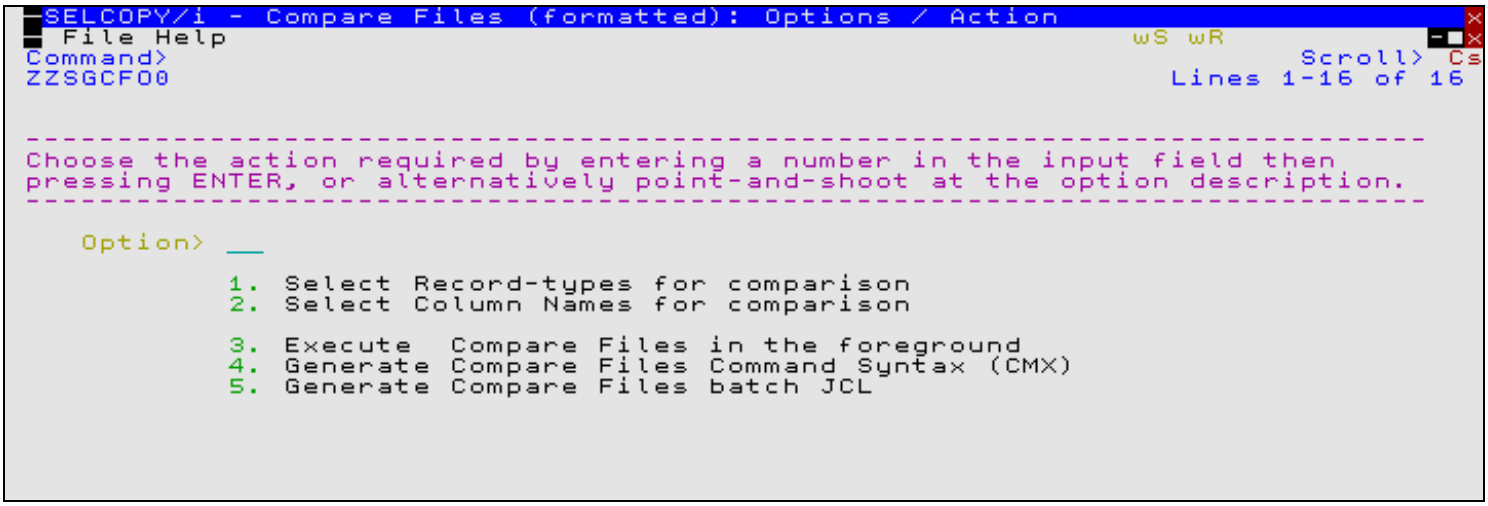


Figure 100. SELCOPYi - Compare Files (formatted): Options / Action.

Option>

Enter the number corresponding to the action required. Alternatively, position the cursor on the action description and press the <Enter> key or, if configured, double-click the left mouse button.

1. **Select Record Types for comparison**
   By default records of any record type are included in the compare process. Use this option to restrict the compare process to records assigned the specified record types only.

   A separate panel will be displayed containing a list of record types defined by the NEW structure (SDO) or copybook. For informational purposes only, each record type is accompanied by its **USE WHEN** condition, where specified, which enables SDE to identify each particular record type.

   From this panel, select those record types to included in the compare process.

2. **Select Column Names for comparison**
   By default all fields (from all selected record-types) are included in the compare process. Use this option to restrict the compare process to a subset of field names identified within those record types selected for compare.

   A separate panel will be displayed containing a list of record types defined by the NEW structure (SDO) or copybook. For informational purposes only, each record type is accompanied by its **USE WHEN** condition, where specified, which enables SDE to identify each particular record type.

   From this panel, select those record types for which fields are to be excluded from the compare process. Another panel will be opened for each selected record type allowing the user to exclude and field columns from the compare.

3. **Execute Compare Files in the foreground**
   The compare utility will run from your SELCOPYi session and the structured output report file will be automatically displayed in an SDE browse window view.

4. **Generate Compare Files Command Syntax (CMX)**
   COMPFILE command line syntax to run the compare files utility using the chosen options is generated and placed in a temporary CMX file. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

5. **Generate Compare Files batch JCL**
   Creates a JCL job stream that executes the **SDEAMAIN** program. SDEIN input comprises the COMPFILE

command with parameters reflecting options specified in these panels.

Following The output report, generated on execution of this batch job, may be viewed from your SELCOPYi session by issuing the command **CFOUT *report_file_name***, or by issuing CFOUT as a prefix command against the report DSN or member name in a dataset list or library list window.

SELCOPYi Compare Files - Select Record Types to Compare

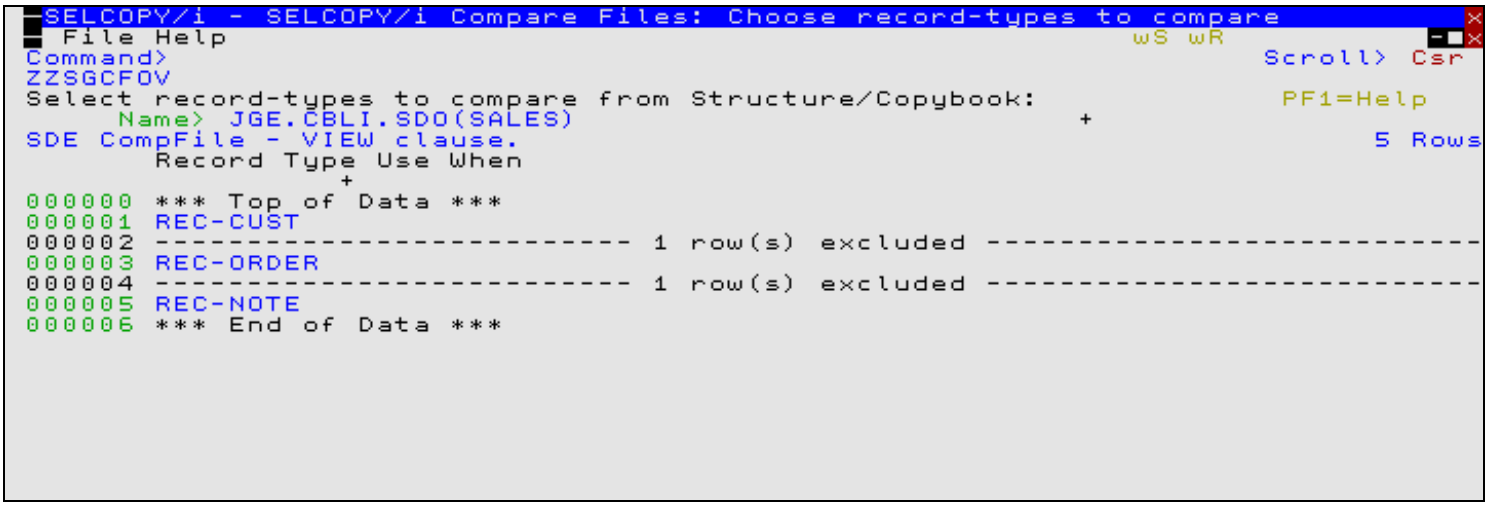


Figure 101. SELCOPYi - Compare Files - Select Record-Types.

The **Compare Files - Select Record Types to Compare** panel (ZZSGCFOV) is displayed following selection of option 1. from the Compare Files (formatted): Options / Action panel view.

This panel contains a list of all record types defined by the NEW structure (SDO) or copybook presented to the user as an editable table. Any **USE WHEN** condition, used to determine whether record data fits the record type definition, is also displayed.

Records assigned a record type that is included in this list will be eligible for compare. Use standard SELCOPYi table editing techniques to exclude or delete record types from this list and so exclude records assigned these record types from being compared. For example, the following command may be executed to exclude all rows except those where the record type name begins with literal "**ABC-**":

```
WHERE   ViewRT >> 'ABC-'
```

Pressing <PF3> to exit the panel, will also save the table of selected keyed record types and return to the **Compare Files (formatted): Options** / **Action** panel view.

This panel corresponds to the COMPFILE parameter VIEW *rectype*.
Default is to include all records in the NEW and OLD files in the compare regardless of whether they are assigned a record type in the NEW structure.

SELCOPYi Compare Files - Select Field Names to Compare

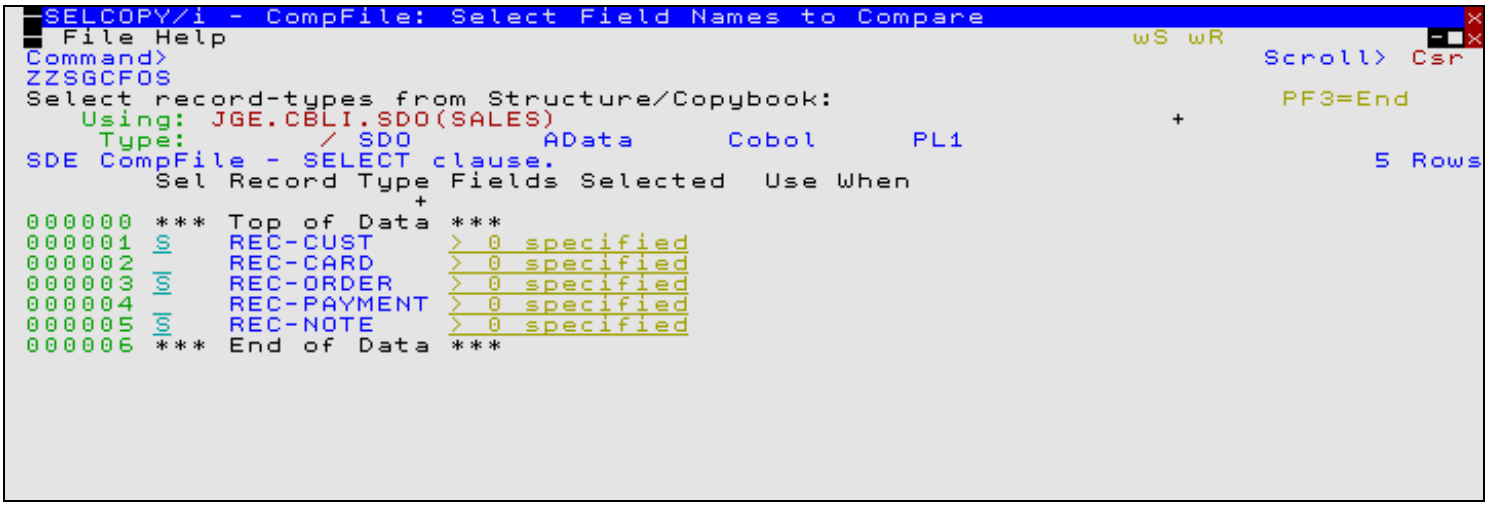


Figure 102. SELCOPYi - Compare Files - Select Field Names to Compare.

The **Compare Files - Select Field Names to Compare** panel (ZZSGCFOS) is displayed following selection of option 2. from the Compare Files (formatted): Options / Action panel view.

Select each record type for which specific field columns are to be selected by entering 'S' against the record type in the **Sel** column or by positioning the cursor on the required record type then either pressing the <Enter> key or, if configured, double-click the left mouse button. To deselect the record type key field definition, remove the 'S' against its entry in the 'Sel' column.

For each selected record type, the Compare Files - Select from Field Names List panel (ZZSGCFOF) is opened displaying a list of fields comprising that record type. The list of field names should be edited so that only the required key fields are diplayed. The order in which these fields occur in this list will be the order in which they are compared.

On return from the selectable field list, the **Fields Selected** column will be updated to indicate the number of fields included for compare.

Note that, selecting fields from a record type that has been excluded from the compare in panel Compare Files - Select Record-Types (ZZSGCFOV), will generate the appropriate COMPFILE syntax but will ultimately have no effect. The record type must be included for compare in order for its selected fields to be compared.

Pressing <PF3> to exit the panel, will also save the table of selected record types for which selected fields will be compared, and return to the **Compare Files (formatted): Options** / **Action** panel view.

This panel corresponds to the COMPFILE parameters SELECT *fieldname*, ... FROM *rectype*
Default is to include all field columns of the same name belonging to record types of the same name in *old_structname* and *new_structname*.

SELCOPYi Compare Files - Select Field Names List

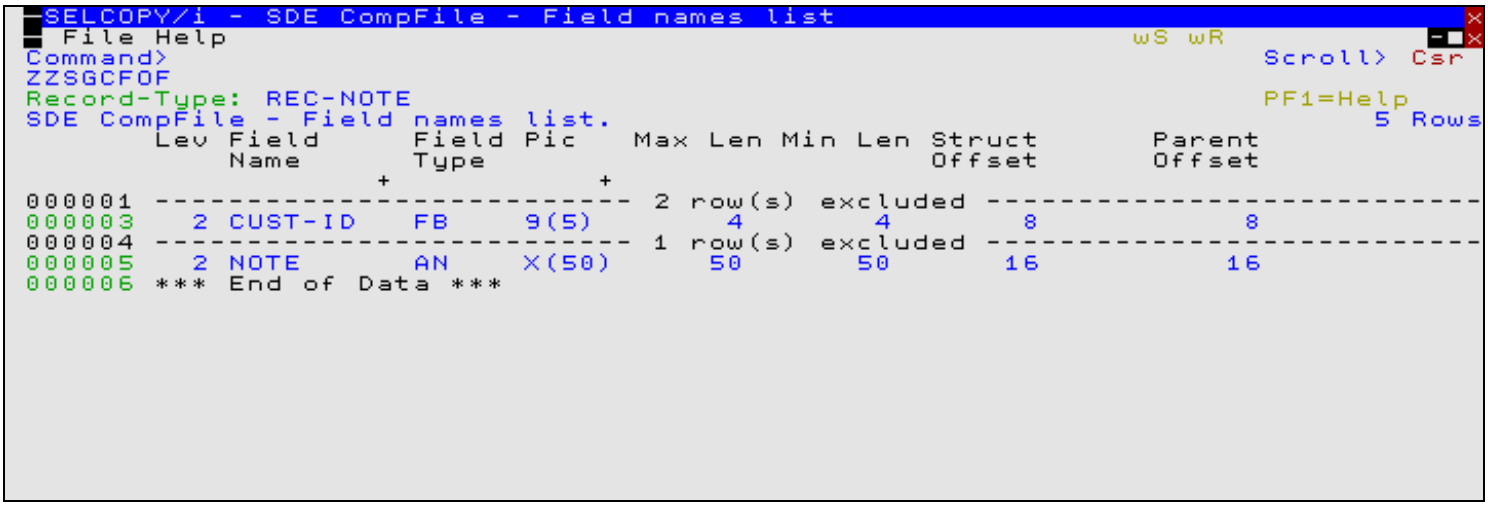


Figure 103. SELCOPYi - Compare Files - Select from Field Names List.

The **Compare Files - Select from Field Names List** panel (ZZSGCFOF) is displayed for each record type selected from the Compare Files - Select Field Names to Compare panel.

A list of field names, defined by the selected record type, is presented to the user as an editable table. Standard SELCOPYi table editing techniques should be used to exclude and re-order the fields so that only fields which are to be compared are displayed in the order in which they are to be compared.

Only included field name entries are compared, therefore a field name may be excluded (as opposed to deleted) in order to exclude it from the compare. This has the benefit that it may easily be included again later if necessary. For example, the following commands may be executed to filter (include) specific table rows:

`WHERE (length(strip(SelectFld),'T') > 5) and (#3 = 'BN')`
: Exclude all rows except those where the length of the Field Name entry is greater than 5 and the Field Picture Type is "BN". Note that the "Field Pic Type" column is field reference number 3.

`MORE SelectLev < 3`
: Include previously excluded entries where the field level is 1 or 2. (Entries that are already included will remain included.)

The order in which the field names occur is the order in which the fields will be compared. Performance may be improved if the Exclude Changed option is set and fields where differences are expected are specified first in this list.

For Formatted or Hierarchical compare involving sorted or unsorted key synchronisation where key segments are specified as field names, then the key segment field names may be excluded from the compare. This is because the key segment fields must match in order to establish the synchronised record pair prior to comparing the remaining record field data.

Pressing <PF3> to exit the panel, will also save the table of selected field names and return to the **Compare Files - Select Field Names to Compare** panel.

This panel corresponds to the COMPFILE parameters SELECT *fieldname*, ... FROM *rectype*
Default is to include all field columns of the same name belonging to record types of the same name in *old_structname* and *new_structname*.

Compare Files Output

Report Format

The report generated by the compare files utility is available in two formats controlled by the LIST=FMT|TEXT option.

LIST=TEXT is the default when **COMPFILE** is executed from JCL as a batch job. Using this option the report generated is a traditional formatted text document, designed to be printed if necessary.

LIST=FMT is the default when **COMPFILE** is executed within the **SELCOPYi online environment**. Using this option the report generated is a **structured data file**. This is designed to be (automatically) browsed (not printed) from within a SELCOPYi online session,

The associated SDO is automatically generated when the compare files utility is run. The SDO dataset name is always the DSN of the report with a suffix of "**.SDO**". If the report output is to a PDS/PDSE library member, then the SDO will be written to a library member of the same name belonging to PDS/PDSE library DSN with suffix ".SDO".

If the report file and/or the SDO file do not already exist, then they will automatically be allocated by the compare files utility, relying on SMS ACS to select a suitable storage group of eligable DASD volumes. The report file is allocated using DCB geometry RECFM=VB, LRECL=32756, BLKSIZE=0 and a space allocation of TRACKS(150,75). The SDO is allocated using DCB geometry RECFM=VB, LRECL=16380, BLKSIZE=0 and a space allocation of TRACKS(2,2).

The default DSN for the report and SDO data sets is "*user*.SELCOPYI.COMPFILE.REPORT" and "*user*.SELCOPYI.COMPFILE.REPORT.SDO" respectively.

If the compare process is run in the foreground of a SELCOPY/i session, then the output report file is automatically browsed on completion.

If the compare process is run as a batch job and **LIST=FMT** is explicitly coded, then the output report file may then be subsequently browsed from your SELCOPY/i session by issuing the command **CFOUT *report_file_name***, or by issuing CFOUT as a prefix command against the report DSN or member name in a dataset list or library list window.

In order to display the more of the record data, when a **LIST=FMT** output report is viewed on an 80-character width 3270 terminal, the prefix area and report record type Compare fields **zNewRecNo**, **zOldRecNo** and **zLrecl** are automatically suppressed from view.

Figure 104. SELCOPY/i - Compare Files - Output Report - 80 column screen.

To view these suppressed fields press function key <F2> with the cursor on any report record to see it in vertical format, with all fields included. (Further useful function key definitions are detailed below.) Alternativeley, type primary command **SELect *** with the cursor positioned on any record of record type "Compare" to reveal the suppressed fields. If necessary, the primary command **PREfix ON n** will provide a prefix area of length *n* (default is 8).

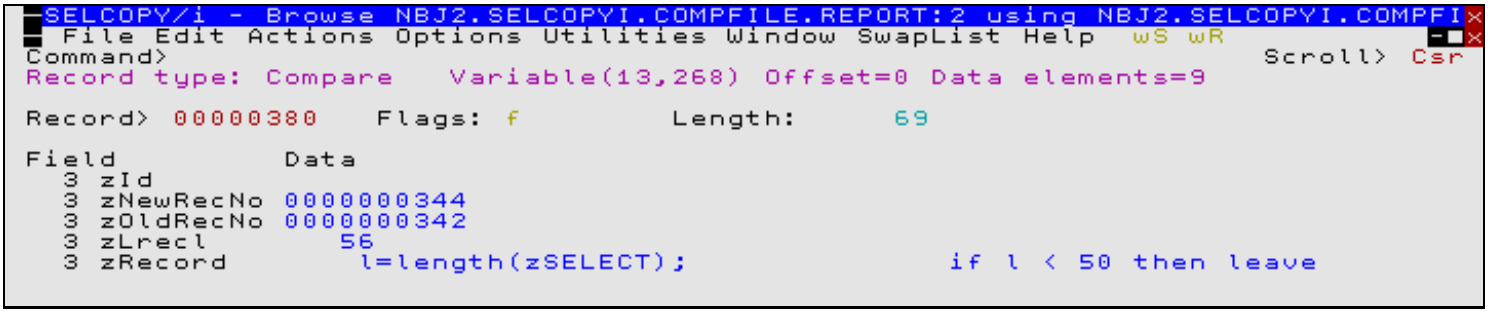


Figure 105. SELCOPY/i - Compare Files - Output Report - Zoomed view.

A compare files output report viewed on a 3270 terminal of width greater than 80 characters will include a prefix area with no report field suppression.

```
SELCOPY/i - Browse NBJ2.SELCOPYI.COMPFILE.REPORT using NBJ2.SELCOPYI.COMPFILE.REPORT.SDO     32752 V SEQ
 File Edit Actions Options Utilities Window SwapList Help  wS wR
Command>                                                                                    Scroll> Csr
Record type: Compare   Variable(13,268) Offset=0 Data elements=9
         zId  zNewRecNo  zOldRecNo zLrecl zRecord
         <>  <---+----> <---+----> <---> <---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+--
00000374     0000000338 0000000336     73 AddSELECT:                                          /* *** .select .sel */
00000375     0000000339 0000000337     63   if xSELECT     <> '/'    then return   /* Option not active */
00000376     0000000340 0000000338      1
00000377     0000000341 0000000339     74   'ec i   SELECT                                                    \'
00000378     0000000342 0000000340     26   zSELECT = strip(zSELECT)
00000379     0000000343 0000000341     12   do forever
00000380     0000000344 0000000342     56     l=length(zSELECT);                if l < 50 then leave
00000381 I   0000000345                51                     p=lastpos(' ',left(zSELECT,50));
00000382 I   0000000346                51     if p =  0 then p=lastpos(',',left(zSELECT,50));
00000383 I   0000000347                24     if p =  0 then leave
00000384 D              0000000343     57     p=lastpos(' ',left(zSELECT,50)); if p =  0 then leave
00000385     0000000348 0000000344     53     interpret 'parse var zSELECT  SELECT' p 'zSELECT'
00000386     0000000349 0000000345     30      SELECT    = strip( SELECT)
00000387     0000000350 0000000346     30     zSELECT    = strip(zSELECT)
00000388 I   0000000351                41     'ec i            'left( SELECT,53) '\'
00000389 D              0000000347     41     'ec i            'left(xSELECT,53) '\'
00000390     0000000352 0000000348      5   end
00000391     0000000353 0000000349     41   'ec   i            'left(zSELECT,53) '\'
00000392     0000000354 0000000350      6 return
00000393     0000000355 0000000351      1
00000394     0000000356 0000000352      1
00000395     0000000357 0000000353      1
00000396     0000000358 0000000354      1
00000397     0000000359 0000000355     72 AddWHERE:                                           /* *** .where .wh  */
00000398     0000000360 0000000356     63   if xWHERE      <> '/'    then return   /* Option not active */
00000399     0000000361 0000000357      1
00000400     0000000362 0000000358     74   'ec i   WHERE  (                                                  \'
00000401     0000000363 0000000359     24   zWHERE = strip(zWHERE)
00000402     0000000364 0000000360     49   if pos('|'      ,zWHERE  ) > 0 then IncludesOR=1
00000403     0000000365 0000000361     49   if pos(linend.2,zWHERE  ) > 0 then IncludesLE=1
00000404     0000000366 0000000362     12   do forever
00000405     0000000367 0000000363     56     l=length(zWHERE);                if l < 50 then leave
00000406     0000000368 0000000364     57     p=lastpos(' ',left(zWHERE,50)); if p =   0 then leave
00000407     0000000369 0000000365     50     interpret 'parse var zWHERE  WHERE' p 'zWHERE'
00000408     0000000370 0000000366     29      WHERE     = strip( WHERE)
00000409     0000000371 0000000367     29     zWHERE     = strip(zWHERE)
00000410     0000000372 0000000368     41     'ec i            'left( WHERE ,53) '\'
00000411     0000000373 0000000369      5   end
00000412     0000000374 0000000370     41   'ec   i            'left(zWHERE ,53) '\'
00000413     0000000375 0000000371     74   'ec i          )                                                   \'
00000414     0000000376 0000000372      6 return
00000415     0000000377 0000000373      1
00000416     0000000378 0000000374      1
00000417     0000000379 0000000375      1
00000418     0000000380 0000000376      1
00000419     0000000381 0000000377     70 AddFIND:                                            /* *** .find .f  */
00000420     0000000382 0000000378     63   if xFIND       <> '/'    then return   /* Option not active */
00000421     0000000383 0000000379      1
00000422     0000000384 0000000380     74   'ec i   FIND   (                                                  \'
00000423     0000000385 0000000381      1
---- Press PF1 for Help, PF4 for options, PF6 to edit NEW/OLD file(s) ----
Se | Line=374 | Col=1 | Alt=0,0;0 | Size=693 | Recl=32752 | Fmt=V | Files=1 | Views=1 |      2012/04/13 10:19:35
```

Figure 106. SELCOPY/i - Compare Files - Output Report - wide-screen.

For **LIST=FMT**, the report file is comprised of several different *"mapped"* record-types detailed below.

Record Type: Command

Timestamp
The date and time (yyyy/mm/dd hh:mm:ss) at which the comparison was run.

Command
This is the COMPFILE command syntax generated by the dialog panels that was used to run the compare files utility.

Record Type: Files

Type
The file reference type ("New" or "Old").

Dataset
The fully qualified dataset name (including library member name) or HFS file path.

Record Type: Compare | Compare-*record_type*

zID
This field displays one of the following codes which correspond to the data displayed in the zRecord or *<field1>*, *<field2>*, etc. fields.

(blank)	A record in both the NEW and OLD files that is flagged as being **matched**. Matching record data in these report records are colour coded **BLUE**.
D	An OLD file record that is flagged as having been **deleted** from the NEW file. Deleted records are colour-coded **RED**.
I	A NEW file record that is flagged as having been **inserted**. Inserted records are colour-coded **GREEN**.
CN	A NEW file record flagged as having been changed. Report records with a zID field of "CN" are always followed by a report record with zID field "CO" for the corresponding OLD file record in the record pair. Changed-New records are colour-coded **WHITE**.
CO	An OLD file record flagged as having been changed. Report records with a zID field of "CO" are always preceeded by a report record with zID field "CN" for the coresponding NEW file record in the record pair. Changed-Old records are colour-coded **YELLOW**.
H	Applicable to Basic and Extended Unformatted Compare only, a hilight-changes record follows a CN/CO zID pair of report records, underlining with character "#" (hash) each byte that is different in a record flagged as having been **changed**. Hilight-Changes records are colour-coded **PINK**.

zNewRecNo
The NEW file record number. (Field source is 4-byte binary numeric).
By default, this field is suppressed in a table format view of the report on an 80-character width 3270 terminal.

zOldRecNo
The OLD file record number. (Field source is 4-byte binary numeric).
By default, this field is suppressed in a table format view of the report on an 80-character width 3270 terminal.

zLrecl
The length of the original NEW or OLD file record. (Field source is 2-byte binary numeric).
By default, this field is suppressed in a table format view of the report on an 80-character width 3270 terminal.

zRecord | *<field1> <field2> etc.*
For Basic and Extended Unformatted compare, the zRecord field is displayed containing the original OLD or NEW file record data.

For Formatted and Hierarchical compare, the zRecord field is replaced with each of the fields selected for compare belonging to the record type assigned to the OLD or NEW file record data.

For formatted NEW file records, the contents of these fields are a faithful copy of the original data. For readability, where NEW and OLD structures are not identical, formatted OLD file records are **remapped** to fit the NEW structure, in order to align like named fields in a table view.

```
SELCOPY/i - Browse NBJ2.SELCOPYI.COMPFILE.REPORT using NBJ2.SELCOPYI.COMPFILE.REPORT.SDO     32752 V SEQ
 File Edit Actions Options Utilities Window SwapList Help  WS WR
Command>                                                                                Scroll> Csr
Record type: Command   Fixed(371) Offset=0 Data elements=4
         Timestamp            Command
         <---+----1----+---> <---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9
00000001 2012/04/13 09:58:08 CompFile CBL.FDD.DATA(NEW) using cbl.cbli.SDO(FDDNEW) CBL.FDD.DATA(OLD) using cbl.cbli.SDO

Record type: Files   Variable(6,23) Offset=0 Data elements=4
         Type Dataset
         <->  <---+----1----+->
00000002 New  CBL.FDD.DATA(NEW)
00000003 Old  CBL.FDD.DATA(OLD)

Record type: Compare-YYY-S-03   Fixed(53) Offset=0 Data elements=13
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-03 YYY-S03-F1 YYY-S03-F2 YYY-S03-F3
         <>  <---+---->  <---+---->  <--> <---+-->      -          <->        <---+----1----+----2----+-->
00000004 I   0000000004                40 STR-0003      ?          ABC        TEXT IN THE LENGTH 40 xxxxxx

Record type: Compare-YYY-S-02   Fixed(46) Offset=0 Data elements=12
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-02 YYY-S02-F1             YYY-S02-F2
         <>  <---+---->  <---+---->  <--> <---+-->      <---+----1----+----2-> <->
00000005 I   0000000005                33 STR-0002      SAMPLECARDFORTEST12345 NUM
00000006 I   0000000006                33 STR-0002      SAMPLECARDFORTEST44444 NUM
00000007 I   0000000007                33 STR-0002      SAMPLECARDFORTEST55555 NUM
00000008 D                0000000005   33 STR-0002

Record type: Compare-YYY-S-03   Fixed(53) Offset=0 Data elements=13
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-03 YYY-S03-F1 YYY-S03-F2 YYY-S03-F3
         <>  <---+---->  <---+---->  <--> <---+-->      -          <->        <---+----1----+----2----+-->
00000009 D                0000000006   40 STR-0003

Record type: Compare-YYY-S-02   Fixed(46) Offset=0 Data elements=12
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-02 YYY-S02-F1             YYY-S02-F2
         <>  <---+---->  <---+---->  <--> <---+-->      <---+----1----+----2-> <->
00000010 D                0000000007   33 STR-0002
00000011 D                0000000008   33 STR-0002

Record type: Compare-YYY-S-03   Fixed(53) Offset=0 Data elements=13
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-03 YYY-S03-F1 YYY-S03-F2 YYY-S03-F3
         <>  <---+---->  <---+---->  <--> <---+-->      -          <->        <---+----1----+----2----+-->
00000012 D                0000000009   40 STR-0003

Record type: Compare-YYY-S-02   Fixed(46) Offset=0 Data elements=12
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-02 YYY-S02-F1             YYY-S02-F2
         <>  <---+---->  <---+---->  <--> <---+-->      <---+----1----+----2-> <->
00000013 I   0000000010                33 STR-0002      SAMPLECARDFORTESTABCDE num
00000014 I   0000000011                33 STR-0002      SAMPLECARDFORTEST55555 NUM

Record type: Compare-YYY-S-03   Fixed(53) Offset=0 Data elements=13
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-03 YYY-S03-F1 YYY-S03-F2 YYY-S03-F3
         <>  <---+---->  <---+---->  <--> <---+-->      -          <->        <---+----1----+----2----+-->
00000015 I   0000000012                40 STR-0003      ?          ABC        TEXT IN THE LENGTH 40 XXXXXX
00000016 I   0000000013                40 STR-0003      ?          DEF        TEXT IN THE LENGTH 40 XXXXXX

Record type: Compare-YYY-S-01   Fixed(41) Offset=0 Data elements=13
         zId  zNewRecNo  zOldRecNo zLrecl YYY-STRUCT-01 YYY-S01-F1 YYY-S01-FN YYY-S01-F2
Se | Line=1 | Col=1 | Alt=0,0;0 | Size=31 | Recl=32752 | Fmt=V | Files=1 | Views=1 |      2012/04/13 10:00:56
```

Figure 107. SELCOPY/i - Compare Files - Output Report - formatted compare.

Record Type: Field

Report records of this type are only included for Formatted or Hierarchical compare only.

zID
This field displays one of the following codes which corresponds to the field data displayed in other report record types.

C	This Field report record identifies a **changed field** in the preceeding Compare-*record_type* report record. A separate Field report record is written for each changed field.
K	This Field report record identifies a **key field segment** in a record type. A separate Field report record is written for each key field segment in each keyed record type. If key segments are specified using absolute position and length (instead of by field name) then Key report record types are displayed instead. Field report records identifying key field segments are displayed following the Summary report records.
S	This Field report record identifies a **field specifically selected for compare** in a record type. If fields in a record type have **not** been specifically selected for compare (COMPFILE syntax SELECT *field* FROM *record_type*), then all fields in the NEW structure record type are selected by default and no "S" Field report record is generated for that record type. A Field report record is written for each field within a record type which has been selected for compare. Field report records identifying compare fields are displayed following the Summary report records.

zRecType
Identifies the record type name.

zFieldName
Identifies the field name within the record type specified by zRecType.

If the mismatching field is an element of an array (e.g. a COBOL OCCURS DEPENDING field), then the field name includes a parenthesised, numeric subscript identifying that field element.

The CFDIFF operation (assigned to F18 by default) opens a new view of the data to display only mismatched fields belonging to the focus record pair. Where a mismatch has occurred within an array, only those elements for which a mismatch has occurred are included in the new window view.

The CFSCROLL operation (assigned to F19 by default) may be used to scrolls the display of a mismatching pair of records horizontally to the mismatching field name identified by a subsequent "Field" report record. i.e. Position the cursor on the required field name in the "Field" report lines and press F19.

Record Type: Summary

This report record type provides a report summary of the compare files execution. Only one Summary record exists for any compare files execution.

SyncType
This field displays the record pair synchronisation technique used for the compare operation. (**Read-Ahead**, **One-to-One**, **Keyed** or **Keyed (unsorted)**)

NewRecsTot
The total number of records processed from the NEW file.

OldRecsTot
The total number of records processed from the OLD file.

Matches
The total number of NEW and OLD file records that match.

Changed
The total number of changed records.

Deleted
The total number of OLD file records deleted.

Inserted
The total number of NEW file records inserted.

NewNotSel
For a formatted compare only, the total number of records from the NEW file that were not selected for comparison because they did not fit any record type selection criteria and so were not assigned a record type.

OldNotSel
For a formatted compare only, the total number of records from the OLD file that were not selected for comparison because they did not fit any record type selection criteria and so were not assigned a record type.

Record Type: Key

This report record type displays information about a key specified as absolute record key positions and lengths. A Key report record is displayed for each segment of the key defined this way.

`Length`
The Key segment length.

`NewKeyPos`
The Key segment position in the NEW file.

`OldKeyPos`
The Key segment position in the OLD file.

Record Type: Gap

This report record type is included for formatted compares only, as a visual separation between CONTEXT blocks, and comprises a single field only.

`zID`
Set to " -".

Function Keys

<F1>	Display context sensitive help.
<F2>	Display the report record in a new window in single format (vertical) view. In single format view, use **<PF10>**/**<PF11>** to display the previous/next report record respectively.
<F16>	Display the **CFUTIL** compare files report, multi-function menu. This includes show and hide of report records based on their type, and show **Changed fields only** in a separate window.
<F6>	Applicable to report records of record type Compare or Compare-*record_type* only, <PF6> edits the file(s) referenced by the focus report record. The SELCOPY/i text editor is used to edit the file, and the display is scrolled directly to the record number referenced in the report focus record. If the focus is a **Matched** record, then both the OLD and NEW files will be placed in the edit ring, with focus passed to the NEW file.

Compare Libraries (=7.2)

Overview

The Compare Libraries (COMPLIB) utility provides a method of performing an unformatted, 1-to-1 compare of records belonging to selected members of two (NEW and OLD) PDS/PDSE libraries. All record data in selected members of the NEW library are compared with members of the same name in the OLD library.

If NEW/OLD library definition is supplied as a DD name that refers to a concatenation of libraries, then the multiple libraries will be treated as a member directory search path. i.e. for each matching member name, the first located along the NEW search path will be compared with the first located along the OLD search path, and no other occurences of the same member further down the search path will be processed.

The compare process utilises the SELCOPY program with input control statements being read from member ZZSCOMPL of the distributed sample library SZZSSAM1. The Compare Libraries utility can execute SELCOPY in the foreground (interactively) or may be used to generate JCL to execute the SELCOPY job in batch.

The SELCOPY report output is written to SYSPRINT so that each report line details the result of the compare between one NEW and OLD library member pair, and also includes a COMPFILE command to compare the two library members individually. When the SYSPRINT output is displayed in a SELCOPY/i text edit view, the generated COMPFILE commands may be executed using the ACTION facility (i.e simply position the cursor on the required COMPFILE command and press the <F16> key.

If the Compare Libraries utility is executed in the foreground, then the result of each member compare is logged to the terminal and the SYSPRINT output presented to the user in a temporary text edit view.

Compare Libraries Panel

The Compare Library Members utility panel window (ZZSCOMPL) is an interactive panel window (window class WINWIPO0) and may be started via the following:

- Select 'Compare Libraries' from the Utilities menu.
- Execute the command COMPLIB with no parameters from the command line of any window.
- Execute the prefix command "**CL**" against a PDS/PDSE library DSN entry of a file List type window. The resulting Compare Library Members panel window will treat the corresponding list entry as the New DSN.

By default, field entries are populated with arguments and options that were entered the last time the Compare Library Members Utility panel was used.

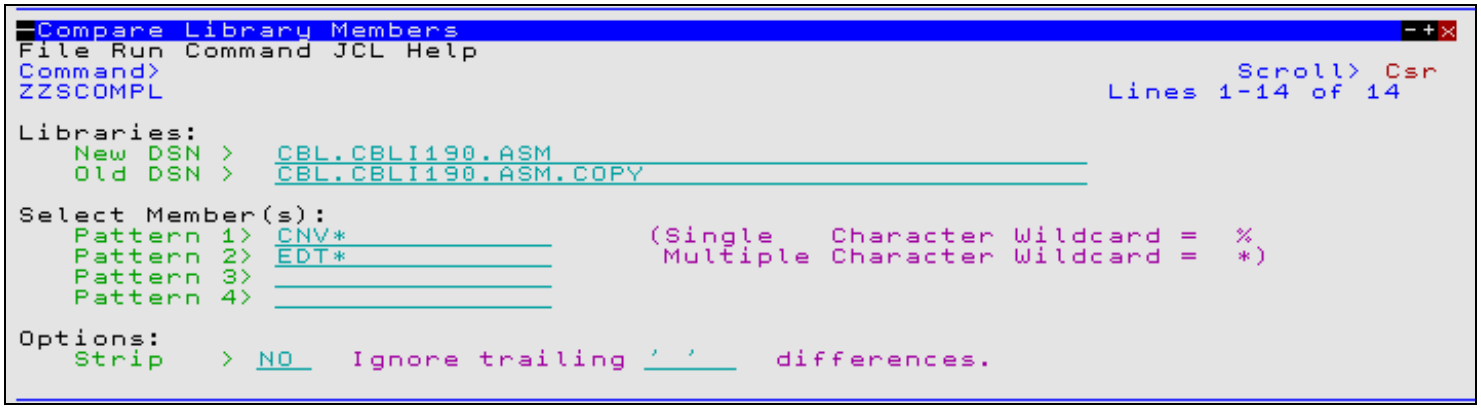


Figure 108. Compare Library Members Panel.

Having typed entries in the required panel fields, simply pressing the <Enter> key will action the library compare in the foreground.

Alternatively, the user may select an item from the menu bar.

Menu Bar Items

Run
Run the library compare in the foreground.

Command
Generate the COMPLIB command line syntax for field entries specified by the user and display it in a temporary CMX file text edit view. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

JCL
Generate a JCL job stream that executes the **SDEAMAIN** program with input (SDEIN) containing the COMPLIB command determined from the specified panel field values.

The job stream is displayed in a temporary text edit view and may be submitted to batch using the SUBMIT command.

StaticJCL
Generate a JCL job stream that executes the **SELCOPY** program. The SYSIN input comprises the SELCOPY control statements member ZZSCOMPL and a list of member names that match the specified member patterns.

The job stream is displayed in a temporary text edit view and may be submitted to batch using the SUBMIT command.

Panel Input Fields

Libraries:
New Lib>
This input field is mandatory and identifies the new version of the PDS or PDSE library or libraries to be compared. As an alternative to a fully qualified library dataset name, a DDName which is allocated to a PDS/PDSE library concatenation may be specified. If so, the compare process ignores all but the first occurrence of each member found along the libary search path.

For an uncataloged library, the volume name on which the data is stored must be specified in the accompanying **Volume>** input field.

Select Member(s):
Pattern 1/2/3/4>
These input fields are optional and allow the user to provide up to 4 alternative member name masks for selecting members to be compared.

A member name mask supports the following wild cards:

`*` A single asterisk represents an entire member name or zero or more characters within a member name mask.

`%` A single percent sign represents exactly one character within a member name mask. Up to 8 percent signs can be specified in each member name mask.

If no member name masks are specified, then all members of the NEW **and** OLD libraries will be compared.

`Options:`

`Strip>`
This input field (STRIP) contains either "YES" or "NO" and indicates whether trailing characters that match the specified strip character, are to be stripped from the longer record, to the length of the shorter record, when the records to be compared are of different lengths.

`Ignore trailing '`*`char`*`' differences.`
This input field (STRIPC) specifies the strip character to be used if the STRIP field conmtains "YES".

`Report:`

`Include Matched`
Select this option to report members that exist in both NEW and OLD library and whose contents match.

`Include Missing New`
Select this option to report members that exist in the OLD library but are missing from the NEW library.

Performance may be improved by deselecting this option, since once all selected members from the NEW library have been processed then Compare Libraries may terminate without processing any remaining members of the OLD library.

`Include Missing Old`
Select this option to report members that exist in the NEW library but are missing from the OLD library.

Performance may be improved by deselecting this option, since once all selected members from the OLD library have been processed then Compare Libraries may terminate without processing any remaining members of the NEW library.

`CompFile Options:`

`Sync>`
Defines the synchronisation type to be used on the generated CompFile command. Select either **Read-Ahead** or **1-to-1**.

Read-Ahead uses default read ahead limit value of 200 records.

`Limit>`
Defines the LIMIT parameter to be used on the generated CompFile command.

Use this option in order terminate the CompFile process as soon as the specified number of record mismatches has been encountered.

Specifying zero or blank indicates that no limit is placed, and therefore the whole of each file is processed.

Note that Compare Libraries itself processes each member only to the point required to establish a single mismatch.

`Context>`
Defines the CONTEXT parameter to be used on the generated CompFile command.

A selection list will be displayed if the option entered is unrecognised or left blank. Choose from one of the following options:

Partial
Show 10 lines before and after each difference in order to provide additional context without displaying all matching records. Each difference "block" will be followed by 2 "gap" records designed to provide visual separation.

This option corresponds to the COMPFILE parameters "CONTEXT 10" and "GAP 2".

For full control over the number of context and gap records select Use Extended Options below.

Full
All corresponding records from both NEW and OLD files that match are to be included in the output report file.

This option corresponds to the COMPFILE parameter "INCMATCHED".

None
No matching records are to be displayed.

Compare Libraries Output

The output generated by the Compare Libraries utility is a report of the compare on each NEW and OLD library member pair for which the data did not match. This report is displayable EBCDIC text and may be viewed using the SELCOPY/i (CBLe) text editor.

Each line of the output report corresponds to a single member compare.

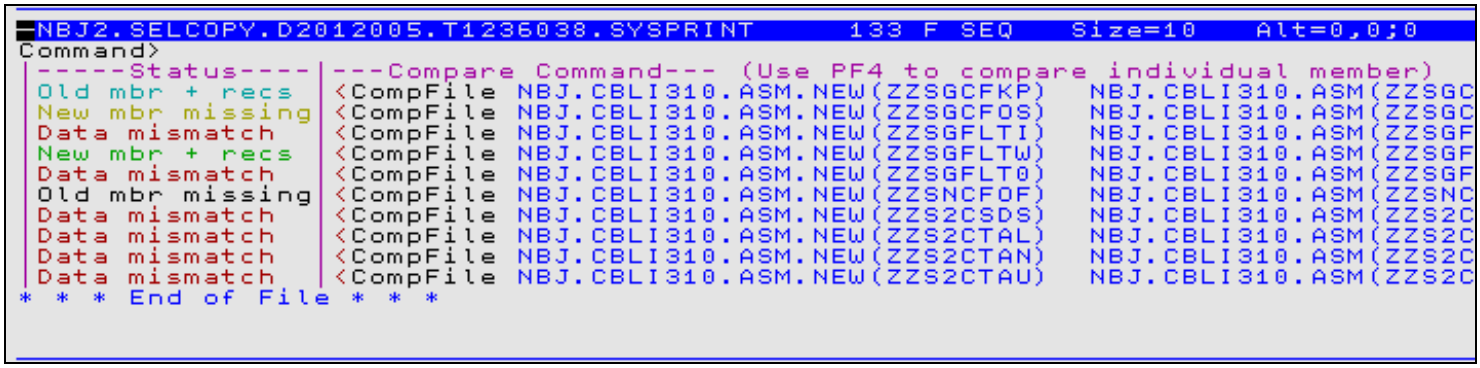


Figure 109. Compare Libraries Output.

Each output report record is comprised of the following fields:

Status

The status identifies the cause of the unsuccessful member data compare.

Data mismatch
A difference was identified in at least one record of the library members.

On encountering a difference in the record data, no further record matching occurs for that member. Note that the members may also contain a different number of records but reporting the data mismatch takes precedence.

Old mbr + recs
Data in the NEW and OLD library member pair matches, however, additional records were found in the OLD library member.

New mbr + recs
Data in the NEW and OLD library member pair matches, however, additional records were found in the NEW library member.

Old mbr missing
No member exists in the OLD library that matches the NEW library member name.

New mbr missing
No member exists in the NEW library that matches the OLD library member name.

Note that, if a member mask has been used to select a subset of members from the NEW library, then this report status will never occur.

Compare Command

The Compare Command field contains a COMPFILE command which may be executed to generate a more detailed report of the differences that exist between the pair of library members. COMPFILE involes the Compare Files utility.

This command syntax also identifies the NEW and OLD library members of the same member name to which the report line refers.

To execute the COMPFILE command directly from the edited report, simply position the cursor on the command and press the <F16> key.

SELCOPY Debug & Development (=8.1)

Procedures written in the SELCOPY or SLC batch language may be executed interactively for debug using the SELCOPY Debug application. This takes full advantage of SELCOPYi application windows and Text Editor features in order to assist with debug operations.

SELCOPY source code is written in the Basic Assembly Language (BAL) whereas the SLC program, which also executes SELCOPY control statements, is written using the C++ language.

Note that SLC was first developed to provide a version of SELCOPY that executes on Microsoft Windows, Linux and various Unix platforms. Therefore, the control statement syntax for both programs are very similar although not identical. Since its inception, this C++ version of SELCOPY has introduced many new facilities that are not available in the BAL version. So that programmers on mainframe systems could take advantage of these new facilities, the C++ version was compiled for z/OS and z/VM CMS systems and included as the executable load module, SLC, in the SELCOPY Product Suite package.

Although the SELCOPY (BAL) program remains the default SELCOPY language interpreter for legacy programs, developers are encouraged to explore and try the additional facilities offered by SLC in new applications. Note, however, that SLC does have some limitations, in particular DB2, IMS/DL1 and ADABAS data base processing is not yet supported.

To assist development of procedures written for either program, both the *"SELCOPY Manual"* and *"SELCOPY C++ (SLC) Language Reference"* documents are included in the SELCOPYi on-line help system.

The SELCOPY Debug application is a special instance of the Text Editor (CBLe) and always opens as an MDI frame window within the SELCOPYi main window display area (i.e not as a child window of the existing Text Editor main window). To switch between the two Text Editor applications, execute primary command NEXTMAINWINDOW.

See SELCOPY Debug Main Window for information on the SELCOPY Debug window environment.

SELCOPY Debug Startup

SELCOPY Debug is started via the following:

- Select option 1. 'SELCOPY/debug' from the Utilities Menu (z/OS only).
- Select 'SELCOPY Debug/Dev' from the File menu in the SELCOPY Text Editor main window menu bar.
- Execute the primary command SELCOPY (minimum abbreviation, SELC).

Unless primary command SELCOPY has been executed with parameters, in which case the application is started directly, z/OS users are first presented with a panel which prompts for the input type. Options are as follows:

1. Supply JCL (=8.1.1)
2. Supply SYSIN (=8.1.2)

Invocation via supplied JCL is not supported for z/VM and z/VSE versions of SELCOPYi.

Supply JCL (=8.1.1)

JCL Job Input Panel

This panel allows the user to specify the name of an existing JCL dataset that contains one or more SLC (PGM=SLC) and/or SELCOPY (PGM=SELCOPY) job steps which are to be executed interactively by the SELCOPY Debug application.

The advantage of invoking SELCOPY Debug from JCL input is that it encapsulates all the input/output file definitions necessary to run each step.

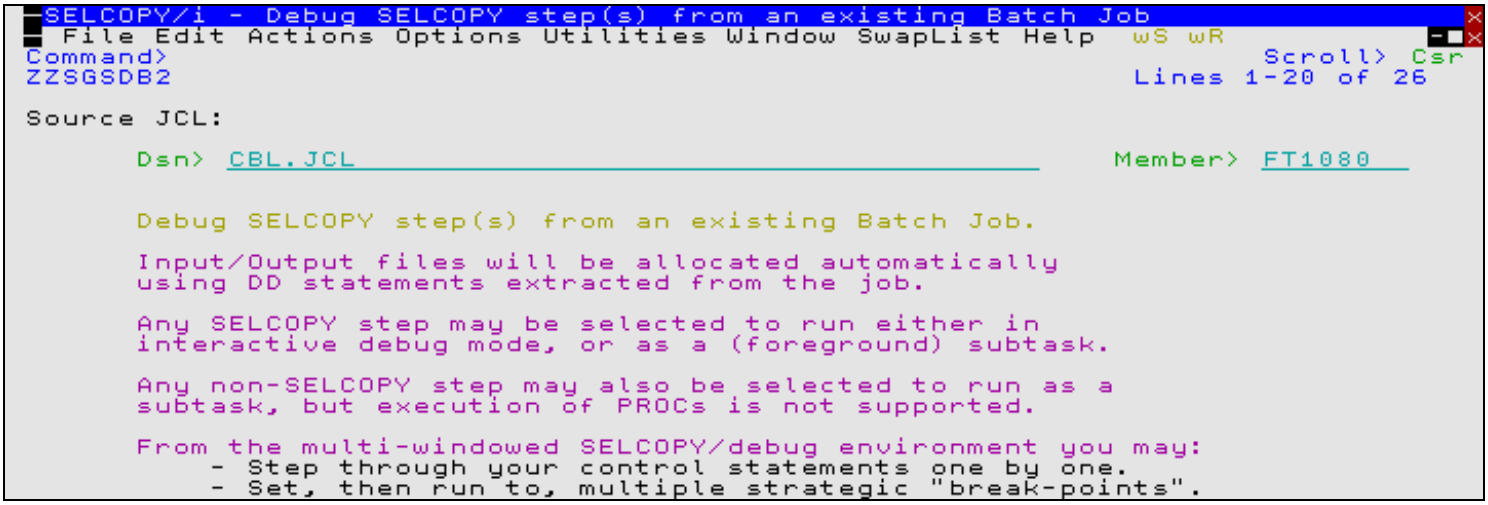


*Figure 110.*SELCOPY/Debug JCL Batch Job Input.

Having selected the source JCL dataset or library member, SELCOPYi performs a rudimentary parse of the supplied JCL, generating temporary work datasets and an executable Rexx procedure for each job step. Note that JCL EXEC statements that execute a procedure (PROC) are not currently supported by the SELCOPY Debug JCL parser and so work datasets will not be generated for these job steps.

By default, the work dataset names have the prefix *hlq*.SELCDBUG.*jobname* where *hlq* is the user's own SELCOPYi DSN prefix (as defined by User INI variable System.UserDSNPrefix), and *jobname* is the job name specified on the JCL JOB statement. If the length of a work dataset name exceeds 44 characters, the qualifier SELCDBUG may be abbreviated to SDB.

Primary command JCL (assigned to F5 by default) will open the specified source JCL dataset or library member in a Text Editor window edit view.

Once parsing has completed, the Job Step Selection List panel is displayed containing a list of the job steps in the order in which they were found in the source JCL statements.

Panel Input Fields

`Source JCL:`
Fields which together indentify the JCL source input.

`Dsn>`
Identifies the fully qualified dataset name of a sequential dataset or PDS/PDSE library containing z/OS batch Job Control.

A selectable list of datasets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

`Member>`
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

JCL Job Step Selection List Panel

The Job Step Selection List panel displays a list of job steps extracted from the supplied batch job.

The job step list entries may be selected or deselected so that, on hitting the <Enter> key, only the selected job steps will be executed in order in which they appear in the list. Entries are selected by entering "S" in the **Sel** list column.

If selected, any step that does not execute program name SELCOPY or SLC will be run as a (foreground) subtask of SELCOPYi. Each selected job step entry will be executed without returning control to the user. However, if "Y" is entered in the **Debug** list column of a job step entry that executes program name SELCOPY or SLC, then, when it is that job step's turn to be executed, the SELCOPY Debug application is opened and execution of the remaining job steps is paused. Interactive debug of the SYSIN control statements may then be performed for that job step before returning to the Job Step Selection list when the SELCOPY Debug application is closed. If "Y" is not entered in the **Debug** list column, then the SELCOPY or SLC job step is run as a subtask.

On returning to the Job Step Selection list following exit of SELCOPY Debug, execution of the unprocessed job steps (flagged with pending status) may be continued by once again hitting the <Enter> key.

Note that any job step that executes a cataloged or in-line procedure is flagged with "PROC not executable" in the **Info** column. An attempt to execute this job step will set a return code of -22222.

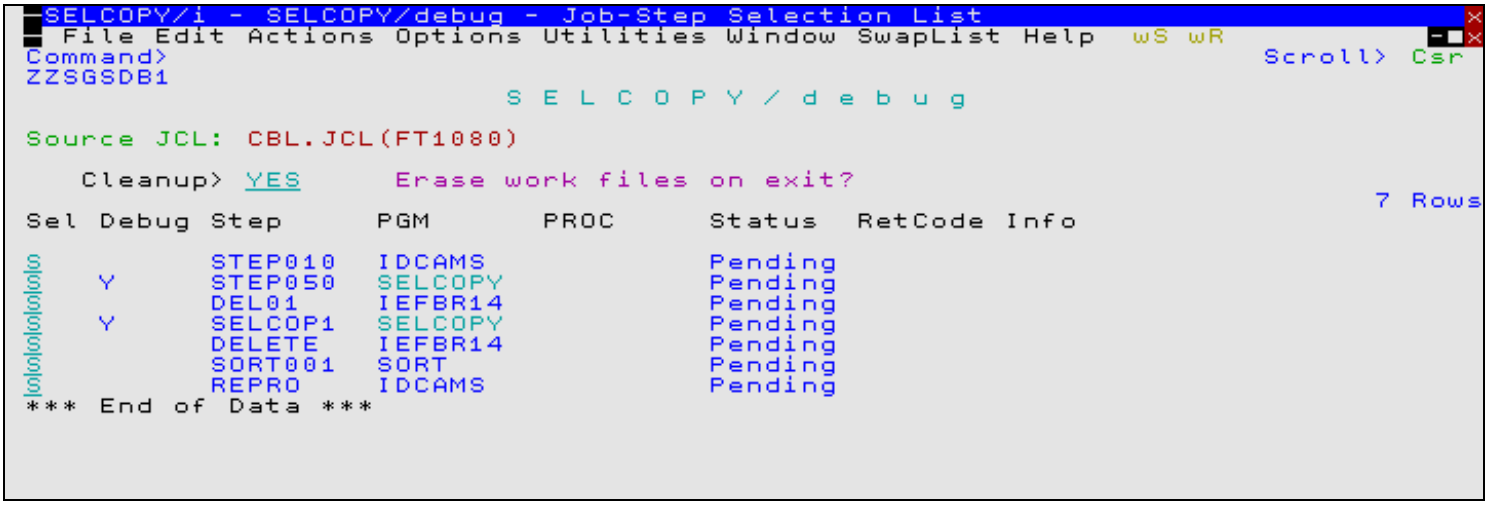


*Figure 111.*SELCOPY/Debug - Job Step Selection List.

Panel Input Fields

Source JCL:
A non-enterable field displaying the JCL source input dataset or library member name.

Cleanup>
Select "YES" or "NO" to indicate whether or not work datasets created for this JCL batch job are to be erased when the panel is closed.

If debug of the JCL input is to be actioned again after the job step list panel has been closed (e.g. in a future SELCOPYi session) and the batch job is not changed, then selecting "NO" to keep the work datasets will bypass the need to re-parse the JCL syntax and so improve startup performance. Note that, if the work datasets already exist for a particular batch job when the job step list panel is opened, the user is prompted to use these work datasets or parse the JCL to re-create them.

Sel
Enter "S" (or any non-blank value) in this field to select the entry for processing. If left blank, processing for the job step identified by this list entry is bypassed.

Debug
Applicable only to job steps that execute program SELCOPY or SLC, enter "Y" in this field to select the job step entry for interactive execution via the SELCOPY Debug application. If left blank, execution of the job step identified by this list entry will be performed as a sub-task of SELCOPYi.

Step
A non-enterable field identifying the name of the job step. If no name was provided for an EXEC statement in the JCL source, then one is generated as STEP*nn* (where *nn* is the job step number). Note that, if a duplicate step name was found during JCL parsing, the user will have been prompted to enter an alternative name.

PGM
A non-enterable field identifying the program name found on a PGM= parameter on an EXEC statement.

PROC
A non-enterable field identifying a procedure name found on an EXEC statement.

Note that, since execution of a procedure is not yet supported using this foreground job execution facility, the presence of a procedure name will be accompanied by the message "PROC not executable" message message in the **Info** column.

Status
A non-enterable field displaying the current status of the job step entry. This may be one of the following:

Value	Description
Pending	The job step has not yet been executed.
Running	The job step is running.
Ended	The job step execution has ended. The return code value is updated in the **RetCode** column.

RetCode
A non-enterable field identifying the return code set by the last execution of the job step.

Info
A non-enterable field containing informational messages relating to the job step and its execution.

JCL Job Panel Primary Commands

The following primary commands are supported:

JCL

```
>>---- JCL ------------------------------------------------------------><
```

JCL will open the source JCL dataset or library member in a Text Editor window edit view and is assigned to <F5> by default.

REFRESH

```
>>---- REFRESH ---------------------------------------------------------><
```

Supported by the Job Step Selection list panel only, REFRESH will re-parse the JCL input, re-generate the work datasets and refresh the job step list.

WORKF

```
>>---- WORKF -----------------------------------------------------------><
```

Supported by the Job Step Selection list panel only, WORKF will open a dataset list displaying the work datasets generated for the input JCL source and is assigned to <F6> by default.

Supply SYSIN (=8.1.2)

Control Statement Dataset Input Panel

The Control Statement Input Dataset panel allows specification of options necessary for execution of the SELCOPY/Debug application using a single, user specified SELCOPY or SLC control statement SYSIN input file (sequential dataset, HFS file path or library member).

Any input or output file name (DD/DLBL/FILEDEF name) referenced within the control statements that is not dynamically allocated to a dataset by a SELCOPY I/O operation, must be specifically allocated by the user. This excludes SYSIN and SYSPRINT which are handled by SELCOPY Debug window management. See the Text Editor ALLOCATE command for dataset allocation within SELCOPYi.

File name references that are dynamically allocated within the SELCOPY control statements do not require specific allocation. e.g.

```
READ  INDD  DSN='MY.INPUT.DATASET.NAME'
WRITE OUTDD DSN='MY.OUTPUT.DATASET.NAME'
```

If the specified control statement file is empty or does not yet exist, then SELCOPY Debug opens an empty SYSIN/SYSIPT Text Editor view and, on performing its initial control statement analysis, reports ERROR 14 "NO INPUT FILE" in the SYSPRINT/SYSLST view with pop-up message:

```
SDB002E SELCOPY has ended with control card errors. Return Code 52.
```

Having selected OK to continue, the user may proceed by adding SELCOPY control statement records to the control file SYSIN/SYSIPT window. When complete, the changes to the control file should be saved before executing RERUN to restart debug of the new control statements. If the control statment file does not already exist, then, for z/OS systems, the Allocate NonVSAM dialog will be opened before a save is actioned.

Note that interactive execution of SELCOPY or SLC using statement stepping and/or break points is not supported if the control statement file contains the SELCOPY option **NOPRINT**, **NOP** or **NOPCTL** to suppress print of the SELCOPY control statements. If any of these options are specified prior to the first control statement, then the job will run to completion without stopping.

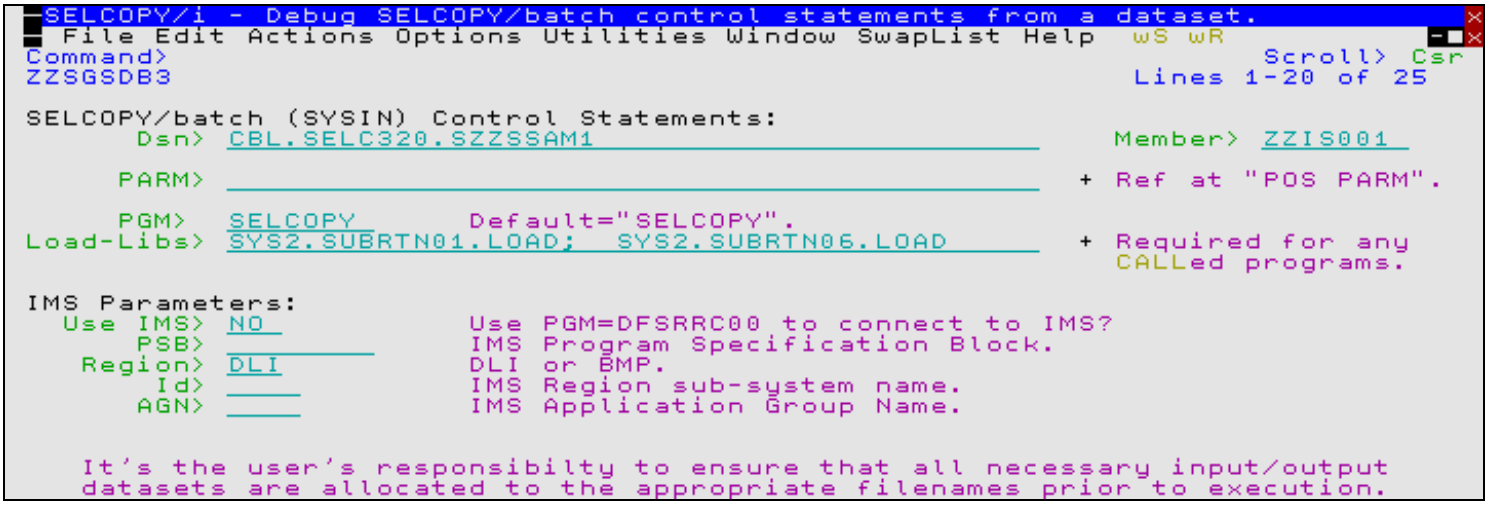


*Figure 112.*SELCOPY/Debug - Control Statement Dataset Input.

Panel Input Fields

SELCOPY/SLC:
Input fields which together identify the program to be used to interpret and run the SELCOPY control statements.

Version>
Specifies "SELCOPY" or "SLC" and identifies the SELCOPY language interpreter to be used.

SLC identifies the program source as being for the SELCOPY C++ version and implies a default program name of SLC. SELCOPY identifies the program source as being for the SELCOPY Assembler (BAL) version and implies a default program name of SELCOPY.

Pgm>
Specifies the name of the program load module to be used to process the control statements. This program name is used in place of the default program name implied by the selected SELCOPY language interpreter (SELCOPY or SLC).

For example, SLC320 may exist as the name of the SLC version 3.20 program in your installation's version 3.30 load library. To execute SLC320 instead of SLC (the 3.30 version), SLC330 must be entered in this field with SLC in the **Version>** field.

Control Statement (SYSIN) Input:
Input fields which together identify the input SYSIN file.

Dsn>
Identifies the fully qualified data set name of a sequential data set or PDS/PDSE library containing SELCOPY control statements.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

PARM>
Specify optional parameter data to be passed to SELCOPY which, in JCL terms, is the equivalent of coding:

```
//STEP1  EXEC PGM=SELCOPY,PARM='MY PARM DATA'
```

PARM data may be accessed at SELCOPY run time by referring to **POS PARM** within your SELCOPY control statements. The length of the data passed on the PARM field is provided as a two byte binary field at **POS PARM-2**. e.g.

```
//STEP1  EXEC PGM=SELCOPY,PARM='MY PARM DATA'
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  option worklen=222
  @PLEN =   2 at PARM-2  type=b   * Set @PLEN var to length of parm data.
  print from pos PARM  len=@PLEN  * Print data passed as PARM='xxxx'.
/*
```

Load-Libs>
The location of the SELCOPY or SLC program to be executed and any routines called by the SELCOPY CALL operation is determined by the standard search chain for the current environment.

Note: The SELCOPY CALL operation is used to pass control to an external Assembler or COBOL routine or any z/OS program module developed using Language Environment.

For z/OS systems only, SELCOPY Debug provides users with the ability to include additional libraries to the start of the search chain. This gives the SELCOPY Debug environment an equivalent to the STEPLIB JCL statement, which may occur in SELCOPY batch jobs.

The included library path may be entered in the "Load-Libs>" field as one of the following:

- A single DDname *libdd* which has been allocated to one or more load libraries.
- One or more load library DSNs *libdsn* separated by ',' (comma), ';' (semi-colon) or blank characters. Note that if blank separators are used, the list of DSNs must be enclosed in '( )' (parentheses), "" (quotation marks) or "'" (apostrophes).

IMS Parameters:

The SELCOPY Assembler program supports data management calls to IMS/DL1 data bases to perform segment I/O operations for offline (DLI) processing or via an IMS online batch message processing (BMP) region. Currently, the SLC program does not support calls to IMS/DL1.

If the SELCOPY control statement input includes operations that perform IMS/DL1 database I/O, then the SELCOPY program must be started as a subtask of the IMS/DL1 region controller (DFSRRC00).

Input fields that follow, identify whether or not SELCOPY is run as a subtask of the IMS/DL1 region controller and the parameters that will be passed to the DFSRRC00 program.

Use IMS>
Specifies whether or not the SELCOPY control statements perform IMS/DL1 calls and so must be called by the IMS/DL1 region controller. If this value is set to "YES", the remaining IMS/DL1 input field values will be implemented, otherwise they are ignored.

PSB>
The Program Specification Block (PSB) containing the Program Control Blocks (PCB) which process the required IMS database(s).

Region>
The IMS region type. "DLI" for offline batch or "BMP" for IMS online batch message processing.

Id>
The IMS region sub-system name identifier.

This value will override the identifier specified during system definition of the running IMS system. The IMS identifier forms part of IMS messages that are written to the system log.

AGN>
The IMS Application Group Name. This value is only used by the batch message processing (BMP) region in IMS v9.1 and earlier where Security Maintenance Utility based AGN security is supported.

Later releases of IMS use Resource Access Security (RAS) and/or the DFSRAS00 user exit to perform application group name authorisation.

CKPTID>
An IMS check point id that may be used specifically by SELCOPY procedures that execute CALL ASMTDLI to perform a DLI extended restart (XRST) operation.

SELCOPY Debug Load Library Search Chain

The location of the SELCOPY or SLC program to be executed and any routines called using the SELCOPY CALL operation is determined by the standard search chain for the current environment.

Note: The SELCOPY CALL operation is used to pass control to an external Assembler or COBOL routine or any z/OS program module developed using Language Environment.

For z/OS systems only, SELCOPY Debug provides users with the ability to include additional libraries to the start of the search chain. This gives the SELCOPY Debug environment an equivalent to the STEPLIB JCL statement, which may occur in batch jobs.

The included library path may be entered in the Control Statements from a Dataset panel or via the -LIBRARY parameter on the SELCOPY SELCOPY primary command, as one of the following:

- A DDname which has been pre-allocated to one or more load libraries.
- One or more load library DSNs each separated by a ',' (comma), ';' (semi-colons) or a blank character. If the DSNs are separated by blanks, quotes or '( )' parentheses must be used to delimit the list of DSNs, not the individual DSNs.

The following SELCOPY line command illustrates use of a library path:

```
SELCOPY  -CTL CBL.SELC320.SZZSSAM1(ZZIS001)  -LIB SYS2.SUBRTN01.LOAD; SYS2.SUBRTN06.LOAD
```

SELCOPY Debug IMS/DL1

The SELCOPY program supports data management calls to IMS/DL1 data bases to perform segment I/O operations for offline DLI processing or via an IMS online batch message processing (BMP) region. The SLC program does not yet support IMS/DL1 processing.

When SELCOPY control statements that perform IMS/DL1 database I/O are executed, the SELCOPY program is started under the IMS/DL1 region controller (DFSRRC00). When run in SELCOPY Debug, a PSB name and region type (DLI or BMP) must be passed as parameters to DFSRRC00. Other optional DFSRRC00 parameters and arguments that are supported by SELCOPY debug are as follows:

CKPTID	The 1-4 character checkpoint ID that exists in IMSLOGR. Applicable only to SELCOPY control statements that execute a CALL to the DL/1 XRST (extended restart) function to restart the SELCOPY processing from a check point. (See **IMS/DL1 Restart** below.)
IMSID	The 1-4 character IMS identifier that will be used in IMS messages that written to the system log. It overrides the identifier specified at the time of system definition of the running IMS system.
AGN	The 1-4 character application group name required to access IMS online data (i.e. via BMP). This parameter may only be valid in versions of IMS up to V9.1. Later releases of IMS no longer support Security Maintenance utility and AGN security but Resource Access Security (RAS) and/or the DFSRAS00 user exit is used instead to perform application group name authorisation.

The SELCOPY Debug application intercepts all SELCOPY DL/I calls with IMS DB and executes the call itself using the ASMTDLI assembler interface called from a sub-task of SELCOPYi.

Before the SELCOPY Debug session can be started, the IMS program load libraries (e.g. IMS.SDFSRESL) should be included in the program search path, specified on the -lib parameter of the SELCOPY primary command or specified in the Load-Libs field of the SELCOPY/Debug panel. Additionally, the following DDnames required for successful IMS/DL1 batch execution must also be allocated.

DFSRESLB	The IMS resource library concatenation. The libraries contain the IMS SVC modules and must be APF authorised. This DDname is necessary only for offline (DLI) batch processing. If DDname DFSRESLB is not allocated, SELCOPY Debug will attempt to dynamically allocate DFSRESLB to the single library DSN specified by the SELCOPYi INI option **DLI.RESLIB**.
DFSVSAMP	The dataset containing the VSAM buffer subpool definitions. This DDname is necessary only for offline (DLI) batch processing.
IEFRDER	The primary system log datasets. This DDname is necessary only for offline (DLI) batch processing where database-update intent is also declared. IEFRDER may be allocated to DUMMY unless the SELCOPY procedure performs a CALL to execute a DL/1 extended CHKP and the output checkpoint records are potentially to be used for extended restart (XRST) IMSLOGR input.
IMS	The concatenation of required PSB and, if necessary, DBD libraries. These libraries must include the PSB specified to DFSRRC00. This DDname is necessary only for offline (DLI) batch processing and optional for online (BMP) processing. If DDname IMS is not allocated, SELCOPY Debug will attempt to dynamically allocate IMS to the single library DSNs specified by the SELCOPYi INI options **DLI.PSBLIB** and **DLI.DBDLIB**.
IMSLOGR	The input log data set for extended restart. This DDname is only necessary for offline (DLI) batch processing where an extended restart (XRST) call is performed by the SELCOPY control statements. IMSLOGR is not necessary if XRST is called for an online region (BMP) and the checkpoint records required to restart the BMP exist in the online log data sets (OLDS).

IMS/DL1 Restart

SELCOPY does not have native support for extended check point (CHKP) and restart (XRST) which is required to restart the SELCOPY program from a check point. Extended restart will recover user buffers (e.g. the SELCOPY workarea) and reposition database pointers to their status at the time an extended check point was performed.

Although not supported natively, the SELCOPY CALL operation may be used to call the ASMTDLI load module with a parameter list that executes the call to extended CHKP or XRST. Doing this, it is possible to write a re-startable SELCOPY procedure. See the CBL SELCOPY success story at the following URL for an example:

```
http://www.cbl.com/success.php#tab=finance
```

To restart a program from a check point, the check point id must be passed on the call to XRST. This is achieved when SELCOPY Debug is started by specifying the value in the CKPTID> input field of the SELCOPY/Debug - Control Statement Dataset Input panel or on the -CKPTID parameter of the SELCOPY primary command.

SELCOPY Debug Loop Break-in

The nature of SELCOPY and SLC execution is such that statements are executed sequentially, or as directed by logic flow operations (e.g. GOTO, PERFORM), until either the last control statement of the SYSIN/SYSIPT input is encountered or a GOTO GET operation is executed.

When one of these conditions occur and at least one input (e.g. READ) and one output (e.g. WRITE, PRINT, UPDATE) operation exists, then processing is passed back to the first run-time control statement in the SYSIN input.

This looping through the control statements will continue until one of the following occurs:

1. End-of-File condition is encountered following an attempted READ of the **prime** input file and no IF EOF condition exists for the file.
2. No further output operations are eligable for execution as a result of a explicit or implicit STOPAFT value. e.g. STOPAFT=50 is implied for LOG output operations and STOPAFT=1 is implied for operations executed based on a true IF INCOUNT condition for equality.
3. GOTO EOJ or GOTO CANCEL operation is executed.
4. A Selection Time Error is encountered.

If none of these conditions occur, then it is possible to introduce an infinite loop in SELCOPY control statement processing.

SELCOPY Debug may be used to identify the cause of this situation or any sequence of statements that cause the control statement stream to loop. It is possible, however, that the user may not know that the loop condition exists until SELCOPY processing has been restarted without a break point in which case, since SELCOPY is executing in the foreground, the 3270 session becomes unresponsive.

It is for this reason that the SELCOPY default break-in facility exists to allow the user to pre-define a default number of times that any control statement may be executed before a virtual break point is encountered and processing is paused.

This break-in threshold is initially set by the SELCOPYi INI option **SELCOPY.LoopBreakIn** which has a default value of 1,000. This value may be updated and interrogated using the SELCOPY Debug option BREAKIN.

When the break-in threshold has been reached, a pop-up message window is opened and control is passed back to the user to continue debug investigation. This means that there is no need to forcibly end the SELCOPYi session and restart the SELCOPY debug process.

Note that a loop break-in may occur even though a loop is not infinite. (e.g. the prime input file may have a number of records greater than the break-in threshold.)

SELCOPY Debug Windows

SELCOPY Debug Main window

Like the CBLe text editor, SELCOPY Debug is an MDI (Multiple Document Interface) application. An MDI application comprises a parent (frame) window with a menu bar and a client area within which one or more MDI child windows are displayed. All MDI child windows are confined to the parent window's client area.

The SELCOPY Debug Main (frame) Window supports all MDI child windows supported by the CBLe frame window (including SDE Edit). The SELCOPY Debug frame window is actually a CBLe frame window with additional features and characteristics specifically relating to SELCOPY execution. These features are discussed in this section whereas details on CBLe frame window features may be found in the CBLe Text Edit documentation.

The SELCOPY Debug Main window must always contain the Control Cards and Output Listing. Closing either of these windows will quit the SELCOPY Debug main window and so end the Debug session.

When a session is started, these 2 child windows are automatically opened, together with a work area storage window, at fixed locations within the main window client area. The position and size of each window have been pre-determined so that the contents of each window are easily visible when used with terminals of width greater than 80 bytes. Where the terminal display is of width less than 80 bytes, the SELCOPY Debug child windows are opened in a maximised state, however, these may be subsequently restored, resized and repostioned as in Figure 112.

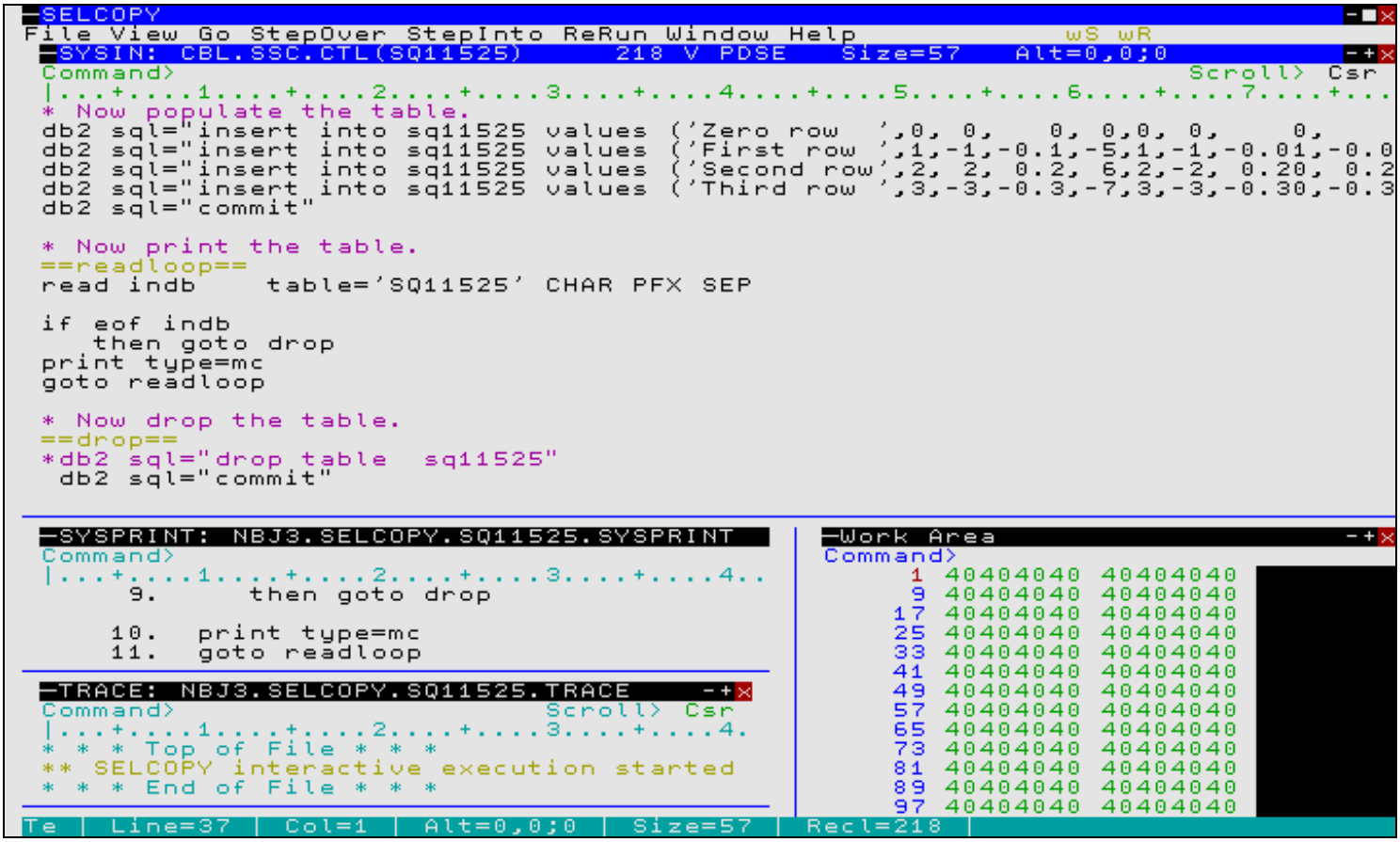


Figure 113. SELCOPY Main Window in 43x80 3270 Session - Resized Child Windows.

Note that the "Ws" (Window Save) button may be used to save a focus child window's size and location within the parent window so that it may subsequently be restored using the "Wr" (Window Restore) button. This enables the user to maintain preferred window size and location across invocations of SELCOPY Debug.

All SELCOPY Debug child windows, other than list and storage windows, are CBLe text edit windows (i.e. Control Cards, Output Listing, trace and log windows.) This allows the user to edit the data in these windows and to issue CBLe commands and macros such as FIND, EXCLUDE, CHANGE and SAVE.

In addition to the standard SELCOPY Debug windows, the user can open a CBLe edit view for any other file (e.g. the input data sets, etc.), thus giving SELCOPY Debug all the features provided by the CBLe text editor. Also, any LIST window opened from a SELCOPY Debug child window will itself be a child window of SELCOPY Debug.

By default, function key F4 is assigned to line command, WINDOW, and is used to pass focus between the SELCOPY Debug child windows.

SYSIN Window

The SYSIN window is opened automatically when SELCOPY Debug is started. It may also be opened via the following:

- Select 'Control Cards' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug primary command WINDOW CTL.

The SYSIN window is an edit view that contains the control statement source file as required for execution of SELCOPY Debug. This window highlights the current operation and allows the user to set and unset break points.

By default, SELCOPY Debug attempts to edit the SYSIN file read/write. If this is not possible, the user is prompted to continue the session with the file edited in read only mode. In either case, the edit profile macro is executed when the file is loaded. If the CBL supplied macro PROFILE is set as the default edit profile, then useful edit buttons are added to the menu bar. See the PROFILE and PROFIRST macros for a description of each button's use.

Note that SELCOPY analyses the control statements prior to execution and it is at this point that SELCOPY Debug associates each operation in the SYSIN display with its appropriate selection id. Therefore, any alterations made to the SYSIN data during SELCOPY debugging must first be saved and the job re-started before any further statement execution can take place.

The contents of the window scroll automatically in order to display the current statement in the SELCOPY execution. As for any edit view, CBLe commands and macros may be used to manipulate, highlight and locate data in the view (e.g. FIND, LOCATE, TAG, ALL, CHANGE, SET ZONE, etc.)

In addition to any CBLe edit highlighting, during the course of execution control statements are highlighted as follow:

1. Next executable SELCOPY statement. Default highlight - pink reverse video.
2. Break Point. Default highlight - red reverse video.

The primary command SDBPOPUP, which in the SYSIN window is assigned to F16 by default, will display the point-and-shoot popup menu. This menu includes a number of executable actions for the valid positional expression located at the focus (cursor) column/row.

Closing the SYSIN window also exits SELCOPY Debug.

SYSPRINT Window

The SYSPRINT window is opened automatically when SELCOPY Debug is started. It may also be opened via the following:

- Select 'Listing' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW LIST.

SELCOPY Debug intercepts output to SYSPRINT/SYSLST and displays it in the SYSPRINT window instead. For this reason, SYSPRINT or SYSLST does not need to be allocated and no output is written to the system spool.

The contents of the SYSPRINT window scroll automatically to display any new output to SYSPRINT/SYSLST. Data written to the SYSPRINT window is maintained until the SELCOPY Debug session is closed. Therefore, so long as the SELCOPY Debug session is not closed, the job may be re-run any number of times without loosing the SYSPRINT/SYSLST output from a previous run.

The SYSPRINT window is an edit view which supports execution of CBLe commands and macros. This allows the user to manipulate, highlight and locate data in the view (e.g. LOCATE, TAG, ALL, CHANGE, SET ZONE, etc.)

Unless SELCOPY options NOPRINT or NOPCTL are specified in the control statements, the input statements and their selection ids are also written to SYSPRINT. Similarly, unless SELCOPY options NOPRINT, NOPSUM or NOPTOT are specified in the control statements, the summary totals are written to SYSPRINT at end of job.

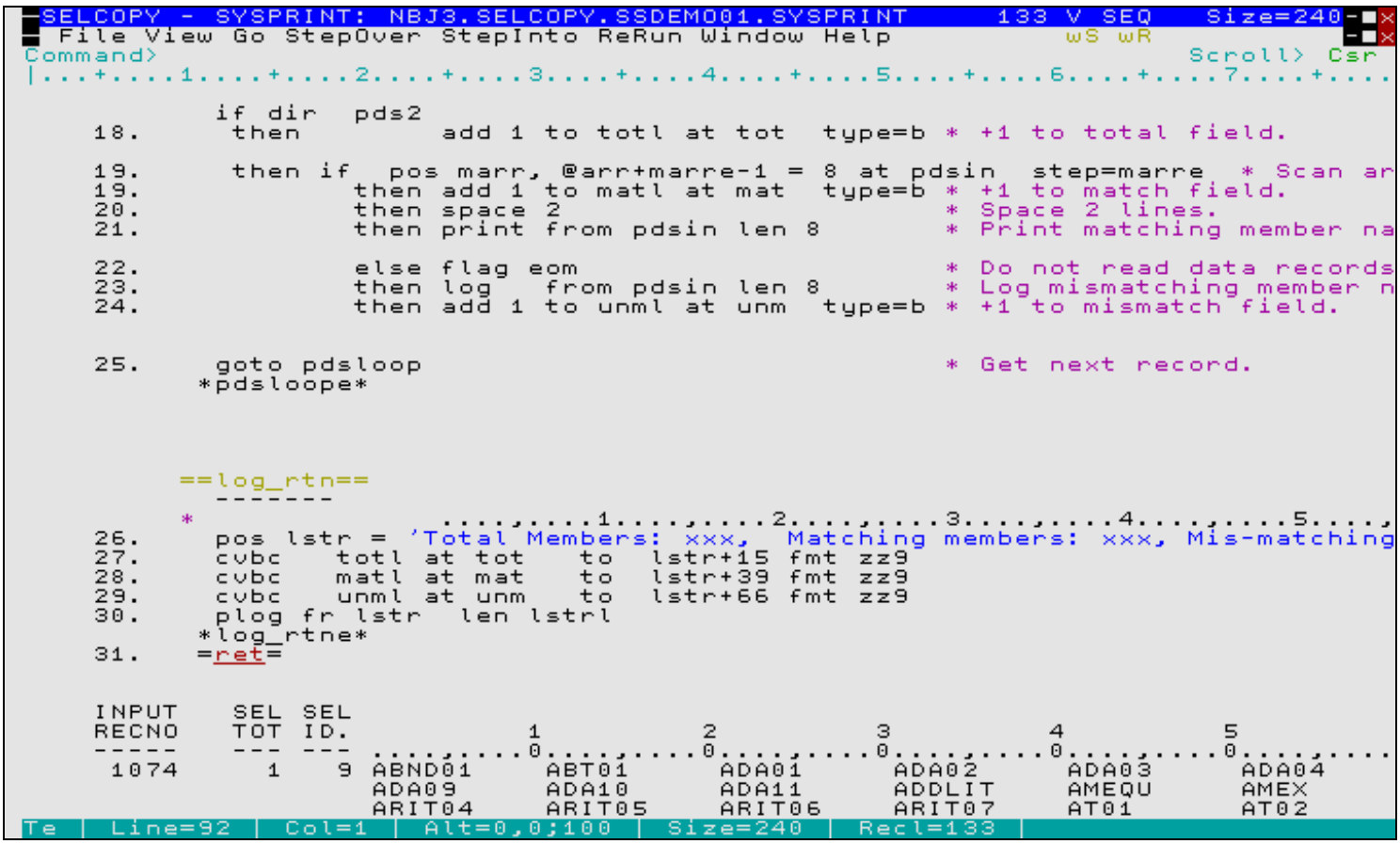


Figure 114. SELCOPY SYSPRINT Window.

SQL Log Window

The SQL log window may be opened via the following:

- Select 'SQL log' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW SQL.

A SELCOPY job that submits SQL statements to a DB2 data base, also writes detailed information about the SELCOPY SQL processing to a data set allocated to ddname **CBLSQLOG**.

SELCOPY Debug intercepts output to CBLSQLOG and displays it in the SQL Log window instead. Because of this, CBLSQLOG does not need to be allocated to display this information.

The SQL Log window is an edit view which supports execution of CBLe commands and macros. This allows the user to manipulate, highlight and locate data in the view (e.g. LOCATE, TAG, ALL, CHANGE, SET ZONE, etc.)

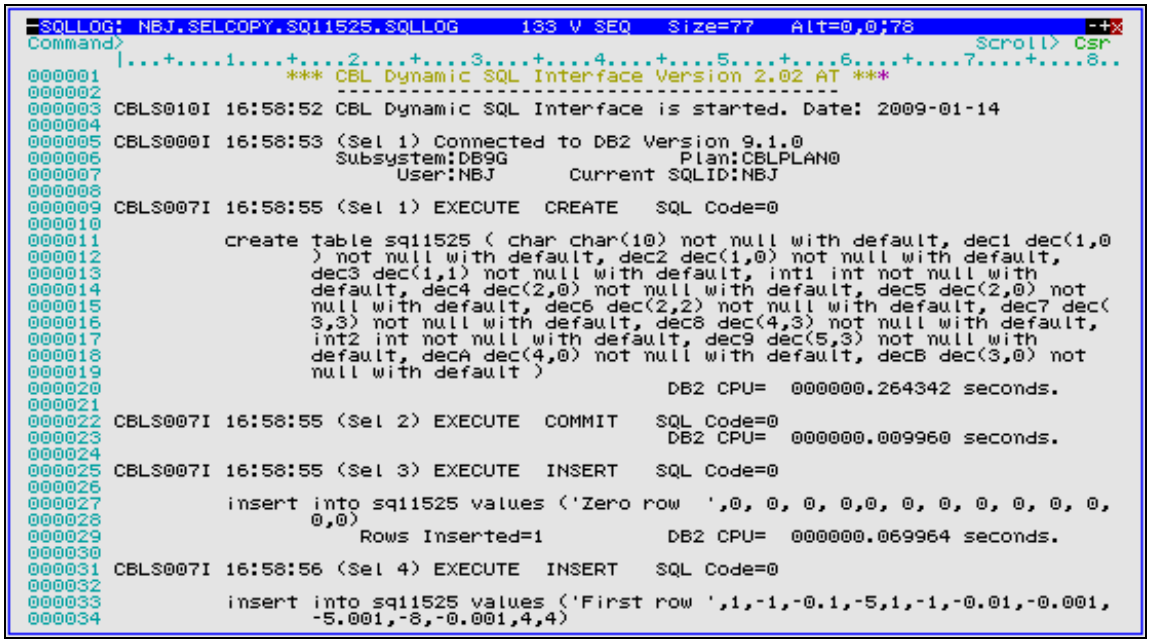


Figure 115. SELCOPY SQL Log Window.

WTO Log Window

The WTO log window may be opened via the following:

- Select 'WTO log' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW WTO.

SYSLOG output to the Operator's Console, TSO, CMS or ICCF user terminals is intercepted by SELCOPY Debug and is displayed in the WTOLOG window instead.

The WTO Log window is opened automatically when SYSLOG output is received. This may be warning/error messages returned by SELCOPY, or output generated by a SELCOPY LOG operation.

The WTO Log window is an edit view which supports execution of CBLe commands and macros. This allows the user to manipulate, highlight and locate data in the view (e.g. LOCATE, TAG, ALL, CHANGE, SET ZONE, etc.)

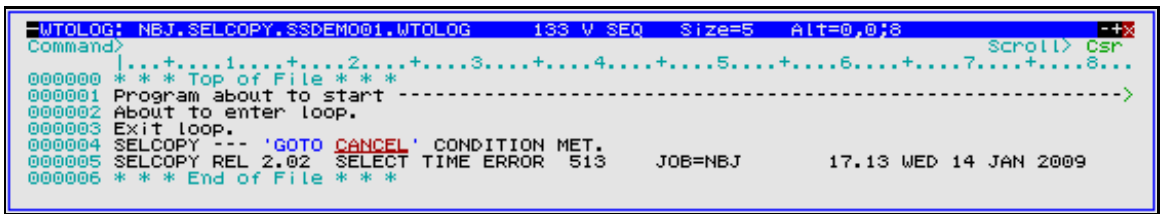


Figure 116. SELCOPY WTO Log Window.

Work Area/Current Input Record Window

A Work Area/Current Input Record storage display window is opened automatically when SELCOPY Debug is started. Further storage display windows may also be opened via the following:

- Select 'Work area' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW WORKAREA.

The current status of the user work area (or input record buffer if no work area is allocated) is displayed in the Work Area window. A Work area window is a storage display window.

Note that, if WORKLEN is not supplied, the Work Area window has the title: Current Input Record.

Any number of Work Area windows may be opened and each window may be tailored to display different portions of the work area.

The appearance of the Work Area window may be updated using the storage window display options popup menu. The options available and methods used to display this menu are documented under the line command, SHOWPOPUPMENU.

The work area position, in the first row of the Work Area window, is an enterable field (highlighted in red by default.) Here, you may enter the work area position from which data is to be displayed.

Line commands UP CURSOR and DOWN CURSOR may also be used to navigate the Work Area window. By default, UP CURSOR is assigned to PF07 and DOWN CURSOR is assigned to PF08.

Data in the work area may be altered at any point during the run by overtyping text in either the character or hexadecimal display. A change to text in the one display will automatically be reflected in the other.

Figure 117. SELCOPY Work Area Window.

POS Expression Window

The POS expression window may be opened using the SELCOPY Debug CLI command WINDOW POS expr. POS expression windows for special positions POS PARM, DATE, SQLCA, SQLDA and SQLMA may be opened by selecting the "Pos" sub-menu from the View menu in the SELCOPY Debug Main Menu.

The POS window displays storage in the exactly same way as the Work Area/Current Input Record window with the exception that the start address of the displayed data is a position in storage evaluated by a valid SELCOPY POS expression instead of position 1 of the work area. Like the Work Area window, the appearance of the POS expression window may be updated using the storage display window options popup menu. The options available and methods used to display this menu are documented under the line command, SHOWPOPUPMENU.

The POS expression is re-evaluated at each break in the SELCOPY execution and the data at the new position displayed in the POS window.

The POS window title contains the POS expression and the evaluated position in the work area in parentheses. If the evaluated position falls outside the work area, then **(Not in WorkArea)** is displayed instead.

Any number of POS windows may be opened.

Figure 118. POS Window (inside work area)

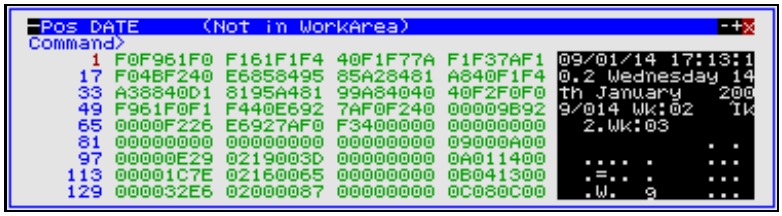


Figure 119. POS Window (outside work area)

@ Pointer Window

The @ Pointer window may be opened via the following:

- Select '@ Pointers' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW @.

The current status of the @ pointer, LRECL and of all the user @ pointers to be used in the current execution of SELCOPY, is displayed in the @ Pointer window.

The @ Pointer window has the same characteristics as a SELCOPY/i List window including selecting, sorting and filtering of row and column data and "point and shoot" sorting on column headers.

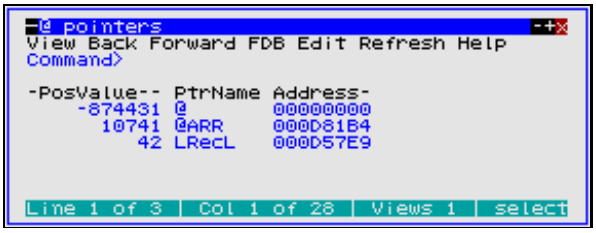


Figure 120. SELCOPY @ Pointer Window.

Columns Displayed

Name	Type	Description
PosValue	Int	Value as a position in the work area.
PtrName	Char	Pointer Name.
Address	Hex	Address in storage of position in work area.

Equates Window

The Equates window may be opened via the following:

- Select 'EQUates' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW EQUATES.

All equated names and their values, set by the user via an EQU statement and subsequently allocated by SELCOPY during control statement analysis, are is displayed in the Equates window.

The Equate window has the same characteristics as a SELCOPY/i List window including selecting, sorting and filtering of row and column data and "point and shoot" sorting on column headers.

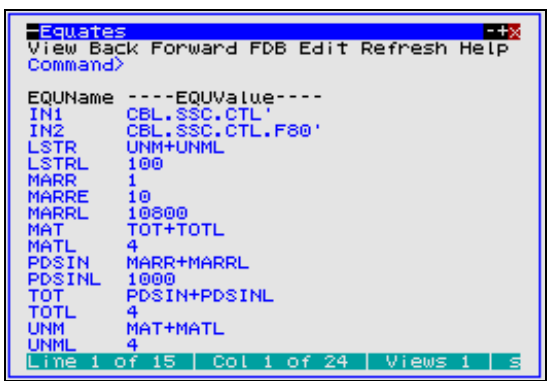


Figure 121. SELCOPY EQUates Window.

Columns Displayed

Name	Type	Description
EQUName	Char	Equated name.
EQUValue	Char	Equated value.

PCB Window

The PCB window may be opened via the following:

- Select 'PCB' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW PCB.

This window shows the PCB which was used to execute the most recent IMS call

The PCB displayed will change if different PCBs are used in the SELCOPY program.

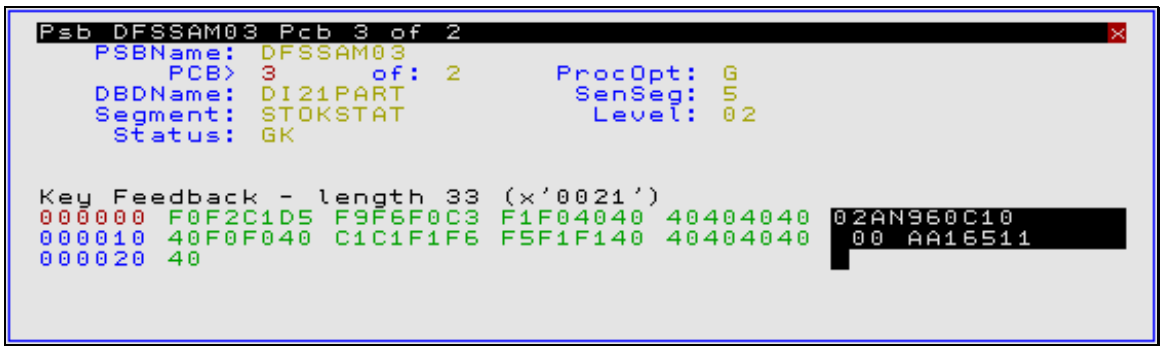


Figure 122. SELCOPY PCB Window.

TRACE Window

The TRACE window is opened via the following:

- Select 'Execution trace' from the View menu in the SELCOPY Debug Main Menu.
- Enter the SELCOPY Debug CLI command WINDOW TRACE.

The TRACE window is a CBLe edit view that contains all the SELCOPY control statements at which processing has been stopped. i.e a break point was set and encountered. Each logged statement begins with the statement's selection id.

Note that the STEPINTO and STEPOVER commands dynamically set and unset break points to allow stepping through the SELCOPY job. The STEPINTO command sets a break point on the next control statement to be executed following the current control statement. Therefore, when repeatedly issuing STEPINTO or STEPOVER, the TRACE window displays a log of all the statements executed so far.

Figure 123. SELCOPY TRACE Window.

Watch List Window

A SELCOPY Debug Watch List window displays the current value of specific variables and fields in storage and is opened using the WATCH primary command. It allows the user to focus on specific values of interest throughout the debug session.

Any number of Watch lists may be opened. Each instance of a Watch list has a unique name identifier which is assigned when the list window is opened. The name used may be specified on the WATCH command or otherwise allowed to defualt to W*n* (where *n* is a number in an internally maintained sequence).

Each entry of a Watch list must either be a field (identified by a storage location, length and data type) or any variable supported by the SELCOPY or SLC programs. Specifically, these are:

- SELCOPY internal variables (e.g. LRECL, DIFF, RETCODE).
- SELCOPY internal variable named source fields (e.g. UXLRECL, UXADIFF, UXATPTR).
- User @pointer variables.
- User and SLC generated declared variables. (SLC only)

Watch list primary commands and options exist which support insert, move, exclude and delete of entries, and also alter the appearence of the list rows and columns. See WCOMMAND which supports execution of a Watch list command or set/query of an option from any SELCOPY Debug window.

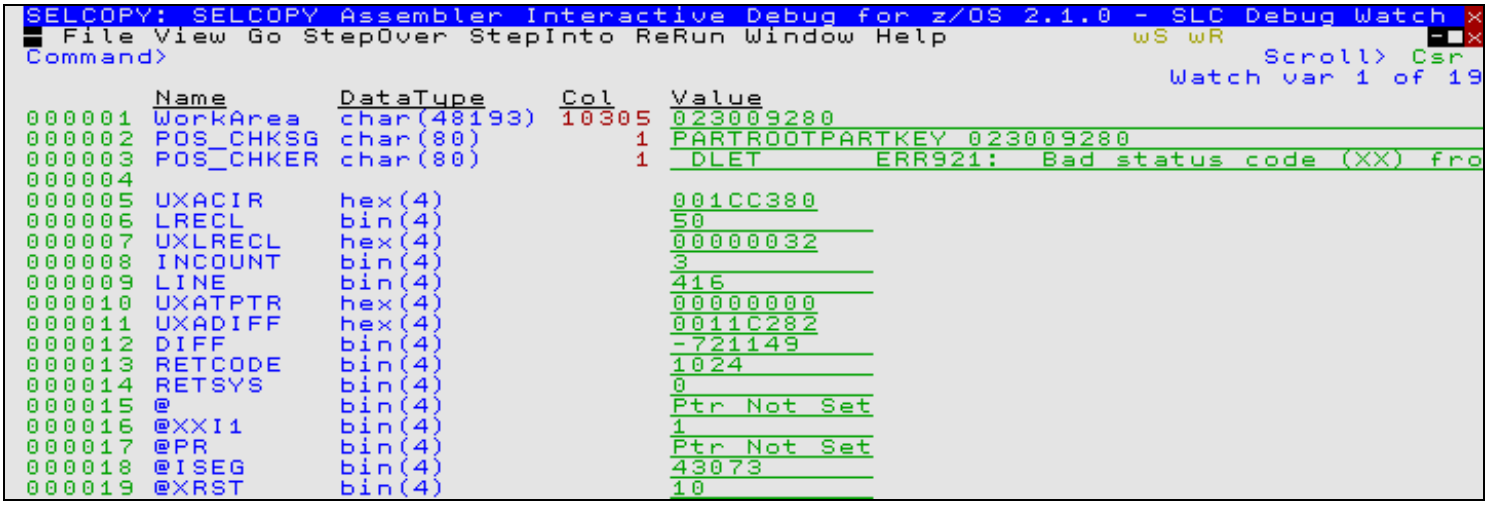


Figure 124. SELCOPY Debug: Watch List.

Watch List Columns

Columns displayed in the Watch list are as follows:

Name
: The name of the variable or the name generated by SELCOPY Debug for a field definition or POS *expression*. The generated field name is the field definition syntax with each blank replaced by an "_" (underscore) character. If a SELCOPYi datatype and length specification is used (e.g. char(22)), then this is omitted from the generated name so that it is simply "POS_*expression*".

DataType
: The data type of the variable or field definition. For decimal data types (packed and zoned), the precision and scale of the value immediately follows in parentheses. For all other data types, the length of the value/field follows in parentheses.

  Display of this column is optional and may be included or excluded using SET DATATYPE ON/OFF. Possible data type entries are as follows:

bfp(n)	Binary floating point with source field length of *n* bytes.
bin(n)	Binary integer with source field length of *n* bytes.
char(n)	Character of fixed length *n*.
charb(n)	Non-printable character (binary data) of fixed length *n* bytes.
charv(n)	Blank padded variable length character of maximum length *n*.
cstring(n)	Null terminated character string of maximum length *n*.
dec(p,s)	Packed decimal of precision *p* and scale *s*.
dfp(n)	Decimal floating point with source field length of *n* bytes.
hfp(n)	Hexadecimal floating point with source field length of *n* bytes.
ptr(4)	Storage address pointer field of length 4 bytes.
vchar(n)	Variable length character of maximum length *n*.
zd(p,s)	Zoned decimal of precision *p* and scale *s*.

Col

Applicable to values of a character data type only, this column displays the horizontal scrolling position of the first character displayed in the Value column.

The column position may be displayed as either a decimal or hexadecimal value using SET COLUMN DEC/HEX. This value may also be overtyped to reposition the data within the Value column.

Value

The value assigned to the named variable or field definition.

For numeric fields, the value is displayed in decimal with "." (dot/period) representing a decimal point and a "-" (minus) prefix representing a negative value. Positive fixed point values have no sign prefix. Floating point values are normalised, are always displayed with a leading "+" (plus) or "-" (minus) sign and include a signed exponent of the form "E+nn" or "E-nn".

For variables and fields of character data type, an individual value may be scrolled horizontally (LEFT and RIGHT) without scrolling other values in the Value column. Furthermore, a scale line may be displayed above the value using SET SCALE ON/OFF. This scale line scrolls with the value's text.

For @pointer variable values that are unset, the Value field displays "Ptr Not Set". If a storage field position expression cannot be determined, the Value column displays "?? Location unresolved ??" (or a sub-string of this message if the value length is less than 25).

Watch List Prefix Area (Line) Commands

The Watch list includes a prefix area which, unless modified or removed using the SET PREFIX option, is displayed on the left of the list entry data and occupies 6 display columns.

The following Watch list prefix area (line) commands are supported. The list entry against which a line command is entered, is considered to be the focus entry for the command.

Command	Description
`.name`	Set a line pointer (line name) at the focus list entry.
`A`	Set the focus entry as the target of a prefix area move command. (Move entries after the focus entry).
`B`	Set the focus entry as the target of a prefix area move command. (Move entries before the focus entry).
`D[n] D* DD`	Delete an entry or a block of entries. **D*n*** deletes a block of *n* entries starting at the focus entry. D* deletes a block of all entries from the focus entry to the end of the list. A **DD** pair is used to mark the first and last entry in a block of entries to be deleted. Note that excluded enries are included within a block of entries to be deleted.
`F[n] F*`	Applicable only if the focus entry is a shadow line (SHADOW ON), this command includes and displays the first excluded entry or block of entries represented by the shadow line. **F*n*** includes the first block of *n* entries. If *n* is greater than or equal to the number of excluded entries represented by the shadow line, then all its excluded entries are included. **F*** includes all excluded lines represented by the shadow line.
`I`	Open the Add WATCH Var/PosExp panel to insert a Watch List entry following the focus entry.
`L[n] L*`	Applicable only if the focus entry is a shadow line (SHADOW ON), this command includes and displays the last excluded entry or block of entries represented by the shadow line. **L*n*** includes the last block of *n* entries. If *n* is greater than or equal to the number of excluded entries represented by the shadow line, then all its excluded entries are included. **L*** includes all excluded lines represented by the shadow line.
`M[n] M* MM`	Mark an entry or a block of entries to be moved before or after a target list entry line which has been selected using prefix command B or A respectively. **M*n*** marks a block of *n* entries starting at the focus entry. **M*** marks a block of all entries from the focus entry to the end of the list. An **MM** pair is used to mark the first and last entry in a block of entries to be moved. Note that excluded entries are included within a block of entries to be moved.
`S`	Applicable only if the focus entry is a shadow line (SHADOW ON), this command includes and displays all excluded entries represented by the shadow line.
`PW`	Open a POS Expression Window for the focus entry.
`SC`	Applicable only if the focus entry is of character or hex data type, this command toggles display of a counting scale line above the Value column of the focus list entry.
`SP`	Insert a blank (spacer) entry after the focus entry.
`X[n] X* XX`	Exclude an entry or a block of entries. **X*n*** excludes a block of *n* entries starting at the focus entry. **X*** excludes a block of all entries from the focus entry to the end of the list. An **XX** pair is used to mark the first and last entry in a block of entries to be excluded. Note that entries that are already excluded are still included within a count of entries to be excluded.

Add WATCH Var/PosExp Panel

The SELCOPY Debug Add WATCH Var/PosExp panel (ZZSGSDBW) provides a method by which entries may be inserted or updated in the Watch list.

The panel is opened by the INSERT primary command or the **Watch Pos** item of the Point-and-Shoot popup menu.

A variable, field definition or space line is inserted following the focus entry (cursor position) and has equivalent operation to the WATCH POSITION, VARIABLE and SPACE primary commands. A field definition uses the expression supplied to identify the field POS location, together with a length/precision and a datatype supported by SELCOPYi.

Figure 125. SELCOPY Debug: Add WATCH Var/PosExp Panel.

Panel Input Fields

Type>
Specifies the type of Watch List entry to be added. Possible values are "V" (for a variable), "P" (for a field positional expression) or "S" (for a space line).

If "P" is specified, the POS Expression Length and Datatype input field values are used. Otherwise, they are ignored.

VarName/PosExp>
For type "V", specifies the variable name whose value is to be displayed or, for type "P", the POS expression at which the field value is defined.

Length>
For type "P" only, the length (or packed decimal precision) of the field data value to be displayed.

Data-Type>
For type "P" only, the data type of the field value's source data. The data type determines how the field value will be displayed.

Operations List

The SELCOPY Debug Operations List window displays the current status held by SELCOPY Debug for each executable statement (SELCOPY operation) in the control statements input file.

The list is opened using the LIST OPERATIONS primary command and may be used to audit the SELCOPY or SLC program run and to demonstrate successful execution of a particular logic path through the SELCOPY control statements.

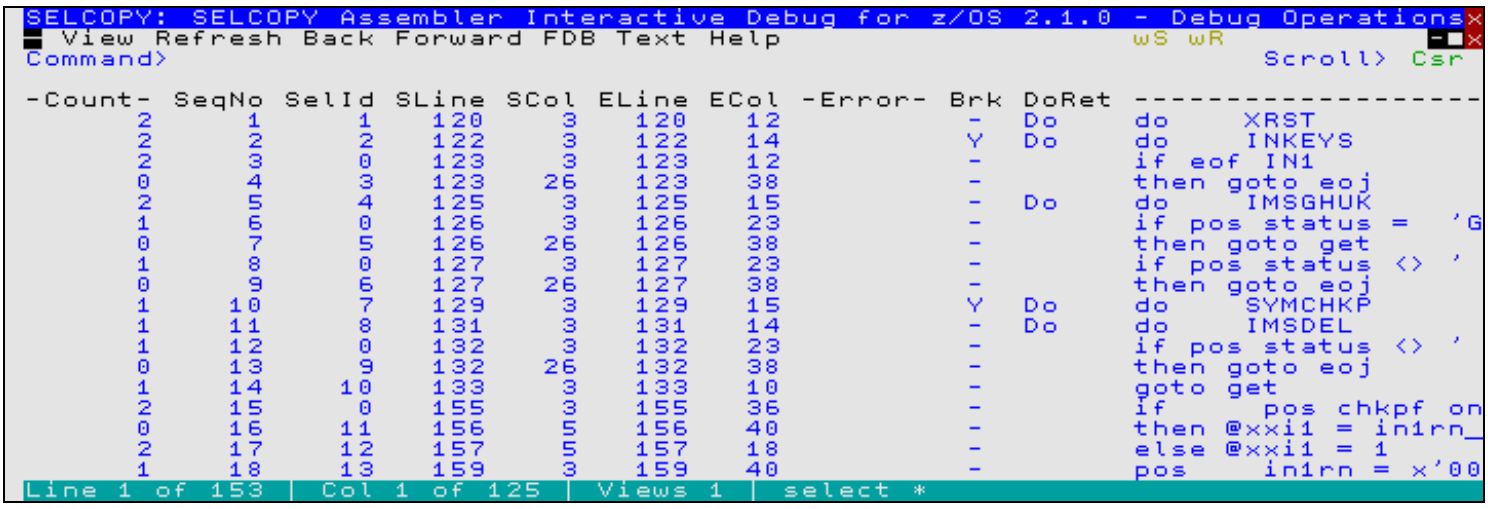


Figure 126. SELCOPY Debug: Operations List.

Columns Displayed

Columns displayed in this list are as follows:

Name	Type	Description
Count	Integer	Operation execution count.
SeqNo	Integer	Operation sequence number.
SelId	Integer	Operation SELCOPY selection id.
SLine	Integer	Control statement start line.
SCol	Integer	Control statement start column.
ELine	Integer	Control statement end line.
ECol	Integer	Control statement end column.
Error	Enum	Control statement is in error. (Seq, Invalid or Overlap)
Brk	Flag	Operation has a breakpoint set.
DoRet	Enum	Operation is DO or RETURN.
Text	Character	Operation text.

Point-and-Shoot Popup Menu

All SELCOPY Debug text edit type windows, including the SYSIN/SYSIPT and SYSPRINT/SYSLST windows, support the point-and-shoot options popup menu. This popup menu is opened using the SDBPOPUP primary command which, for SYSIN/SYSIPT, is assigned to F16 by default.

The cursor position within the edited data identifies the focus text to be referenced in items of the point-and-shoot menu when it is opened. If the text is a SELCOPY expression, then, where applicable, items are displayed for the complete expression as well as the expression term on which the cursor is positioned.

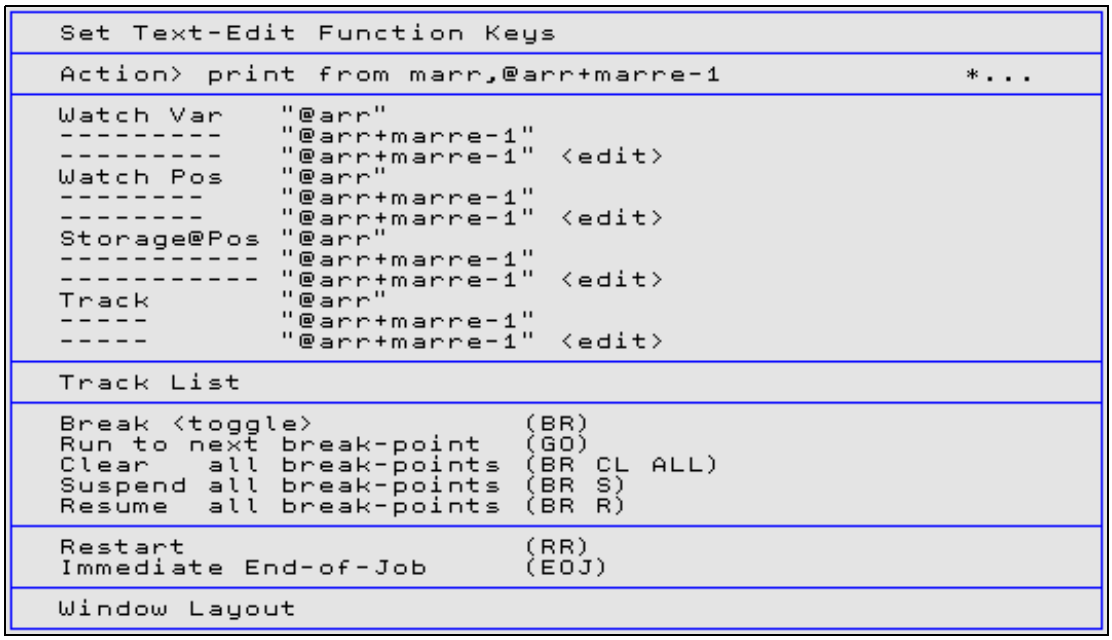


Figure 127. SELCOPY Point-and-Shoot Popup Menu Window.

The menu enables the user to quickly and easily perform the following, commonly used tasks:

`Set Text-Edit Keys/Debug Function Keys`
Certain default PFKey assignments for a text edit view differ to defaults set up for the SELCOPY Debug environment. This entry enables the user to toggle between the Text Edit default keys and SELCOPY Debug keys as follow:

PFKey	Edit	Debug
PF13	SOS LineAdd	STEPOVER
PF14	SOS LineDel	StepInto
PF15	Duplicate	Go
PF19	SpltJoin	BreakPoint

Action> *command_string*
Execute *command_string* as determined by text at the focus line and column. See the ACTION facility for further information on how *command_string* is determined.

Watch Var *varname*
Open the SELCOPY Debug Watch Window and add an entry for variable *varname*. The name *varname* is identified as the token on which the cursor is positioned. If the token is a term in a SELCOPY expression, another Watch Var menu item follows for the complete expression.

The <edit> item places the SELCOPY Debug primary command `WATCH [VARIABLE] varname` at the command prompt.

Watch Pos *expression*
Open the SELCOPY Debug Add WATCH Var/PosExp panel for *expression*, which includes input field entries for data type and length/precision. Hit <Enter> to then open the SELCOPY Debug **Watch Window** and add or update the entry for POS *expression*. The Value field will display the value of data at the storage location identified by POS *expression*.

The *expression* is identified as the token on which the cursor is positioned. If the token is a term in a SELCOPY expression, another Watch Pos menu item follows for the complete expression.

The <edit> item places the SELCOPY Debug primary command `WATCH POS expression CHA(100)` at the command prompt, where *expression* is the complete expression on which the cursor is positioned.

Storage@Pos *expression*
Open a POS Expression Window starting at the position in storage defined by POS *expression*.

The *expression* is identified as the token on which the cursor is positioned. If the token is a term in a SELCOPY expression, another Storage@Pos menu item follows for the complete expression.

The <edit> item places the SELCOPY Debug primary command `WINDOW POS expression` at the command prompt, where *expression* is the complete expression on which the cursor is positioned.

Track *expression*
Invoke the **SDBTRACK** edit macro which issues the SELCOPY Debug primary command `TRACK expression` to start or stop tracking a position in storage defined by POS *expression*.

The *expression* is identified as the token on which the cursor is positioned. If the token is a term in a SELCOPY expression, another Track menu item follows for the complete expression.

If a Track entry is selected, another popup menu is opened prompting the user to select the colour to be used for tracking this POS expression or, alternatively, to turn off tracking for this POS expression.

The <edit> item places the Rexx macro invocation `SDBTRACK expression` at the command prompt, where *expression* is the complete expression on which the cursor is positioned.

Figure 128. SELCOPY TRACK Colour Popup Menu Window.

Track List
Open a popup menu displaying a list of all POS expressions that are being tracked. The user can then select an entry to switch off tracking for that POS expression or select **All** to switch off tracking of all the POS expression entries.

Figure 129. SELCOPY TRACK List Popup Window.

Break <toggle> (BR)
Toggle a break point on and off for the SELCOPY operation at the cursor position. i.e. SELCOPY Debug primary command `BREAKPOINT TOGGLE CURSOR`.

Run to next break-point (GO)
Allow the program to run until either a break point or End-of-Job is encountered.

Clear all break-points (BR CL ALL)
Clear all existing break points. i.e. SELCOPY Debug primary command `BREAKPOINT CLEAR ALL`.

Suspend all break-points (BR S)
Temporarily suspend (de-activate) all existing break points until a BREAKPOINT RESUMEALL is executed. i.e. SELCOPY Debug primary command `BREAKPOINT SUSPENDALL`.

Resume all break-points (BR R)
Re-activate all suspended break points. i.e. SELCOPY Debug primary command `BREAKPOINT RESUMEALL`.

Restart (RR)
Force immediate End-of-Job and restart the program from the beginning. Processing will break prior to execution of the first SELCOPY control statement.

Immediate End-of-Job (EOJ)
Force immediate End-of-Job processing.

Window Layout
Invoke the **SDBWINX** edit macro which opens a popup menu enabling the user to control the configuration of windows within the SELCOPY Debug MDI environment.
The user can save and subsequently restore the characteristics of the current MDI child window or all currently open MDI child windows. Alternatively, the user can select the default configuration for the current, or all SELCOPY Debug MDI child windows.

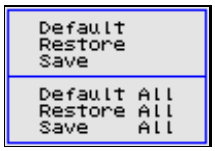


Figure 130. SELCOPY Window Layout Popup Menu.

SELCOPY Debug Commands

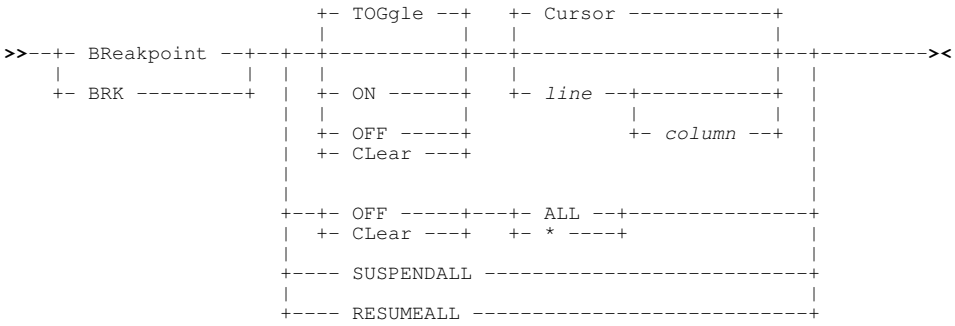
You can issue SELCOPY Debug commands from the command line at the Command> prompt. Most SELCOPY Debug main window menu options have a command line equivalent.

Command	Description
BReakpoint	Set/unset temporary break points.
EOJ	Force SELCOPY End-of-Job.
GO	Continue processing.
LIst OP	Open the Operations List window.
RErun	Re-run from the beginning.
STEPInto	Step (Trace) Into sub-routines.
STEPOver	Step (Trace) Over sub-routines.
TRack	Track a SELCOPY POS expression.
Watch	Open the Watch List window.
WCommand	Pass a command to the Watch list window.
WIndow	Open a specified SELCOPY Debug window.

See SELCOPY Debug Function Keys for default PFKeys settings.

BREAKPOINT

Syntax:



Description:

Use the BREAKPOINT command to set or unset break points in the SYSIN window. By default, BREAKPOINT is assigned to **Shit-F7 (F19)**.

If a break point is set at a particular control statement, then processing will be paused on the next attempt to execute that statement.

Break points may persist across a RERUN of the SELCOPY or SLC program. Beware that break points are assigned to statements at particular line and column numbers within the SYSIN control statement file. If the SYSIN input is updated, then a break point may no longer point at the intended statement.

Any number of concurrent break points may be active during job execution.

Parameters:

TOGGLE | **ON** | **OFF**
TOGGLE, **ON** and **OFF** (synonym **CLEAR**) keyword parameters specify whether a new break point is set (ON) or an existing break point is unset (OFF/CLEAR) at a particular input control statement. TOGGLE will set a breakpoint if it is currently unset, or unset the breakpoint if it is currently set.

CURSOR | *line* [*column*]
The *line* and *column* integer arguments specify the line and column number within the SYSIN control file Text Editor view at which the control statement is located. This control statement is the target of the BREAKPOINT command. If *column* is not specified, the defaut is column 1.

CURSOR is default, indicating that the control statement at the cursor position is the target of the BREAKPOINT command. If the cursor is not positioned within the window display area, the last known position of the cursor within the display area is used. If not positioned on a line containing a control statement, the first control statement within the window's client area is used.

ALL
*****

ALL, or its synonym "*" (asterisk), is applicable only to the OFF (CLEAR) option and indicates that all existing break points are to be unset.

SUSPENDALL

The **SUSPENDALL** keyword parameter indicates that all existing break points are to be temporarily suspended (de-activated) until a BREAKPOINT RESUMEALL is executed.

RESUMEALL

The **RESUMEALL** keyword parameter indicates that suspended break points are to be re-activated.

EOJ

Syntax:

```
>>-- EOJ ------------------------------------------------------------------><
```

Description:

Use the EOJ command to force SELCOPY to immediately execute a "GOTO EOJ" operation.

The SELCOPY job will end without processing any further control statements and will generate its output summary in the SYSPRINT window.

GO

Syntax:

```
>>-- GO -------------------------------------------------------------------><
```

Description:

Use the GO command to continue processing of the control statements. By default, function key **Shift-F3 (F15)** is set to GO.

Processing will continue until a break point or End-of-Job is encountered at which point processing is paused or stopped respectively.

LIST OPERATIONS

Syntax:

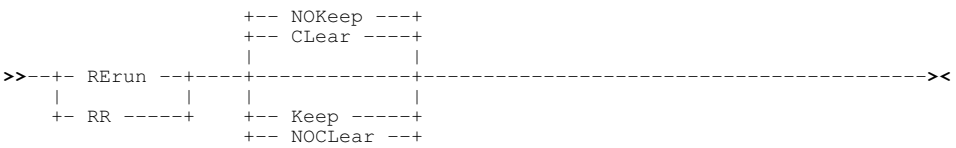
```
>>--+- LIst --- OPerations ---+-----------------------------------------------><
    +- LO --------------------+
```

Description:

LIST OPERATIONS will display the SELCOPY Debug Operations List window which provides statistical analysis information for each executable SELCOPY statement.

RERUN

Syntax:



Description:

Use the RERUN command to Re-Run the job from the beginning. No further statements will be executed from the existing job run.

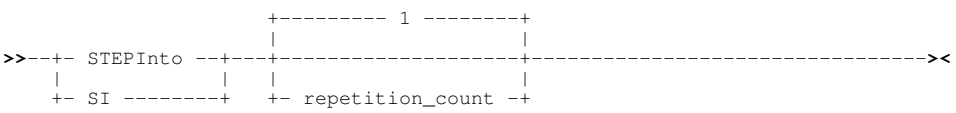
Wherever possible, debug windows, watch list variables, tracked values and break points are preserved from the previous run.

Parameters:

NOKEEP | **CLEAR**
KEEP | **NOCLEAR**
NOKEEP (or CLEAR) will clear the contents of the SYSPRINT window before re-running the SELCOPY control statement analysis. KEEP (or NOCLEAR) will preserve the contents of the SYSPRINT window so that the output from a previous run may still be displayed. The default option is NOKEEP.

STEPINTO

Syntax:



Description:

Use the STEPINTO command to step through the SELCOPY control statements logically one at a time. By default, STEPINTO is assigned to **Shift-F2 (F14)**.

Any branch to a SELCOPY sub-routine via a **DO**, **PERFORM** or **GOSUB** operation will be Stepped Into. i.e. processing is paused on each control statement in the sub-routine.

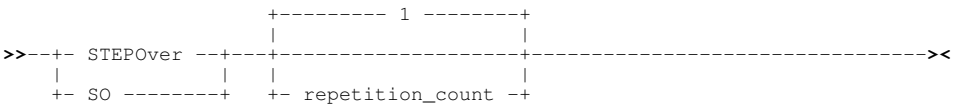
STEPINTO and STEPOVER set and then unset temporary break points in the SELCOPY control statements in order to pause processing.

Parameters:

repetition_count
The number of control statements to step. This parameter is optional and defaults to 1. Note that execution will be paused before this number of control statements have been executed if a breakpoint is encountered.

STEPOVER

Syntax:



Description:

Use the STEPOVER command to step through the SELCOPY control statements logically one at a time. By default, STEPOVER is assigned to **Shift-F1 (F13).**

Any branch to a SELCOPY sub-routine via a **DO**, **PERFORM** or **GOSUB** operation will be Stepped Over. i.e. the sub-routine is executed and processing is paused again on the control statement following the sub-routine call.

STEPINTO and STEPOVER set and then unset temporary break points in the SELCOPY control statements in order to pause processing.

Parameters:

repetition_count
The number of control statements to step. This parameter is optional and defaults to 1. Note that execution will be paused before this number of control statements have been executed if a breakpoint is encountered.

TRACK

Syntax:

```
>>-- TRack -- expr ---+--------------+----------------------------------------><
                      |              |
                      +--- colour ---+
                      |              |
                      +---- OFF -----+
```

Description:

Use the TRACK command to track the value of a valid SELCOPY POS expression as a position in storage.

The single byte, addressed by the POS expression, is highlighted in all open storage windows in which the position is displayed.

The POS expression is re-evaluated for every break in the SELCOPY execution.

Parameters:

expr
A valid SELCOPY POS expression. This may include EQUated names, @ pointers, LRECL special POS keywords (e.g. DATE, COMREG), integer values and arithmetic operators "+" (plus) and "-" (minus).

colour
The colour in which the evaluated position is highlighted. This is a two character code defining the colour and, optionally the extended highlighting, to be used.
Valid colour codes are:

B	Blue
G	Green
P	Pink
R	Red
T	Turquoise
W	White
Y	Yellow

Valid extended highlighting codes are:

B	Blink
N	None (No extended highlighting)
R	Reverse Video
U	Underscore

The default extended highlighting is **R** (reverse video), the default colour is **T** (turquoise).

OFF
Switch off tracking for the specified expression.

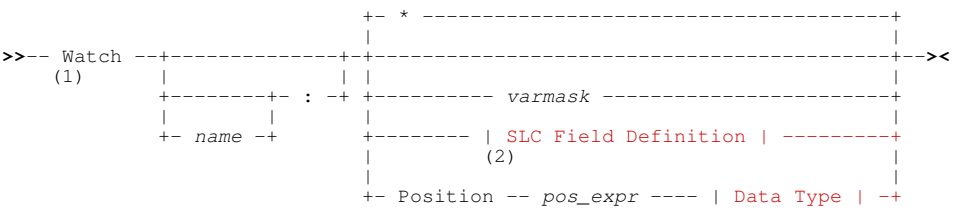
Examples:

```
TRACK @A+10  R
```
Highlight in red (default reverse video) the byte in all storage windows that is referenced by the expression @A+10.

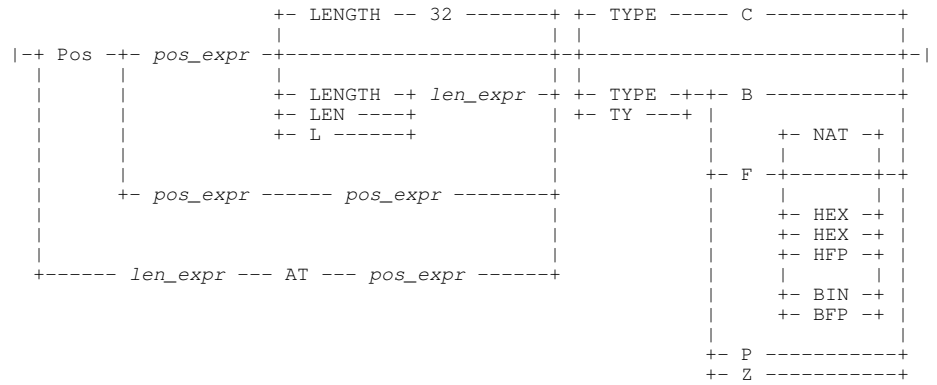
```
TRACK ARRAY+@X-1  GU
```
Highlight in green with underscore the byte in all storage windows that is referenced by the expression ARRAY+@X-1.

WATCH

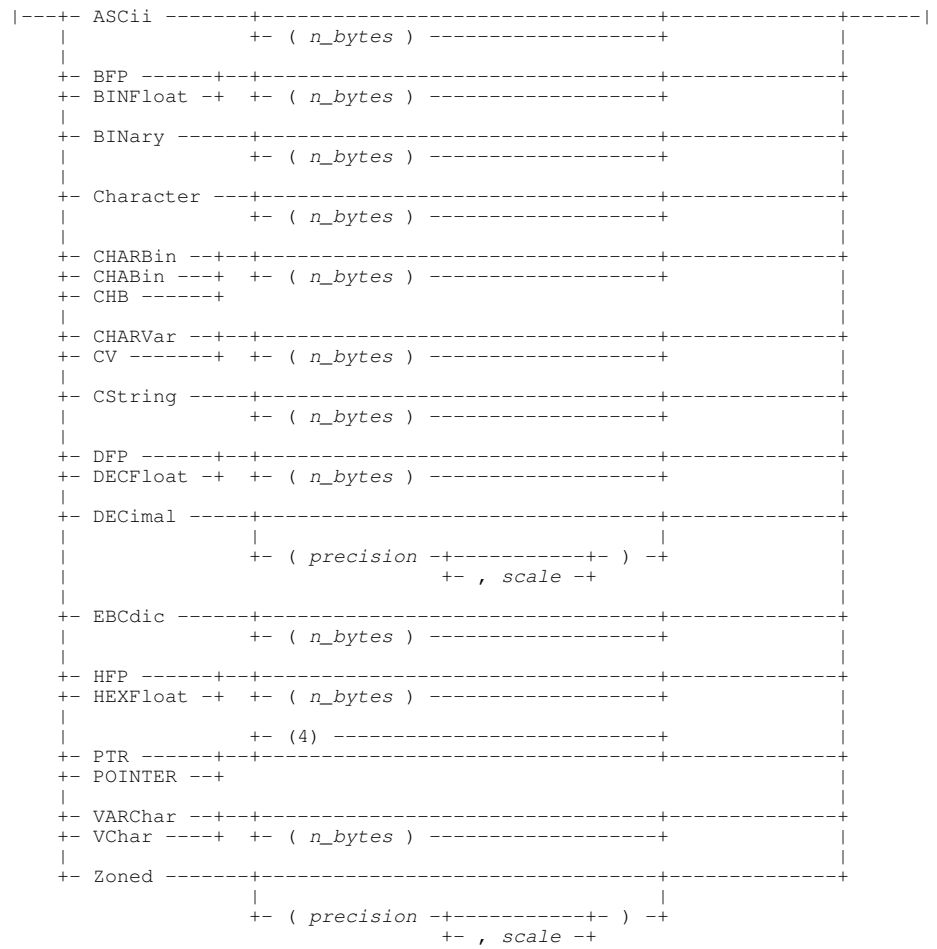
Syntax:



SLC (not SELCOPY) Field Definition:



Data Type:



Notes:

1. If specified with no parameters other than *name*, WATCH will insert an entry for every internal and user defined variable identified within the SELCOPY control statements.
2. SLC Field Definition syntax is supported for the SLC program only.

Description:

WATCH will insert a new variable (*varmask*) or field definition entry in a Debug Watch list immediately following the focus entry. The entry will replace an existing entry of the same name if it already exists in the watch list.

A variable entry displays the value currently assigned to one or more named SELCOPY internal variables (e.g. LRECL, DIFF, RETCODE), user @pointer variables or SLC declared variables that match the specified variable mask.

A field entry displays a field in storage and may be specified using either of the following two methods:

- **POSITION *pos_expr* with Data Type**
  Specifies a start position, expressed as a SELCOPY POS expression, with a field length and data type determined by a data type keyword and parenthesised length/precision value suffix.
- **SLC Field Definition**
  Supported for the SLC program command interpreter only, the field may be specified using SLC Type 1 (***field_pLENn***), Type 2 (***field_p1p2***) or Type 3 (***field_nATp***) field definition syntax. (See *"Field Definitions"* in the *"SLC Language Reference"* for details.)

Multiple Watch list windows may exist, each having a unique name. Therefore, the *name* parameter may be used to identify the Watch list instance to which the list entry will be added.

If *name*: is not specified, the list entry will be added to the current Watch list window (i.e. the last Watch list window to have the focus). If *name* does not match that belonging to an open Watch list window, then a new Watch list window is opened for *name* and the list entry added accordingly. If ":" (colon) is specified without *name*, the list entry will be added to a new Watch list window with a generated name in the format W*n* (where *n* is the next number in an internally maintained sequence.)

WATCH will attempt to interpret parameters in the following order of sequence:

1. Scan the list of known SELCOPY variables for a match on the input parameter(s). If a match is found, the WATCH parameter is treated as *varmask*.
2. **For SLC only**, pass the input parameter(s) to the SLC field interpreter. If SLC verifies the parameters as being valid field specification, the WATCH parameters are interpreted as being an SLC field specification.
3. If keyword POSITION is the first word of the parameter string, the remaining WATCH parameters are interpreted as being a SELCOPY POS expression and Data Type specification.

All Watch list entries are re-evaluated and the contents of the Value columns updated each time processing is paused by SELCOPY Debug. Therefore, any changes to the value of a variable or a variable on which a field position or length expression is based, will be reflected in the watch list Value columns.

The value of a field which has been assigned a numeric data type, is displayed as a decimal. A floating point field value is displayed as a signed mantissa with signed exponent. The value of a CHARBIN field is displayed as printable hex so that each byte is represented by 2 characters, each being a hex digit. (e.g. CHARBIN(2) may have a value displayed as "C1C2" for characters "AB".)

If SELCOPY Debug is running with maximised debug windows, execution of WATCH will also place focus on the Watch List window. Otherwise, focus will remain on the window from which the command was executed.

Parameters:

`[name]` :
Optionally specifies a 1 to 64 character name that identifies the Watch list window in which the list entry will be added. If ":" (colon) is specified without *name*, a name will be generated with format W*n*.
Default is the current Watch list window.

varmask
A character string which identifies 1 or more variables that have been established during SELCOPY control statement analysis. All variables that match *varmask* will be selected for display in the Watch list.

A *varmask* may contain wildcard characters "*" (asterisk), which represents zero or more consecutive characters, and "%" (percent), which represents a single character within the variable name.

If no parameters other than *name* and/or ":" (colon) are specified on the WATCH command, a *varmask* of "*" is implied.

`SLC Field Definition`
Applicable to debug using the SLC program interpreter only, an SLC field definition is any syntax used to specify a field to SLC. This involves use of POS, LENGTH or AT keywords with optional TYPE specification.

Note that fields may also be specified using the SELCOPY Debug format field definition. See POSITION.

POSITION

Indicates that the *pos_expr* that follows is to be used in a **SELCOPY Debug format** field definition. Note that the SLC Field Definition format syntax supports POS or P (not POSITION) as its field position keyword indicator.

If the SLC program interpreter is used and abbreviation POS (or P) is specified, then the SELCOPY Debug format will be used only if the WATCH operands do not describe valid SLC field definition syntax.

This method of specifying a field Watch list entry is the only one available when debugging with the SELCOPY program interpreter.

Data Type

Applicable only to a SELCOPY Debug field definition, Data Type specifies the data type of the field. Default is character fixed length 32 (CHARACTER(32)).

The data type parameter keywords support either a length (*n_bytes*) or a fixed point decimal *precision* and *scale*. The *precision* value defines the total number of significant decimal digits and, if specified, *scale* defines the number of fraction digits. Therefore, the *scale* value must be less than or equal to the *precision* value.

A description of all supported data types together with their potential and default length/precision values are as follows:

Data Type	Max Length/Precision	Default	Description
ASCII	1-2147483647	1	Character fixed length text displayed as ASCII.
BFP	4, 8 or 16	4	Binary floating point number.
BINARY	1-8	4	Signed binary integer. (1-byte is unsigned)
CHARACTER	1-2147483647	1	Character fixed length text.
CHARBIN	1-2147483647	1	Unprintable character (binary) fixed length data displayed as hexadecimal by default.
CHARVAR	1-2147483647	1	Character variable length, padded text with 2-byte length field prefix. (Equivalent to PL/1 CHAR VARYING)
CSTRING	1-2147483647	1	Character null terminated variable length text.
DFP	4, 8 or 16	4	Decimal floating point number.
DECIMAL	1-31	7	Decimal (packed) fixed point number.
EBCDIC	1-2147483647	1	Character fixed length text displayed as EBCDIC.
HFP	4, 8 or 16	4	Hex floating point number.
PTR	4	4	Storage address pointer value.
VARCHAR	1-2147483647	1	Character variable length text with 2-byte length field prefix.
ZONED	1-31	7	Decimal (zoned) fixed point number.

pos_expr

The *pos_expr* operand is an expression, specified using valid SELCOPY or SLC syntax, which resolves to be a position in storage. Note that an expression that evaluates to an integer value (e.g. LRECL+1) corresponds to a storage location which is an offset from the work area or last input record buffer address.

The first occurrence of *pos_expr* identifies the field's start position. For an SLC Type 2 field definition (*field_p1p2*) only, a second occurrence of *pos_expr* exists which identifies the last position of the field.

Unlike SELCOPY Debug (and SELCOPY program) field definitions, *pos_expr* in an SLC field definition may include blank characters between the expression terms and operators. If a blank exists in the *pos_expr* of a SELCOPY Debug field definition, then error ZZSC030E - invalid operand will be displayed.

If *pos_expr* includes a term which is a variable (e.g. an @pointer or declared numeric variable) then the equivalent storage position may change throughout the program execution. If so, the text displayed in the Value column will change accordingly.

len_expr

Applicable only to an SLC Type 1 (*field_p1LENn*) field definition following keyword LENGTH (LEN or L) or an SLC Type 3 (*field_nATp*) field definition, *len_expr* is an expression specified using valid SLC syntax which evaluates to be a positive integer value.

The *len_expr* value identifies the length of the field which is then displayed in parantheses in the Watch list DataType column. If *len_expr* includes a variable term, then the evaluated length may change and so will the length of text displayed in the Value column.

TYPE B | C | F [BFP | HFP | NATIVE] | P | Z
TY

Applicable only to an SLC field definition, TYPE specifies the data type of the field (see SLC documentation for more detail). This data type specification is displayed in its equivalent SELCOPY Debug format in the Watch list DataType column.

Examples:

`WATCH UX*`
Display all variables with names beginning "UX". These include user declared variables and SELCOPY internal variables (e.g. UXATPTR, UXLRECL, etc.).

`WATCH @Vars: @*`
Open a new watch window named "@Vars" and display the default @pointer and all other @pointer variables referenced within the SELCOPY control statements.

`WATCH W1: 4 AT 21 TYPE=B`
For SLC only, add a numeric field entry of length 4 at position 21 and binary datatype to the watch window named "W1".

`WATCH W1: POS 21 BIN`
For SELCOPY and SLC, this will achieve the same result as the previous example.

WCOMMAND

Syntax:

```
>>--+- WCommand --+--+--------------+----- | Watch List Command | ------------><
    |             |  |              |
    +- WCMD ------+  +- name -- : -+
```

Description:

A Debug Watch List window supports its own set of sub-commands and options. Using the WCOMMAND primary command, any command string may be passed to a Watch list window for execution, without first having to place focus on that window.

Multiple Watch list windows may be opened and so the *name* parameter is used to identify the Watch list instance to which the command will be passed. If *name* is not specified, the command will be passed to the current Watch list window (i.e. the last Watch list window to have the focus). An error occurs if no Watch list window exists with the specified name.

If a Watch list option value is to be updated, the SET command keyword is mandatory.

If SELCOPY Debug is running with maximised debug windows, execution of WCOMMAND will also place focus on the Watch List window. Otherwise, focus will remain on the window from which the command was executed.

Parameters:

name :
Specifies a 1 to 64 character name which identifies the Watch list window to which the command will be passed. Default is the current Watch list window.

Watch List Command
A primary command text to be passed and executed at the nominated Watch list window.

Examples:

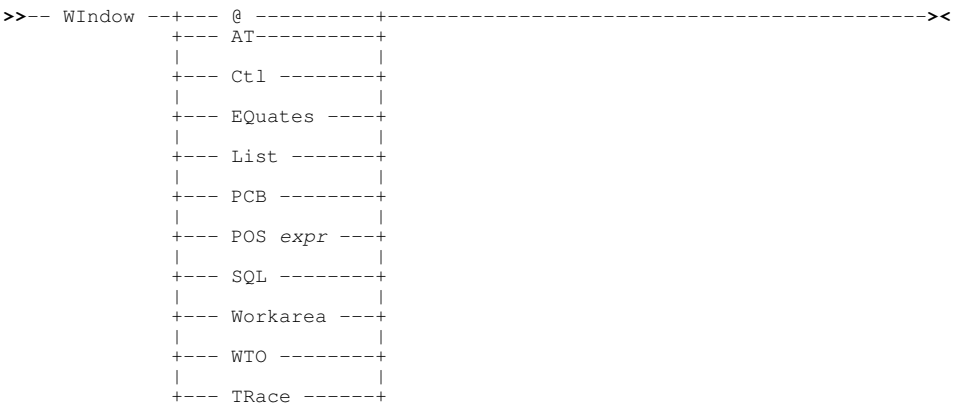
`wc /ret`
Locate the next occurrence of string "ret" in the Name column of the current watch list window.

`wcmd w1: hex on`
Set on the hex display of field and variable values within the watch list window named W1.

`wc MyWatch: movewindow to x=3 y=3`
Move the Watch list window named "MyWatch" to x,y co-ordinates (3,3).

WINDOW

Syntax:



Description:

Use the WINDOW command to open and place focus on the nominated window type.

Windows may also be opened via the Window menu of the SELCOPY Debug main window menu bar.

Parameters:

`@`
`AT`
: Open and place focus on the @ Pointer window.

`CTL`
: Open and place focus on the Control Cards window.
Note that, closing the Control Cards window also exits SELCOPY Debug.

`EQUATES`
: Open and place focus on the Equates window.

`LIST`
: Open and place focus on the Output Listing window.
Note that, closing the Output Listing window also exits SELCOPY Debug.

`PCB`
: Open the PCB window.

`POS expr`
: Open a POS window. A valid SELCOPY POS expression must be specified to define the start address of the storage display.

`SQL`
: Open the SQL Log window.

`WORKAREA`
: Open a storage window (Work Area window.)

`WTO`
: Open the WTO Log window.

`TRACE`
: Open and place focus on the Trace window.
Note that, closing the Trace window also exits SELCOPY Debug.

SELCOPY Debug SET/QUERY/EXTRACT Options

Syntax:

```
>>-+-----------+----- option_name ----- value -------------------------------><
   |           |
   +- SET -----+

>>--- Query --------- option_name -----------------------------------------><

                +--------------------+
                v                    |
>>--- EXTract ---+-- /option_name ---+---- / -------------------------------><
```

Description:

SELCOPY Debug environment options may be set, and their current values queried or extracted into stem-variables for use in REXX macros using the SET, QUERY and EXTRACT commands repectively.

Parameters:

option_name
: The SELCOPY Debug environment option(s). For EXTRACT, multiple options maybe requested at once by separating each with a blank or "/" (forward slash).

value
: For SET, the new value to be assigned for *option_name*.

BREAKIN - SET/QUERY/EXTRACT Option

Syntax:

```
>>-+-----------+-- BREAKIn ------ n_execs ----------------------------------><
   |           |
   +- SET -----+

>>--- Query ------ BREAKIn -------------------------------------------------><

>>--- EXTract --- /BREAKIn/ -------------------------------------------------><
```

Description:

This option controls the maximum number of times that any single control statement operation can be executed before the SELCOPY Debug loop break-in is activated and processing paused.

The initial value for BREAKIN is determined by the SELCOPYi INI variable SELCOPY.LoopBreakIn which gets updated when SET BREAKIN is executed.

SET Value:

n_execs
: The maximum number of times an operation will be executed before loop breakin is triggered.

QUERY Response:

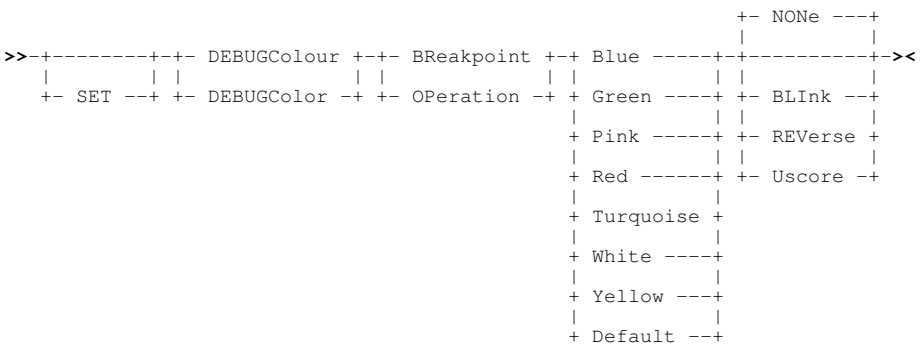
The current value of the BREAKIN option.

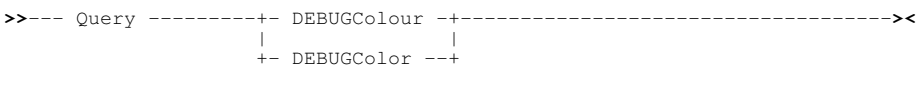
EXTRACT Rexx variables:

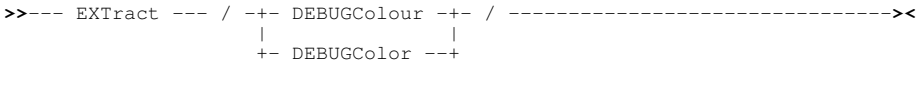
breakin.0	1
breakin.1	The current value of the BREAKIN option.

DEBUGCOLOUR, DEBUGCOLOR - SET/QUERY/EXTRACT Option

Syntax:







Description:

This option controls the colour display of areas within the SELCOPY Debug SYSIN input control statement window view.

SET Value:

`BREAKPOINT`
: Control statement operation text on which a break point has been set. Default colour is RED REVERSE.

`OPERATION`
: Control statement operation text to be executed next. Default colour is BLUE REVERSE.

`BLUE | GREEN | PINK | RED | TURQUOISE | WHITE | YELLOW | DEFAULT`
: Supported colours. If DEFAULT is specified, the default colour for the area is set.

`BLINK | REVERSE | USCORE | NONE`
: Extended highlighting of the specified field. The colour may blink, be displayed in reverse video or be underlined. Default is NONE.

QUERY Response:

For each SELCOPY Debug specific coloured area within the SYSIN window display, the display area name, current colour setting and extended highlighting option is displayed on a separate message line.

EXTRACT Rexx variables:

`debugcolor.0` `debugcolour.0`	Number of SELCOPY Debug specific areas within the SYSIN display for which a colour option may be assigned.
`debugcolor.i` `debugcolour.i`	One stem for each SELCOPY Debug area within the SYSIN display. The value of each compound variable is an upper case string containing the display area name, the current colour setting and extended highlighting option.

SELCOPY Debug WATCH Sub-commands

The SELCOPY Debug Watch List window has an independent command environment supporting a number of primary commands and environment options that are specific to Watch list display and operation.

Any command string, entered at the Watch list window command prompt and which starts with a primary command keyword that is not recognised by the Watch list command interpreter, will be passed to the SELCOPY Debug command interpreter instead.

Watch list commands may be submitted from the Watch List window or from any SELCOPY Debug window if prefixed with the SELCOPY Debug primary command keyword, WATCH.

Command	Description
BOttom	Scroll to the bottom of the display.
DELete	Delete an entry.
DOwn	Scroll down towards the bottom of the display.
CLOse	Close the Watch list window.
INSert	Insert a new entry.
LEft	Scroll a character entry's value to the left.
Locate	Scroll to a specific list entry.
POSWINdow	Open a POS storage window for the focus entry.
RESet	Reset entry line flags.
RIght	Scroll a character entry's value to the right.
SPace	Add a space line entry.
TOP	Scroll to the top of the display.
UP	Scroll up towards the top of the display.
WORKAREA	Add/replace a field entry for the work area.

BOTTOM

Syntax:

```
>>-- BOttom ------------------------------------------------------------------><
```

Description:

Display the last page of list data.
BOTTOM is functionally equivalent to DOWN MAX.

CLOSE

Syntax:

```
>>--+- CLOse ---+-------------------------------------------------------------><
    |           |
    +- END -----+
```

Description:

Close the Watch List window. The list entries are preserved and are included in the watch list display if it is re-opened.
CLOSE is assigned to F3 by default.

DELETE

Syntax:

```
>>-- DELete ------------------------------------------------------------------><
```

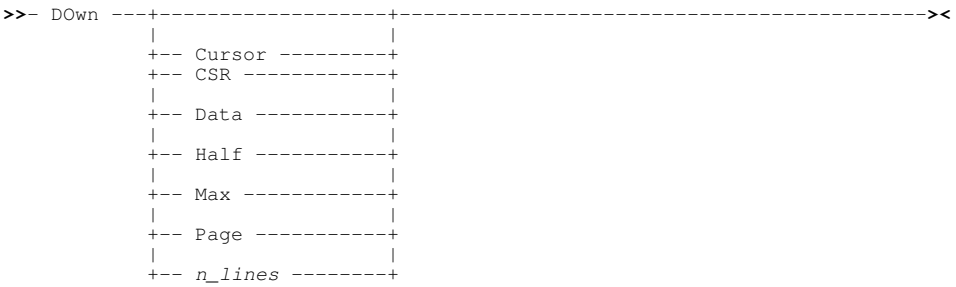
Description:

Delete (remove) the focus entry from the list. If the cursor is not position on an entry in the list, the focus entry is the first in the current client area display. Note that this is not necessarily be the first entry in the list.
DELETE is assigned to F17 by default.

Delete may also be actioned using line (prefix area) command "D" for an individual entry or "D*n*", "D*" or a "DD" pair for groups of entries.

DOWN

Syntax:



Description:

Scroll the view of the entries down towards the bottom of the list.

DOWN is assigned to **F8** by default. Any characters specified on the command line when the PFKey is hit will be concatenated to the command and treated as a parameter string.
Where no scrolling parameter is specified, the scroll amount will be the value specified in the "**Scroll>**" field.

Parameters:

CURSOR
CSR
: The list entry on which the cursor is positioned becomes the first line of the scrolled display. If the cursor is positioned outside the display area or on the first line within the display area, then DOWN PAGE is executed instead.

DATA
: Scroll down so that the last list entry in the current display area becomes the first entry of the scrolled display.

HALF
: Scroll down half a page of data. The list entry that is half way down the page in the current display area becomes the first entry of the scrolled display.

MAX
: Scroll down to display the last page of data. Where more than one page of data exists, the "End of List" line becomes the last line of the scrolled display. Otherwise, the "Top of List" line becomes the first line of the scrolled display.
Equivalent to the BOTTOM command.

PAGE
: Scroll down to display the next whole page of data. The list entry following the last entry of the current display area becomes the first line of the scrolled display.

n_lines
: Scroll down a specified number of lines. The list entry that is *n_lines* below the current entry becomes the first entry of the scrolled display.

INSERT

Syntax:

```
>>-- INSert ------------------------------------------------------------------><
```

Description:

Open the Add Watch Var/PosExp panel to insert a new variable, field definition or space entry following the focus entry in the Watch list. (i.e. The panel provides a dialog interface to the VARIABLE, POSITION and SPACE commands.)
INSERT is assigned to F18 by default.

LEFT

Syntax:

```
>>-- LEft ---+--------------------------+--+--------------------+--------------><
             |                          |  |                    |
             +-- VARiable -- varname --+  +-- Cursor ---------+
             |                          |  +-- CSR ------------+
             +-- Workarea -------------+  |                    |
                                          +-- Data -----------+
                                          |                    |
                                          +-- Half -----------+
                                          |                    |
                                          +-- Max ------------+
                                          |                    |
                                          +-- Page -----------+
                                          |                    |
                                          +-- n_cols ---------+
```

Description:

Scroll the view of the Value column text belonging to an individual Watch list entry left towards the first character of the text. Scrolling left and right is invalid for entries that are not of a character or hexadecimal data type.

Unless parameter keywords VARIABLE or WORKAREA are specified, left and right scrolling operate on the focus list entry. Note that the focus list entry is the entry on which the cursor is located, otherwise the first entry in the current display.

LEFT is assigned to **F10** by default. Any characters specified on the command line when the PFKey is hit will be concatenated to the command and treated as a parameter string.
Where no scrolling parameter is specified, the scroll amount will be the value specified in the "**Scroll>**" field.

If scrolling left would display characters before the start of the Value text, then the first character of the Value text becomes the first character of the scrolled display.

Parameters:

`CURSOR`
`CSR`
: The character within the list entry Value column on which the cursor is positioned becomes the last character of the scrolled display. If the cursor is positioned outside the Value text display area or is already on the last character of the displayed Value text, then LEFT PAGE is executed instead.

`DATA`
: Scroll left so that the first character of the current Value text display becomes the last character of the scrolled display.

`HALF`
: Scroll left a number of characters equal to half the width of the current Value text display. The character that is half way along the current Value text display becomes the last character of the scrolled display.

`MAX`
: Scroll left the maximum number of columns so that the first character of the Value text becomes the first character of the scrolled display.

`PAGE`
: Scroll left a number of characters equal to the width of the current Value text display. The character immediately before the first character of the current Value text display becomes the last character of the scrolled display.

`n_cols`
: Scroll left a specified number of characters. The character that is *n_cols* to the left of the last character of the current Value text display becomes the last character of the scrolled display.

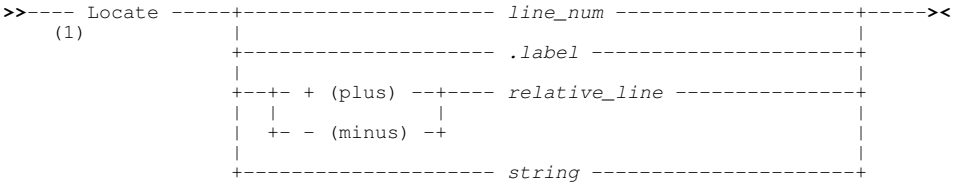
`VARIABLE` *`varname`*
: Specifies the list entry to be scrolled as the variable name entry identified by *varname*.

`WORKAREA`
: Specifies the list entry to be scrolled as the unique Work Area entry.

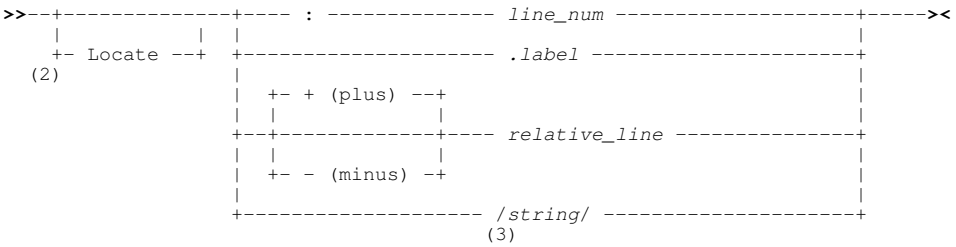
LOCATE

Syntax:

Interface ISPF



Interface XEDIT



Notes:

1. Provided the locate operand is not *string*, the ISPF syntax LOCATE command verb may be omitted in which case the XEDIT form of the LOCATE syntax is used instead.
2. Provided the locate operand begins with a non-alpha character, the XEDIT syntax LOCATE command verb may be omitted.
3. A *string* delimiter is optional if the LOCATE verb is specified. If present it must be a special character in the following list:
   ¬`¦"£$%^_-+={}[]:;~'#<>|\,?/

   The terminating delimiter is optional but if present must be the same as the starting delimiter.

Description:

Locate and scroll to a watch list entry that matches the locate criteria.

The format of the LOCATE syntax is based on the prevailing value of the Text Edit option INTERFACE, as identified by the Text Edit view containing the input control statements (SYSIN).

If *string* is used, it is compared with the Name value in each list entry that follows the current entry until a match is found. If End-of-List is encountered, the search wraps to the first entry in the list and continues until either a match is found or the current list entry line is reached.

The LOCATE command verb may be omitted. However, beware that the WATCH VARIABLE command will be executed instead if all the following conditions are true:

1. The LOCATE command syntax is specified as parameters on the SELCOPY debug WATCH command.
2. The LOCATE command verb is omitted.
3. The LOCATE operand matches the name of a SELCOPY variable.

Parameters:

line_num
: Locate the watch list entry assigned the specified line number *line_num*.
For Interface XEDIT, this number must be prefixed by the ":" (colon) symbol.

.label
: Locate the watch list entry assigned the specified label, *.label*. The preceding "." (dot/period) in *.label* is mandatory.

[+ | -] *relative_line*
: Locate the watch list entry that is a relative number of lines before (-) or after (+) the current list entry line.
For Interface XEDIT, the default relative line location operator is "+" (plus).
For Interface ISPF, specification of this operator is mandatory.

[/]*string*[/]
: Locate the watch list entry with a Name column value that matches the search *string* anywhere within the value's text. The search is not case sensitive and so lower case alpha characters in *string* will match upper case alpha characters in the Name value and vice versa.

If the LOCATE command verb is omitted, specification of a *string* delimiter, for example "/" (slash), is mandatory. This is so that the command is identified as being LOCATE.

For Interface ISPF, a *string* delimiter is not required and, if specified, will be treated as being part of the search string.

A terminating delimiter is optional and is required only if the search string contains blank characters. If specified, a terminating delimiter must be the same character symbol as the starting delimiter.

POSWINDOW

Syntax:

```
>>-- POSWINdow ------------------------------------------------------------><
```

Description:

Opens a SELCOPY Debug POS expression storage window for the focus watch list entry.

The value in the Name column is passed as the *expression* on the implicit WINDOW POS command. If the Name column value is that generated for a position entry (WATCH POSITION), then the "POS_" prefix is stripped from the expression.

POSWINDOW is assigned to F16 by default.

RESET

Syntax:

```
                          +- EXcluded --+
                          |             |
>>-- RESet --+---------+--+-------------+-------------------------------------><
             |         |
             +- ALL --+
```

Description:

Reset individual flags that heve set on for entries in the Watch list.

Parameters:

`ALL`
Equivalent to: `RESET EXCLUDED`

`EXCLUDED`
Reset the excluded flag for all list entries and so include previously excluded list entries in the Watch list display.

RIGHT

Syntax:

```
>>-- RIght --+-------------------------+--+--------------------+-------------><
             |                         |  |                    |
             +-- VARiable -- varname --+  +-- Cursor ---------+
             |                         |  +-- CSR ------------+
             +-- Workarea -------------+  |                   |
                                          +-- Data -----------+
                                          |                   |
                                          +-- Half -----------+
                                          |                   |
                                          +-- Max ------------+
                                          |                   |
                                          +-- Page -----------+
                                          |                   |
                                          +-- n_cols ---------+
```

Description:

Scroll the view of the Value column text belonging to an individual Watch list entry right towards the last character of the text. Scrolling left and right is invalid for entries that are not of a character or hexadecimal data type.

Unless parameter keywords VARIABLE or WORKAREA are specified, left and right scrolling operate on the focus list entry. Note that the focus list entry is the entry on which the cursor is located, otherwise the first entry in the current display.

RIGHT is assigned to **F11** by default. Any characters specified on the command line when the PFKey is hit will be concatenated to the command and treated as a parameter string.
Where no scrolling parameter is specified, the scroll amount will be the value specified in the "**Scroll>**" field.

If scrolling right would display characters beyond the end of the Value text, then the last character of the Value text becomes the last character of the scrolled display.

Parameters:

`CURSOR`
`CSR`
: The character within the list entry Value column on which the cursor is positioned becomes the first character of the scrolled display. If the cursor is positioned outside the Value text display area or is already on the first character of the displayed Value text, then RIGHT PAGE is executed instead.

`DATA`
: Scroll right so that the last character of the current Value text display becomes the first character of the scrolled display.

`HALF`
: Scroll right a number of characters equal to half the width of the current Value text display. The character that is half way along the current Value text display becomes the first character of the scrolled display.

`MAX`
: Scroll right the maximum number of columns so that the last character of the Value text becomes the last character of the scrolled display.

`PAGE`
: Scroll right a number of characters equal to the width of the current Value text display. The character immediately after the last character of the current Value text display becomes the first character of the scrolled display.

`n_cols`
: Scroll right a specified number of characters. The character that is *n_cols* to the right of the first character of the current Value text display becomes the first character of the scrolled display.

`VARIABLE` *`varname`*
: Specifies the list entry to be scrolled as the variable name entry identified by *varname*.

`WORKAREA`
: Specifies the list entry to be scrolled as the unique Work Area entry.

SPACE

Syntax:

```
>>-- SPace ------------------------------------------------------------------><
```

Description:

Insert a blank line in the Watch list following the focus list entry.
SPACE is assigned to F20 by default.

TOP

Syntax:

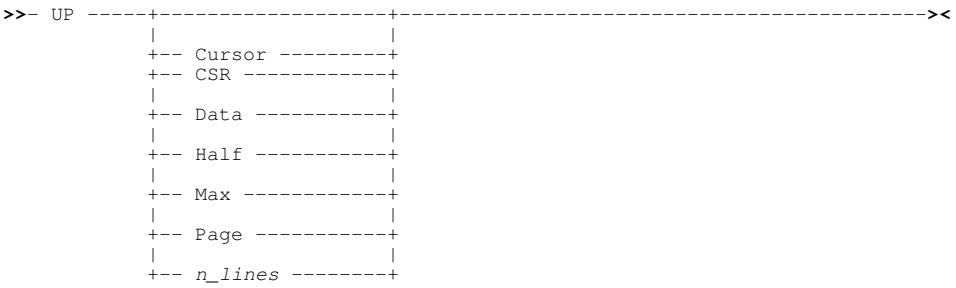
```
>>-- TOP -------------------------------------------------------------------><
```

Description:

Display the first page of list data.
TOP is functionally equivalent to UP MAX.

UP

Syntax:



Description:

Scroll the view of the entries up towards the top of the list.

UP is assigned to **F7** by default. Any characters specified on the command line when the PFKey is hit will be concatenated to the command and treated as a parameter string.
Where no scrolling parameter is specified, the scroll amount will be the value specified in the "**Scroll>**" field.

Parameters:

CURSOR
CSR
: The list entry on which the cursor is positioned becomes the last line of the scrolled display. If the cursor is positioned outside the display area or on the last line within the display area, then UP PAGE is executed instead.

DATA
: Scroll up so that the first list entry in the current display area becomes the last entry of the scrolled display.

HALF
: Scroll up half a page of data. The list entry that is half way down the page in the current display area becomes the last entry of the scrolled display.

MAX
: Scroll up to display the first page of data. The "Top of List" line becomes the first line of the scrolled display. Equivalent to the TOP command.

PAGE
: Scroll up to display the next whole page of data. The list entry before the first entry of the current display area becomes the last entry of the scrolled display.

n_lines
: Scroll up a specified number of lines. The list entry that is *n_lines* above the current entry becomes the first entry of the scrolled display.

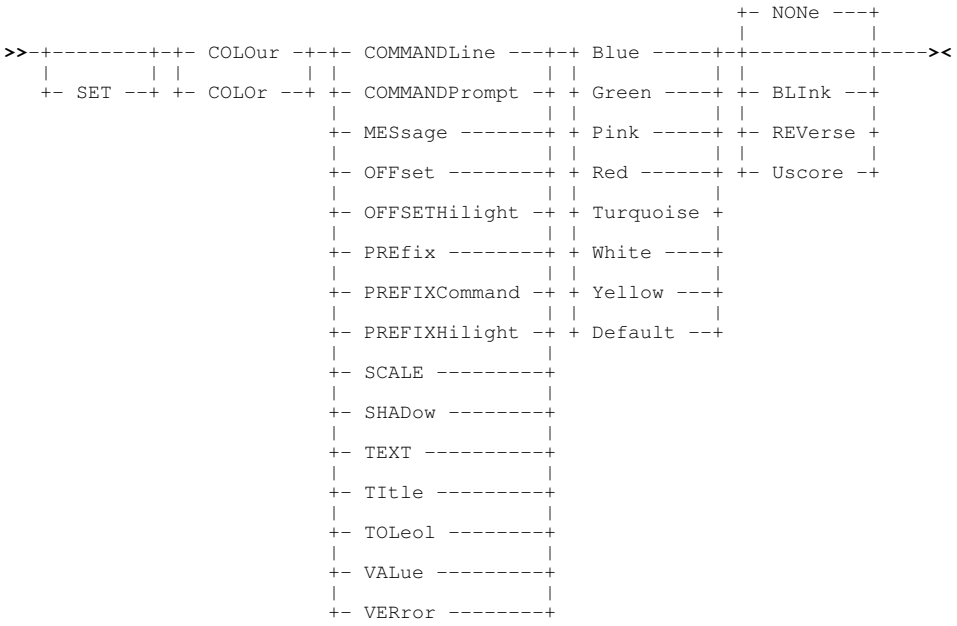
SELCOPY Debug WATCH SET/QUERY/EXTRACT Options

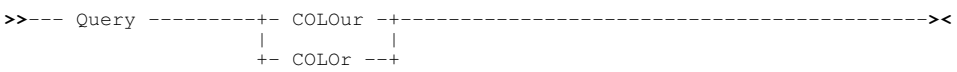
SELCOPY Debug Watch list window specific options may be set, and their current values queried or extracted into stem-variables for use in REXX macros using the SET, QUERY and EXTRACT commands repectively.

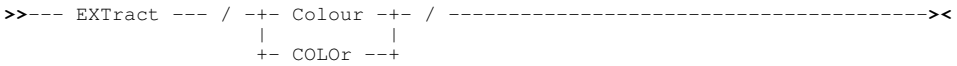
For each option, specification of the SET command keyword is optional unless the SET operation is a parameter of the SELCOPY Debug WCOMMAND command, in which case its specification is mandatory.

COLOUR, COLOR - SET/QUERY/EXTRACT Watch List Option

Syntax:







Description:

This option controls the colour of watch list display area attribute highlighting.

SET Value:

COLUMN
: Watch list "Col" column offset value area. Default colour is RED NONE.

COMMANDLINE
: Command line input text. Default colour is GREEN NONE.

COMMANDPROMPT
: Command and Scroll prompts. Default colour is BLUE NONE.

MESSAGE
: Message line text. Default colour is RED NONE.

PREFIX
: Prefix area. (PREFIX ON) Default colour is GREEN NONE.

PREFIXCOMMAND
: Prefix area (line) command text. Default colour is RED NONE.

PREFIXHILIGHT
: Prefix area (line) command text in error. Default colour is RED NONE.

`SCALE`
: Scale line. (SCALE ON) Default colour is WHITE NONE.

`SHADOW`
: Excluded shadow line. (SHADOW ON) Default colour is WHITE NONE.

`TEXT`
: All other window text. Default colour is BLUE NONE.

`TITLE`
: Column header text. Default colour is WHITE USCORE.

`TOLEOL`
: Top of List/End of List lines. Default colour is WHITE NONE.

`VALUE`
: Watch list "Value" column text. Default colour is GREEN NONE.

`VERROR`
: Watch list "Value" column error text. Default colour is RED REVERSE.

`BLUE | GREEN | PINK | RED | TURQUOISE | WHITE | YELLOW | DEFAULT`
: Supported colours. If DEFAULT is specified, the default colour for the area is set.

`BLINK | REVERSE | USCORE | NONE`
: Extended highlighting of the specified field. The colour may blink, be displayed in reverse video or be underlined. Default is NONE.

QUERY Response:

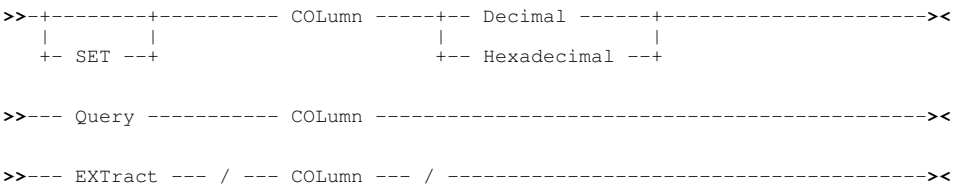
For each specific coloured area within the Watch list window display, the display area colour attribute name, current colour value and its extended highlighting option is displayed on a separate message line.

EXTRACT Rexx variables:

`color.0` `colour.0`	Number of SELCOPY Debug specific colour attribute areas within the Watch list display for which a colour option may be assigned.
`color.i` `colour.i`	One stem for each colour attribute area within the Watch list display. The value of each compound variable is an upper case string containing the display area name, the current colour setting and extended highlighting option.

COLUMN - SET/QUERY/EXTRACT Watch List Option

Syntax:



Description:

This option controls whether the number in the Watch list "Col" column is displayed as a decimal or hexadecimal value. The "Col" value represents the current offset of character text displayed in the "Value" column.

SET Value:

`DECIMAL | HEXADECIMAL`
: Value is displayed as decimal or hex. Default is decimal.

QUERY Response:

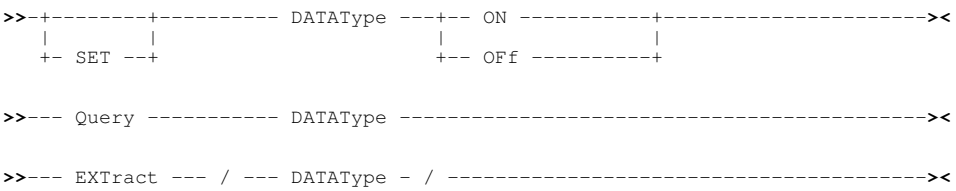
The current value of the COLUMN option (DEC or HEX).

EXTRACT Rexx variables:

`column.0`	1
`column.1`	The current value of the COLUMN option (DEC or HEX).

DATATYPE - SET/QUERY/EXTRACT Watch List Option

Syntax:



Description:

This option controls whether or not the "DataType" column is included in the Watch list display.

SET Value:

`ON | OFF`
: Display of the DataType column is on or off.

QUERY Response:

The current value of the DATATYPE option (ON or OFF).

EXTRACT Rexx variables:

`datatype.0`	1
`datatype.1`	The current value of the DATATYPE option (ON or OFF).

HEX - SET/QUERY/EXTRACT Watch List Option

Syntax:

```
>>-+--------+---------- HEX --------+-- ON -----------+----------------------><
   |        |                       |                 |
   +- SET --+                       +-- OFf ----------+

>>--- Query ----------- HEX ------------------------------------------------><

>>--- EXTract --- / --- HEX ------ / ---------------------------------------><
```

Description:

This option controls whether or not the "Value" column includes the value's source field displayed in hex. If set on, 2 additional lines displaying hex digits in up/down notation are included for each list entry.

SET Value:

`ON | OFF`
: Display of up/down hex is on or off.

QUERY Response:

The current value of the HEX option (ON or OFF).

EXTRACT Rexx variables:

`hex.0`	1
`hex.1`	The current value of the HEX option (ON or OFF).

POINT - SET/QUERY/EXTRACT Watch List Option

Syntax:

```
                                      +- ON --+
                                      |       |
>>-+-----------+--- Point --- .name ---+-------+-----------------------------><
   |           |                      |       |
   +- SET -----+                      +- OFF -+

>>--- Query ------- Point -- * ----------------------------------------------><

>>--- EXTract ---- /Point -- * -- / ----------------------------------------><
```

Description:

Assign or unassign a label name to the focus watch entry line for subsequent reference. (e.g. on command LOCATE)

If SET is omitted, the minimum abbreviation for POINT is POI. ANy further abbreviation would execute the POSITION primary command.

A watch list entry line may be assigned only one label name. The same name may not be assigned to more than one line in the list. When a label name is set, any existing label with the same name will be unassigned from its current list entry line and reassigned to the focus line.

Label names may also be assigned by entering *.name* against the required line in the watch list prefix area.

Set Options:

`.name`
: A label name to be unassigned or assigned to the focus line. The specified name may be of any length, may contain and begin with any alphanumeric or special character, but must be preceeded by a "." (dot/period).

`OFF`
: Unassign the specified label name from its list entry line.

QUERY Response:

QUERY POINT * displays all assigned lebl names and their watch list line numbers.

EXTRACT Rexx variables:

point.0	Number of lines within the watch list list to which a label name is assigned.
point.i	Name and line number of the ith label within the watch list. Point variables identify labels in the order in which they occur in the watch list.

PREFIX - SET/QUERY/EXTRACT Watch List Option

Syntax:

```
>>-+--------+-------- PREfix -----+-- ON ---+--+----------+--+-----------+----><
   |        |                     |         |  |          |  |           |
   +- SET --+                     +-- OFf --+  +- Left --+  +- n_bytes -+
                                               |         |
                                               +- Right -+

>>--- Query ----------- PREfix ------------------------------------------------><

>>--- EXTract --- / --- PREfix --- / ----------------------------------------><
```

Description:

This option defines whether or not the prefix area is displayed in the Watch list view and, if so, whether it is displayed on the left or right of the window view and the number of columns it occupies.

The prefix displays the list line number, and is also where prefix area (line) commands may be entered.

SET Value:

`ON | OFF`
: The prefix area is set displayed (ON) or hidden (OFF).

`LEFT | RIGHT`
: Determines whether the prefix area is displayed on the left or right of the list entry data.

`n_bytes`
: The width of the prefix area.

QUERY Response:

The current setting of the PREFIX option, **ON** or **OFF**, followed by **LEFT** or **RIGHT** and the length **n_bytes**.

EXTRACT Rexx variables:

prefix.0	3
prefix.1	The current setting of the PREFIX option, **ON** or **OFF**.
prefix.2	The current position of the PREFIX option, **LEFT** or **RIGHT**.
prefix.3	The current length of the PREFIX area.

SCALE - SET/QUERY/EXTRACT Watch List Option

Syntax:

```
>>-+--------+---------- SCALe ------+-- ON -----------+----------------------><
   |        |                       |                 |
   +- SET --+                       +-- OFf ----------+

>>--- Query ----------- SCALe ------------------------------------------------><

>>--- EXTract --- / --- SCALe ---- / ---------------------------------------><
```

Description:

This option controls the display of a counting scale line above a character or hex type value in the "Value" column. SCALE ON and OFF applies to the focus watch list entry but will do nothing if this entry is of numeric or hexadecimal data type.

SET Value:

`ON | OFF`
: The scale line is displayed (ON) or hidden (OFF) for the focus entry.

QUERY Response:

The current setting of the SCALE option for the focus Watch list entry, **ON** or **OFF**.

EXTRACT Rexx variables:

`scale.0`	1
`scale.1`	The current setting of the SCALE option for the focus Watch list entry, **ON** or **OFF**.

SHADOW - SET/QUERY/EXTRACT Watch List Option

Syntax:

```
>>-+--------+---------- SHADow -----+-- ON -----------+----------------------><
   |        |                       |                 |
   +- SET --+                       +-- OFf ----------+

>>--- Query ----------- SHADow ------------------------------------------------><

>>--- EXTract --- / --- SHADow --- / ---------------------------------------><
```

Description:

This option controls the display of a shadow line in place of one or more consecutive Watch list entries that have been excluded from the display.

SET Value:

`ON | OFF`
: The shadow lines are displayed (ON) or hidden (OFF).

QUERY Response:

The current setting of the SHADOW option, **ON** or **OFF**.

EXTRACT Rexx variables:

`shadow.0`	1
`shadow.1`	The current setting of the SHADOW option, **ON** or **OFF**.

SELCOPY Debug Function Keys

You can assign 3270 Program Function Keys (PFKeys) to primary commands. The KEYS line command may be used to display and assign function key values for SELCOPY Debug windows of the same keylist name.

SELCOPY Debug PFKeys have default functions assigned as determined by the keylist associated with the particular window. Of particular use in storage display windows, list type windows, the Watch list window and the SYSIN control statement input window (with debug keys active), the following default PFkey definitions are assigned:

PFKey	Action	Window	Description
PF13	StepOver	All	Execute the next SELCOPY operation (step over a sub-routine).
PF14	StepInto	All	Execute the next SELCOPY operation (step into a sub-routine).
PF15	Go	All	Continue SELCOPY processing to next breakpoint or EOJ.
PF16	SDBPopUp	SYSIN	Display the Point-and-Shoot Popup Menu.
	PopUp	Storage	Display the storage window popup menu.
	PosWindow	Watch List	Display a POS Expression window for the focus Watch List entry.
PF17	Delete	Watch List	Delete a Watch List entry.
PF18	Insert	Watch List	Open the Add WATCH Var/PosExp panel to insert a Watch List entry.
PF19	BreakPoint	SYSIN	Toggle a break point on and off at the focus operation.
	Scale	Watch List	Display the scale line above the value of the focus Watch list entry.
PF20	Space	Watch List	Insert a blank (spacer) Watch List entry after the focus entry.

Utilities Menu (=8)

The Utilities Menu panel (ZZSGUTIL) is an interactive panel window opened on selection of option 8. in the SELCOPY/i Primary option menu.

SELCOPY/i supports a number of general purpose utilities and interfaces to a selection of system utilities that may be accessed via this panel.

Options

1 SELCOPY/debug	**SELC**	SELCOPY/batch language interactive debug
2 CBLVCAT	**VCAT**	Catalog/VTOC report online excution
3 IDCAMS	**AMS**	Execute IDCAMS commands interactively
4 Catalog ALIAS	**AMSA**	Define new Catalog Alias
5 Library ALIAS	**ALI**	Create new PDS/PDSE library member Alias
6 IEBCOPY	**IEBC**	Execute IEBCOPY interactively
7 Favourites	**FAV**	Favourite Datesets/Commands
8 System	**SY**	Display System Information
9 Search	**FS**	Basic PDS/PDSE Library string search
10 Find Lib Member(s)	**LLX**	Search for member(s) across multiple libraries
11 Compare Files	**COMPF**	Compare Files
12 Compare Libraries	**COMPL**	Compare Libraries
13 Calendar	**CAL**	Basic Calendar
14 Calculator	**CALC**	REXX expression calculator
15 Alloc/Define		Create new VSAM or Sequential datasets
16 XML-Gen	**XML**	Produce eXtended Markup Language from a Data File
17 CSV-Gen	**CSV**	Produce Comma Separated Variables from a Data File
18 JSON-Gen	**JSON**	Produce JavaScript Object Notation from a Data File
19 Merge Datasets	**MERGE**	Merge a number datasets sorted by a key field

CBLVCAT Interactive (VCI) (=8.2)

If a valid software licence key has been applied, SELCOPY/i may invoke the CBLVCAT program to allow interactive execution within a SELCOPY/i window.

1. CBLVCAT Interactive Window
2. Raw Data Window

CBLVCAT Interactive Window

The Execute CBLVCAT window is used to execute CBLVCAT Interactive and may be opened via the following:

- Select option 2. 'CBLVCAT' from the Utilities Menu
- Select 'CBLVCAT Interactive' from the File menu in the CBLe main window menu bar.
- Enter the command VCAT on the command line of any window.
- Enter the "T" or "VC" prefix command in the prefix area of an existing Execute CBLVCAT window or certain other List type windows. "T" will generate a CBLVCAT Tune report and IDCAMS DEFINE deck for a VSAM file, "VC" will generate a CBLVCAT catalog and/or VTOC report for the list entry.

CBLVCAT is used to generate standard and customised reports on VTOC and ICF/VSAM catalog data. It also supports VSAM file tuning and generation of IDCAMS DEFINE job source.
Details on CBLVCAT output and control statement syntax is found in the CBLVCAT User Manual.

SELCOPY/i loads CBLVCAT and assumes control over its control statement input and report output functions. This allows the user to specify CBLVCAT input statements directly at the VCAT Command prompt or indirectly via a control statement file and view the output in a window.

In order to direct input from a control statement file, the fileid should be entered at the VCAT Command prompt and prefixed with a "<" (less than) symbol.

If you are using the LISTVCAT DEFINE option then the generated IDCAMS control statements are displayed in a CBLe text edit window and may subsequently be saved to a file.

After execution of CBLVCAT control statements (or control statement file), the SYSPRINT (MVS and CMS) or SYSLST (VSE) output is presented in the display area of the Execute CBLVCAT window.

The Execute CBLVCAT window display area is a list window with a single column (i.e. SysPrint) and so has charactersitics defined by the list window class. For example, the Execute CBLVCAT window supports Prefix Commands and filtering, to display new views of the data.

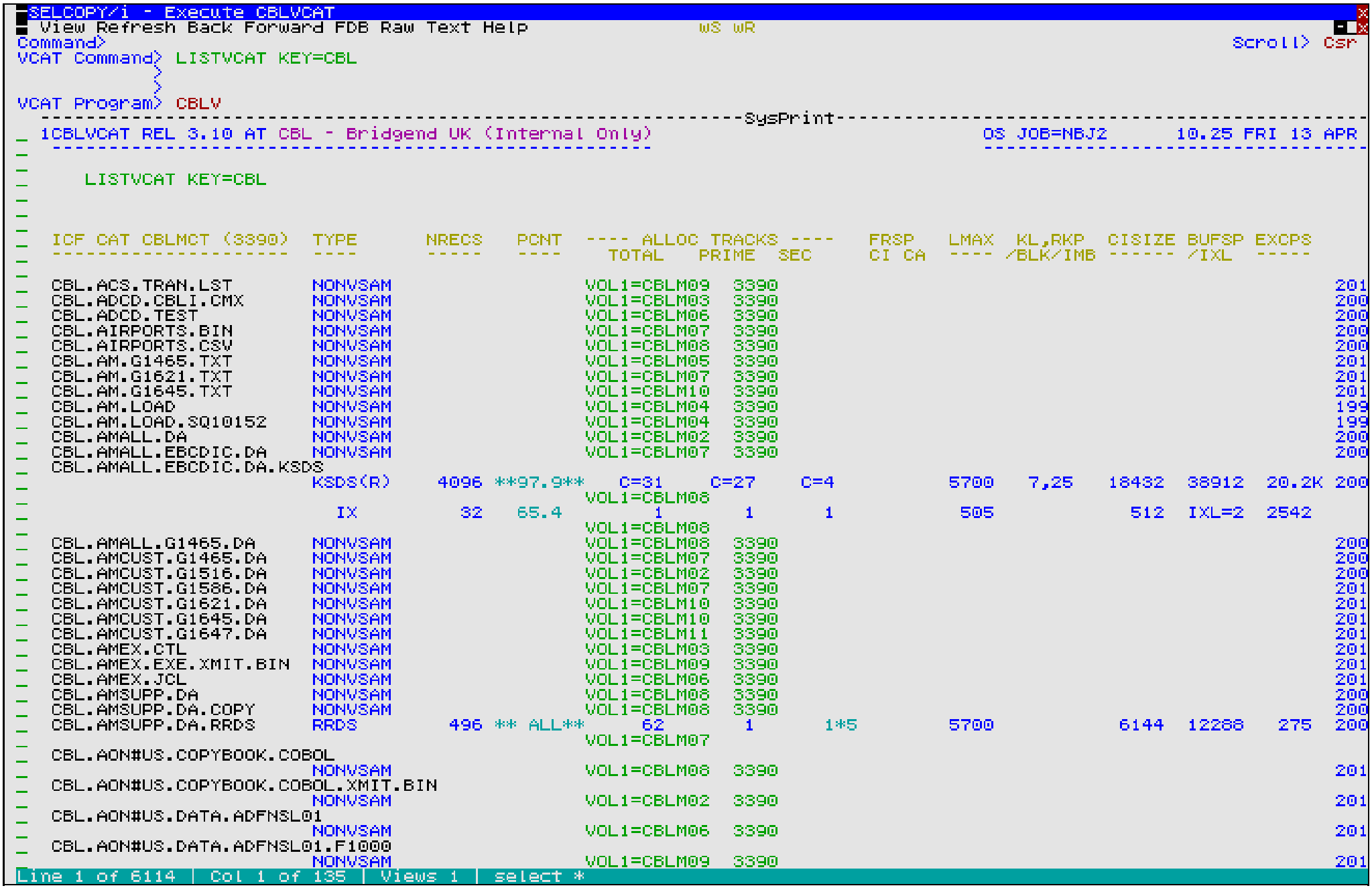


Figure 131. CBLVCAT Interactive in 58x120 3270 Session.

In addition to the standard List window menu items, the Execute CBLVCAT window includes the menu item **RAW** to open the CBLVCAT LISTVCAT or LISTVTOC Raw Data window.

CBLVCAT Log Output window is opened only if the CBLVCAT execution has generated SYSLOG output. This usually occurs if an error has been encountered in which case an information window is also displayed.

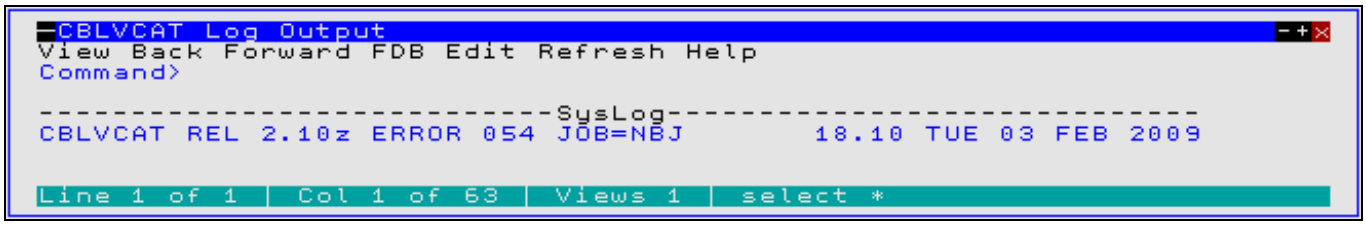


Figure 132. CBLVCAT Log Output window.

Panel Input Fields

`VCAT Command>`

Enter one of the following:

- ◊ A CBLVCAT command, as you would code it on a CBLVCAT control statement.
- ◊ <filename, where filename is the name of a CBLVCAT control statement file.

The CBLVCAT command syntax is described in the CBLVCAT User Manual.

`VCAT Program>`

Specify the name of the CBLVCAT executable MODULE (MVS) or PHASE (VSE). By default, this field contains CBLV.

Prefix Commands

The following prefix area commands are available:

Command	Description
<Dflt>	See **Note 1** below.
AS	Open an Associations list window to list associated objects for this entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
F	Open the FSU - File Search/Update Window to perform advanced file search and optionally update.
FO	Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
FS	If the entry is a PDS/PDSE, open the file search window for the PDS.
I	Open a Data Set Information panel display for the entry.
ID	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
K	Delete (Kill) the entry without prompting for verification.
L	Open a Dataset List window for the entry.
M	If the entry is a PDS/PDSE, open a Library List window.
Q	Open a Dataset Enqueue List window for the entry (major name SYSDSN.)
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
T	Open another Execute CBLVCAT window and issue a LISTVCAT with TUNE DEFINE to generate tuned output for the entry.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open another Execute CBLVCAT window and issue a LISTVCAT and/or LISTVTOC operation (as appropriate) for the entry. See **Note 2**.
Z	Perform a compress of an MVS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
?	Open the volume statistics window for the volume containing the entry. Note that this command will only be successful for lines of a LISTVCAT report containing VOLn=volser.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Notes:

1. The default action on hitting <Enter> or, if configured, double-clicking the left mouse button on a SysPrint line depends on the contents of the report entry, as follows:

1. If the entry contains the TYPE field "USERCAT" or "ALIAS OF", then prefix command "VC" is default.
2. If the entry contains the TYPE field "PDS" or "PDSE", then prefix command "M" (Member List) is default.
3. If the entry contains a fileid, then prefix command "E" (Edit) is default.

2. The "VC" prefix command performs LISTVCAT/VTOC operations based on the contents of the entry fields, as follow:

    1. If the entry contains the TYPE field "USERCAT", then a new report is generated for the entire contents of that catalog. Otherwise, only list the catalog entries that match the fileid.
    2. If the report entry also contains a "VOLn=volser" field, then generate a LISTVTOC report for entries that match the fileid in the volume's VTOC.

Columns Displayed

Name	Type	Description
SysPrint	Char	VCAT output report line

Raw Data Window

The CBLVCAT Raw window may be opened via the following:

- Select 'Raw' from the menu item of the Execute CBLVCAT window.
- Enter the command LVR on the command line of any window.

Where CBLVCAT arranges data in a printable report format, the CBLV Raw Data window provides a list of all report field data accumulated by CBLVCAT in order to generate the report.

The CBLVCAT Raw window has the same characteristics as a SELCOPY/i List window including selecting, sorting and filtering of row and column data and "point and shoot" sorting on column headers.

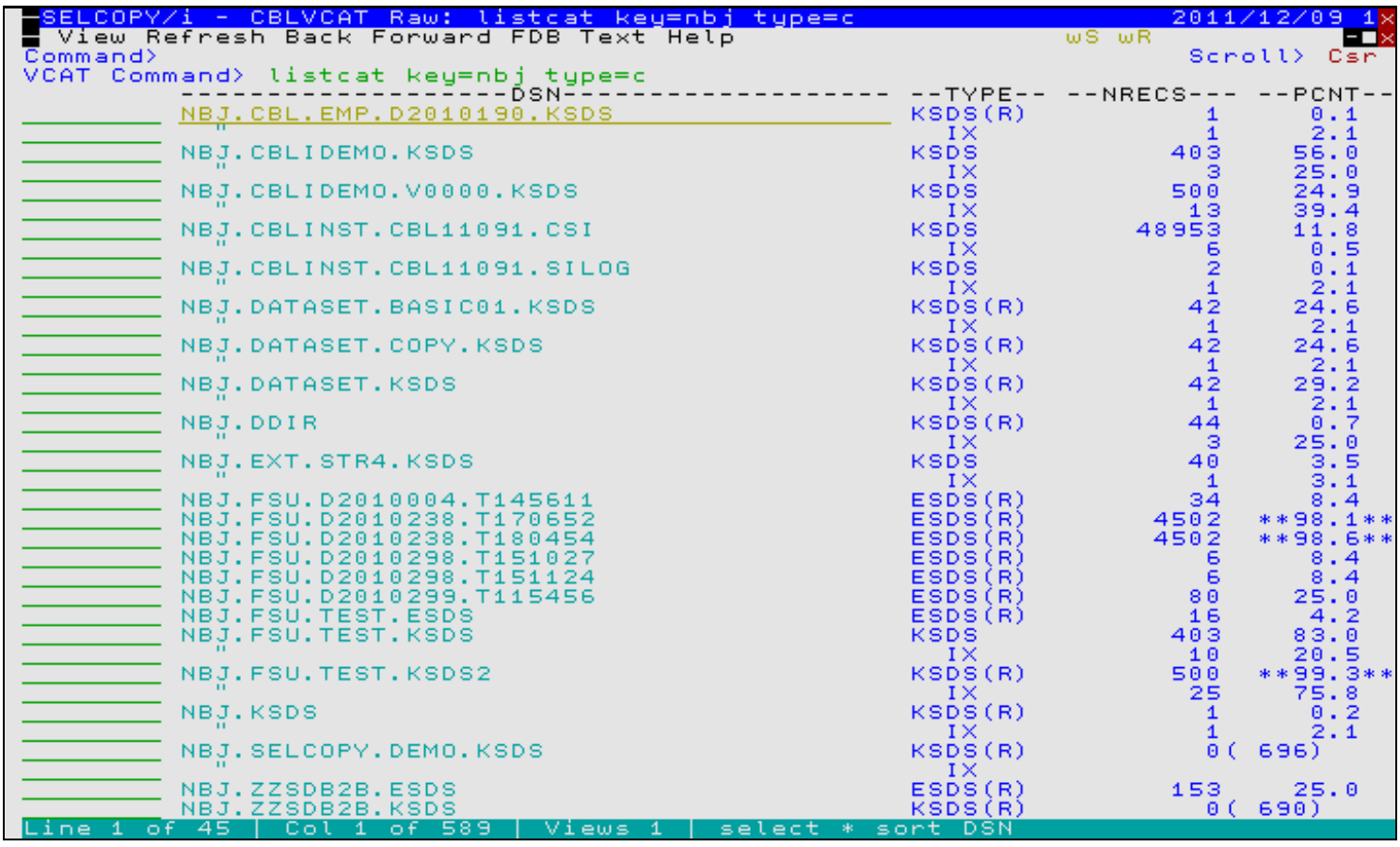


Figure 133. CBLVCAT LISTVCAT Raw Data Window.

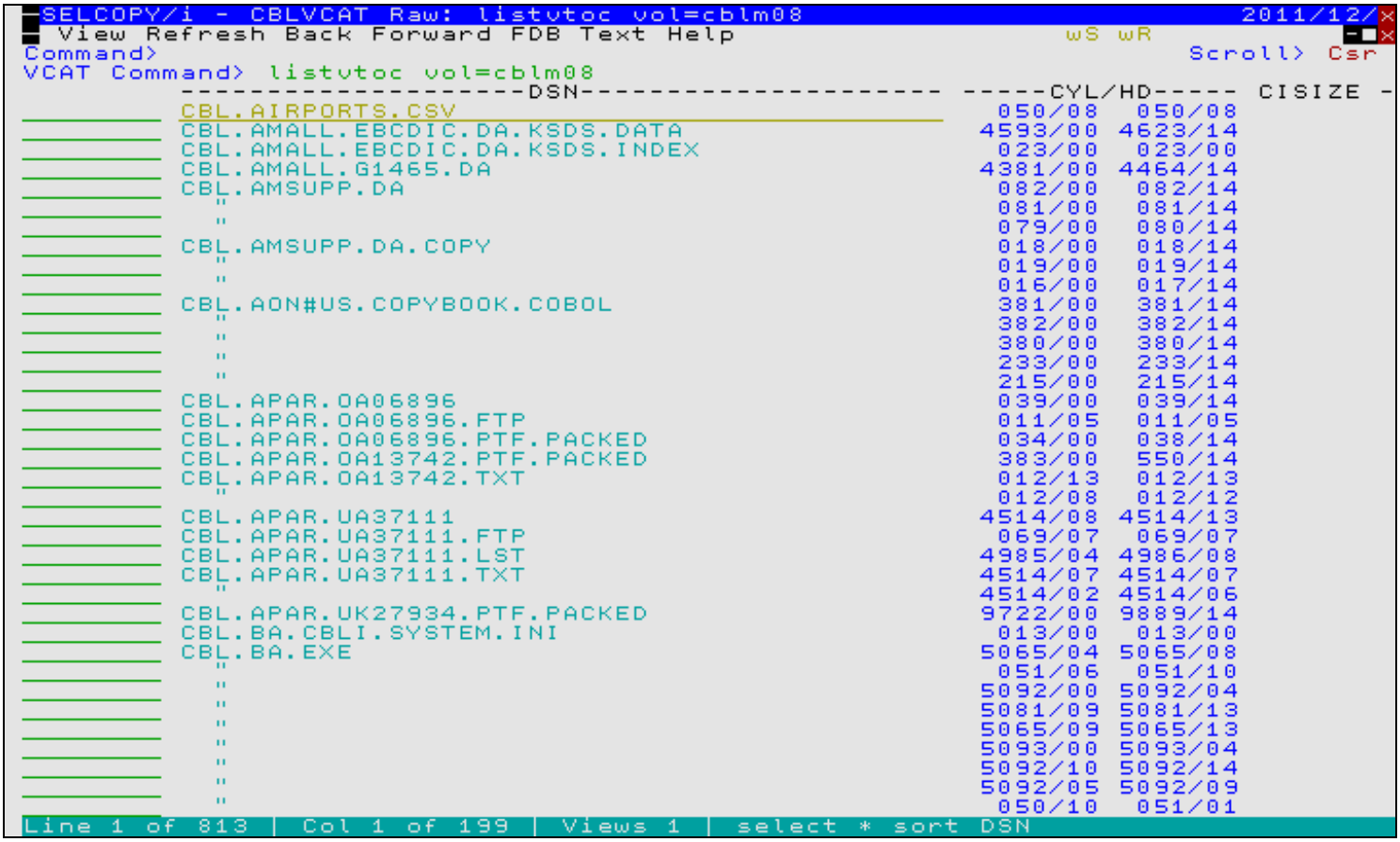


Figure 134. CBLVCAT LISTVTOC Raw Data Window.

Prefix Line Commands

The following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command "M" if entry is a PDS/PDSE library, prefix line command "E" otherwise.
AS	Open an Associations list window to list associated objects for this entry.
AP	Open the DB2 Print Audit Report panel for this entry.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry. (Default for non-PDS/PDSE entries)
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FO	Open an SDE view to display (browse) the entry as FSU - File Search/Update Window output.
FS	Open the File Search window for the entry.
I	Open a Data Set Information panel display for the entry.
ID	Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.
K	Delete (Kill) the entry without prompting for verification.
M	If the entry is a PDS/PDSE, open a Library List window. (Default for PDS/PDSE entries)
Q	List dataset enqueues (major name SYSDSN) for this entry.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse/edit the entry in a Data Editor view.
T	Open an Execute CBLVCAT window and issue a LISTVCAT TUNE DEFINE operation for the entry.
UT	Opens the general file utilities menu to generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
VC	Open an Execute CBLVCAT window and issue a LISTVCAT operation for the entry.
Z	Perform a compress of an MVS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.

/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

For LISTVTOC Output, the data displayed is:

Name	Type	Description
DSN	Char	Data Set Name
CYL/HD	Char	Low and high Cylinder/Head Limits
CISIZE	Char	Control Interval Size
START	Char	Relative track/block start address
ALLOC	Char	Number of allocated tracks/blocks
USED	Char	Number of used tracks/blocks
TYPE	Char	Data set type
EXPIRES	Char	Expiry date
BLKSIZE	Char	Blocksize
LRECL	Char	Logical Record Length
RECFM	Char	Record Format
CREATED	Char	Creation date
INFO	Char	Informational messages
VOLUME	Char	VTOC volume id
ACCESSED	Char	Last Accessed date
UNIT	Char	Unit (cuu) of DASD volume

For LISTVCAT Output, the data displayed is:

Name	Type	Description
DSN	Char	Data Set Name
TYPE	Char	Data set type
NRECS	Char	Number of records
PCNT	Char	Percent of allocated space used
ALLOCT	Char	Total allocated tracks/blocks
ALLOCU	Char	Unused allocated tracks/blocks
ALLOCP	Char	Defined Primary allocation (tracks/blocks)
ALLOCS	Char	Defined Secondary allocation (tracks/blocks)
FRSP	Char	Defined Free Space per CI and CA
LMAX	Char	Defined Maximum Record Length
KL/BLK/IMB	Char	Duplicate of fields KL,RKP **or** BLKSIZE **or** IMB/REP
CISIZE	Char	Control Interval Size
BUFSP/IXL	Char	Duplicate of fields BUFSP **or** IXL
EXCPS	Char	Number of EXecuted Channel Programs
TIMESTMP	Char	Timestamp that file was last closed
NSEC	Char	Number of secondary extents
AVRL	Char	Defined average RECORDSIZE
PHYREC	Char	Physical Record Size
RECSTATS	Char	Records deleted, inserted, updated and read
KL	Char	Defined KEYS Length
RKP	Char	Defined KEYS Position
BLKSIZE	Char	Block size (VSE/VSAM SAM)
IMB/REP	Char	IMBED and/or REPLICATE flags
BUFSP	Char	Defined BUFFERSPACE
IXL	Char	Number of Index Levels
CI/CA	Char	Number of Control Intervals per Control Area
SHR	Char	Defined SHAREOPTIONS (local and cross system)
S/C	Char	Defined SHAREOPTIONS (local only) and USECLASS (primary only)
DEFINED	Char	Date on which file was defined

EXPIRES	Char	Date on which file expires
SPLITCI	Char	Number of Control Interval splits
SPLITCA	Char	Number of Control Area splits
SEVL	Char	Highest CBLVCAT severity message level
VOLUME	Char	Catalog Volume
GMAX	Char	GDG Maximum Level
GVER	Char	GDG Version number
GGEN	Char	GDG Generation number
STD1	Char	Reserved (blank)
STD2	Char	Reserved (blank)
HIUSERBA	Char	High Used Relative Byte Address
HIALLRBA	Char	High Allocated Relative Byte Address
FREEBYTES	Char	Number of unused allocated bytes
COMPONENT	Char	DATA or INDEX component DSN
ENTRY	Char	VSAM CLUSTER entry DSN
SMSS	Char	Defined SMS Storage Class
SMSD	Char	Defined SMS Data Class
SMSM	Char	Defined SMS Management Class
EXT	Char	Extended Attributes
CATALOG	Char	Catalog DSN

Execute IDCAMS (=8.3)

The IDCAMS Command window may be opened via the following:

- Select option 3. 'IDCAMS' from the Utilities Menu
- Select 'Execute IDCAMS' from the File menu in the CBLe main window menu bar.
- Enter the command AMS on the command line of any window.
- Enter the prefix command "I" where supported by a List type window.

The IDCAMS Command window allows the user to enter any IDCAMS command and view the output in the the window display area.

The IDCAMS Command window is essentially a List window and has the same charactersitics as List windows. For example filtering is supported to display new views of the data.

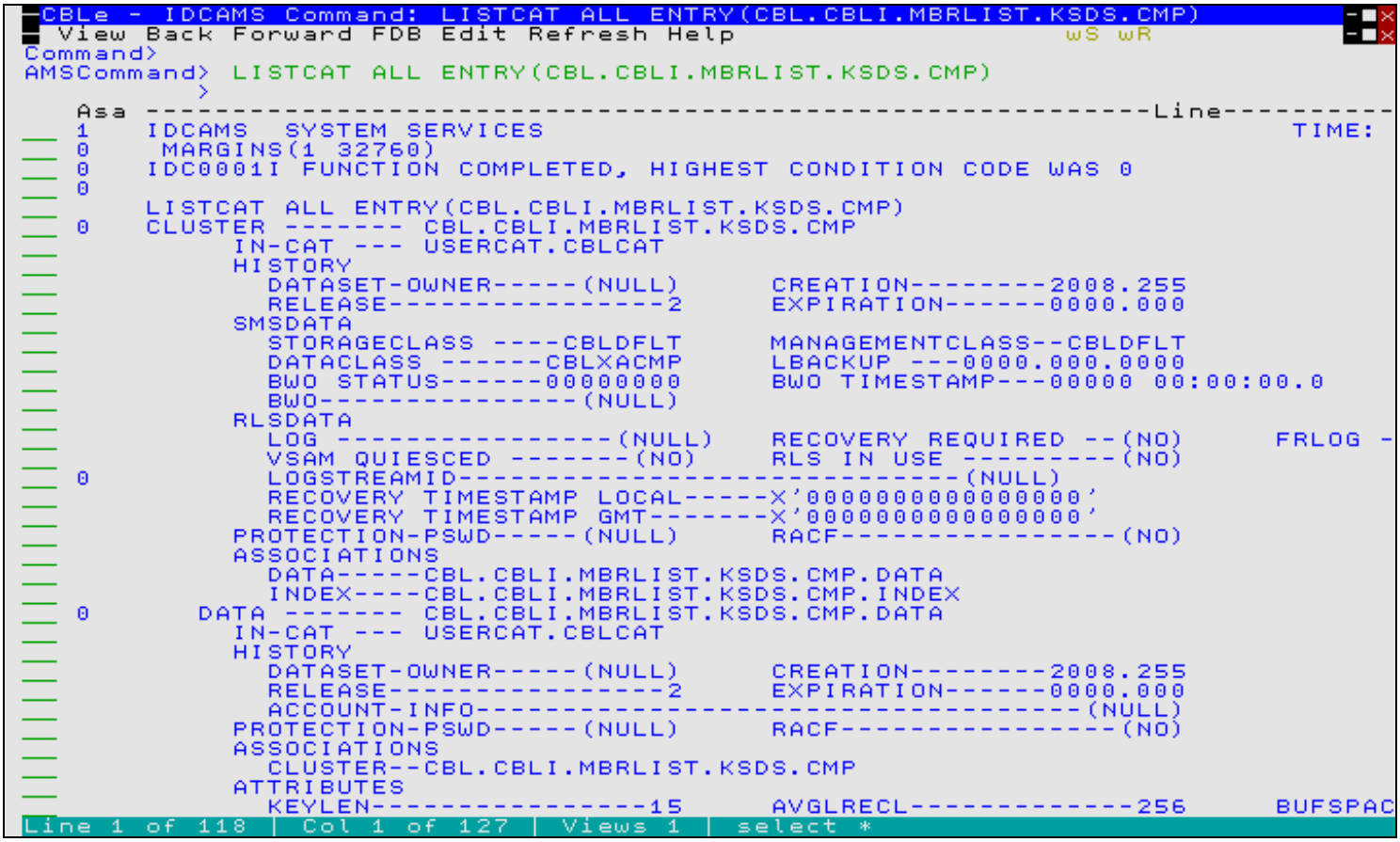


Figure 135. IDCAMS Command window.

Panel Fields

AMSCommand>
Specify valid IDCAMS command syntax.

Prefix Commands

No prefix line commands are supported for IDCAMS Command windows.

Columns Displayed

Name	Type	Description
Asa	Char	ASA print control character
Line	Char	Print line

Execute POWER

The POWER Command Output window may be opened via the following:

- Select 'Execute POWER' from the File menu in the CBLe main window menu bar.
- Enter command POWER on the command line of any window.

The POWER Command Output window allows the user to enter VSE POWER commands and view the output in the window display area.

If SELCOPYi INI variables System.VSESMLogon=No (i.e. no Security Manager is active) and System.TrustedUser=No, then POWER commands are restricted to PDISPLAY operations only.

The POWER Command Output window is essentially a List window and has the same characteristics as List windows. For example select, sort and filter to display new views of the data are supported.

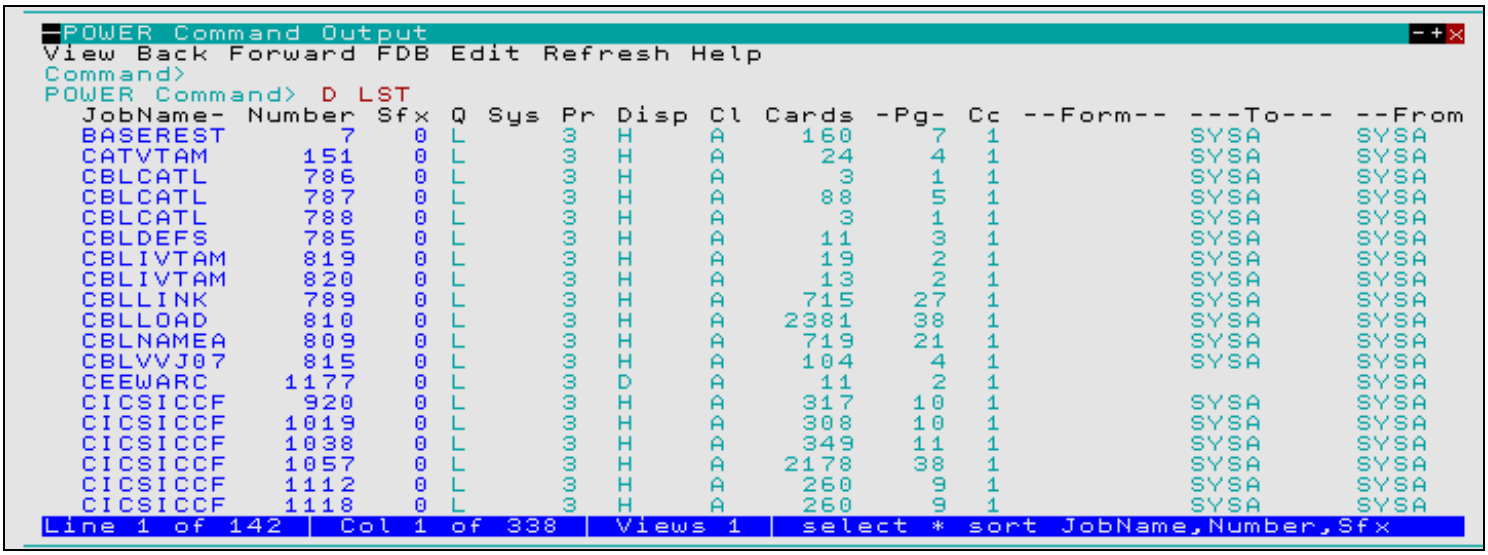


Figure 136. POWER Command Output window for PDISPLAY LST.

Panel Fields

POWER Command>
Specify the VSE POWER command. Note that POWER commands relating to cross partition usage (e.g. PDISPLAY STATUS) are not supported.

Prefix Commands

The following prefix line commands are supported for **PDISPLAY** (RDR, LST or PUN) output only:

Command	Description
<Dflt>	Prefix Line command E.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry. If an entry is password protected, then it may be edited by any user so long as the password is supplied. A pop-up window will prompt the user for the password. Non-password protected entries may only be edited if either of the following are true: • System.TrustedUser=Yes in the SELCOPYi INI file. • System.VSESMLogon=Yes in the SELCOPYi INI file and the TO or FROM attributes match the current user's userid. Note that VSE Basic Security Manager (BSM) alone does not impose access restrictions on the VSE POWER queues.
K	Delete (Kill) the entry without prompting for verification.

Columns Displayed

The data displayed for PDISPLAY ALL/LST/PUN/RDR is:

Name	Type	Description
JobName	Char	JOB NAME
Number	UInt	JOB NUMBER
Sfx	UInt	JOB SUFFIX NUMBER
Q	Char	QUEUE IDENTIFIER (R, L, P)
Sys	Char	SYSTEM ID. (TARGET/PROCESS.)
Pr	Char	PRIORITY
Disp	Char	DISPOSITION (*..IN EXEC.)
Cl	Char	CLASS
Cards	UInt	NUMBER OF RECORDS SPOOLED
Pg	UInt	NUMBER OF PAGES SPOOLED
Cc	UInt	NUMBER OF COPIES
Form	Char	FORMS IDENTIFIER
To	Char	TARGET DESTINATION USER/REMOTE ID
From	Char	ORIGINATING USER/REMOTE ID
Cn	Char	CENTURY OF CREATION DATE
Date	Char	CREATION DATE OF QUEUE ENTRY
Start	Dec	START TIME (0HHMMSSF)
Stop	Dec	STOP TIME (0HHMMSSF)
PXFMRLEN	UInt	RECORD LENGTH
PXFMTYPE	Hex	RECORD TYPE
PXFMVOL	Hex	TAPE BAM VOLUME NUMBER
PXFMUSER	Char	USER INFORMATION
PXFMFLG1	Hex	CONTROL FLAG 1
PXFMRCFM	Hex	RECORD FORMAT
PXFMSTAT	Char	PAPER STATUS BYTE
PXFMLNE#	UInt	NUMBER OF LINES/CARDS SPOOLED
PXFMFLSH	Char	FLASH IDENTIFIER
PXFMCPYG	Hex	COPY GROUPINGS
PXFMFLG2	Hex	CONTROL FLAG 2
PXFMNSEP	UInt	NUMBER OF SEP. PAGES / CARDS
PXFMJBO#	UInt	ORIGINAL JOB NUMBER
PXFMCMPT	Char	COMPACTION TABLE NAME
PXFMNODE	Char	TARGET DESTINATION NODE NAME
PXFMORGN	Char	ORIGINATING NODE NAME
PXFMSUBS	Char	SUBSYSTEM NAME (EXTERNAL WRITER ID)
PXFMDDND	Char	NEXT DUE DATE
PXFMDDNT	Char	NEXT DUE TIME
PXFMQNUM	UInt	QUEUE ENTRY NUMBER
PXFMSECN	Char	QUEUE ENTRY SECURITY ZONE (SECNODE)
PXFMDIST	Char	OUTPUT DISTRIBUTION CODE
PXFMMACN	UInt	.. NON SHARED ACCESS COUNT
PXFMMAC1	UInt	.. SHARED SYSID 1 ACC. CNT.
PXFMMAC2	UInt	.. SHARED SYSID 2 ACC. CNT.
PXFMMAC3	UInt	.. SHARED SYSID 3 ACC. CNT.
PXFMMAC4	UInt	.. SHARED SYSID 4 ACC. CNT.
PXFMMAC5	UInt	.. SHARED SYSID 5 ACC. CNT.
PXFMMAC6	UInt	.. SHARED SYSID 6 ACC. CNT.
PXFMMAC7	UInt	.. SHARED SYSID 7 ACC. CNT.
PXFMMAC8	UInt	.. SHARED SYSID 8 ACC. CNT.
PXFMMAC9	UInt	.. SHARED SYSID 9 ACC. CNT.

The data displayed for other POWER commands is:

Name	Type	Description
Text	Char	Power Display Output

Define Catalog ALIAS (=8.4)

The Define Catalog ALIAS Dialog window may be opened via the following:

- Select option 4. 'Catalog ALIAS' from the Utilities Menu or select option 6. 'Alias' from the Create New Datasets menu panel.
- Select 'Create Catalog ALIAS' from the File menu in the CBLe main window menu bar.
- Enter command AMSA on the command line of any window.
- Enter the List window prefix command "A" against a non-VSAM data set list entry.

The Define Catalog ALIAS dialog provides a simple interface for the user to supply IDCAMS DEFINE ALIAS characteristics for a new alias name. Aliases may be defined for non-VSAM data sets and, if the user has sufficient authority, user catalogs. Compare with the Create Library ALIAS dialog window which creates PDS/PDSE library member aliases.

Select the appropriate menu bar item (see below) to define the new entry.

Fields within this dialog represents the relevant IDCAMS DEFINE ALIAS parameters as appropriate for the entry being defined. Please refer to *"DFSMS Access Method Services for Catalogs"* for further information.

Note: Not implemented for CMS and VSE.

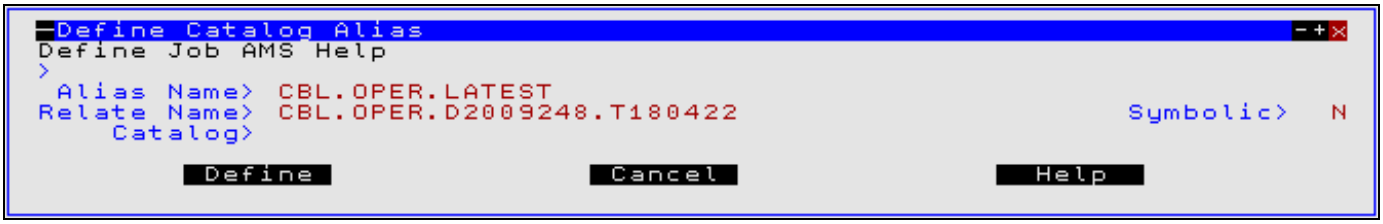


Figure 137. Define Catalog ALIAS Dialog window.

Menu Bar Items

`Define`
: Start the VSAM object definition. (Foreground)

`Job`
: Creates and edits the IDCAMS DEFINE statement including job control ready for submission to batch. (See CBLe command SUBMIT.) (Background)

`AMS`
: Opens a CBLe edit view containing generated AMS command syntax to perform the IDCAMS DEFINE. Execute by placing the cursor on the first line of the command and hitting <F16> The command may be copied to the user's HOME command centre for future reference.

`Help`
: Open the help window for the Define Catalog ALIAS dialog window.

Panel Fields

`Alias Name>`
: Name of the ALIAS object to be defined.

`Relate Name>`
: Name of the object to which the ALIAS will relate.

`Catalog>`
: This field entry specifies the catalog in which the alias is to be defined. If the alias is for a user catalog connector, this field should contain the name of the master catalog. Please refer to section *"Catalog Selection Order for DEFINE"* in *"DFSMS Access Method Services for Catalogs"* for catalog selection when this field is null.

`Symbolic>`
: This field entry may be "Y" or "N" to indicate that the **Relate>** field is a SYMBOLICRELATE containing system symbols (i.e. an Extended ALIAS is to be defined.) See *"DFSMS Managing Catalogs"* for further information.

Create Library ALIAS (=8.5)

The Create Library ALIAS Dialog window may be opened via the following:

- Select option 5. 'Library ALIAS' from the Utilities Menu.
- Select 'Create Library ALIAS' from the File menu in the CBLe main window menu bar.
- Enter command ALIAS -DLG on the command line of any window.
- Enter the List window prefix command "A" against an entry in a Library List.

The Create Library ALIAS dialog provides a simple interface to create a new PDS or PDSE library member alias. Compare with the Define Catalog ALIAS dialog window which creates cataloged aliases for non-VSAM data sets and user catalogs.

Note that aliases for PDSE load-library members are created using the binder to relink the module being aliased. This will result in an update to the module's **TTR**.

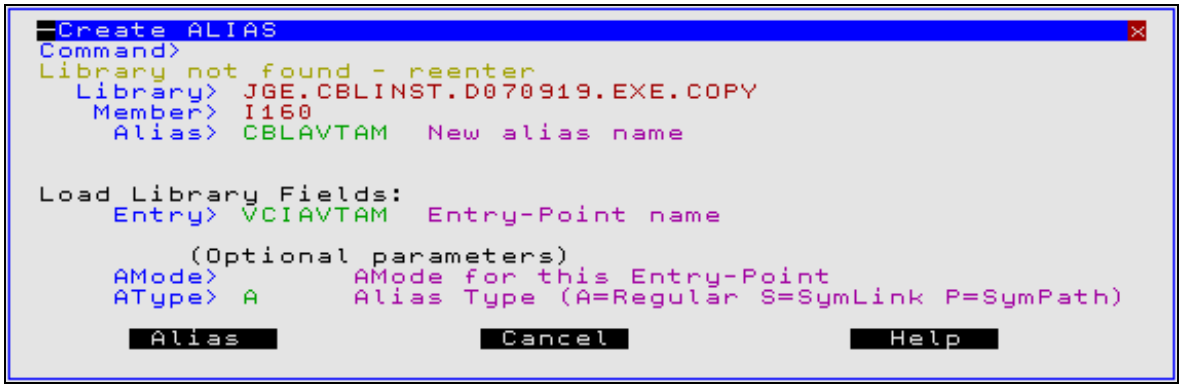


Figure 138. Create Library ALIAS Dialog window.

Panel Fields

Library>
The DSN of the PDS(E) library. (This may be a LOAD Library.)

Member>
The library member name for which an alias will be generated.

Alias>
The new alias name to be generated.

Entry>
For load library aliases only, the symbolic name of the entry-point address to be used.

AMode>
For a load library aliases only, the Addressing Mode for the entry point specified in Entry>. Valid arguments are 24, 31 and ANY.

AType>
For a load library aliasses only, the alias type to be generated. Valid arguments are A=Regular, S=SymLink, P=SymPath.

Execute IEBCOPY (=8.6)

The IEBCOPY Dialog window may be opened via the following:

- Select option 6. 'IEBCOPY' from the Utilities Menu.
- Select 'Execute IEBCOPY' from the File menu in the CBLe main window menu bar.
- Enter command IEBCOPYDIALOG on the command line of any window.
- Enter the List window prefix command "C" against a PDS(E) entry in a Dataset List or Catalog List, or any entry within in a Library List.

The IEBCOPY Dialog provides an intuitive interface to copy PDS(E) libraries or individual members to a new or existing target library.

Select "Copy" to perform the IEBCOPY in the foreground or "JCL" to generate a batch job stream in a CBLe text edit view. Having selected "JCL" the user can issue "JOBCARD" to insert a skeleton JOB statement, before executing SUB to submit the job to batch.

Note: Unless already positioned on one of the window buttons (Copy, JCL, Cancel or Help), <Enter> will first position the cursor on the "Copy" button, <Enter> a second time will select (press) the button to action the command.

Any non-zero return code encountered using the foreground "Copy" option will open the Execute IEBCOPY output listing displaying the SYSPRINT output.
Unless the Output> field value is "YES", if a zero return code is encountered, no output window is opened and a message reporting the number of members copied is returned.

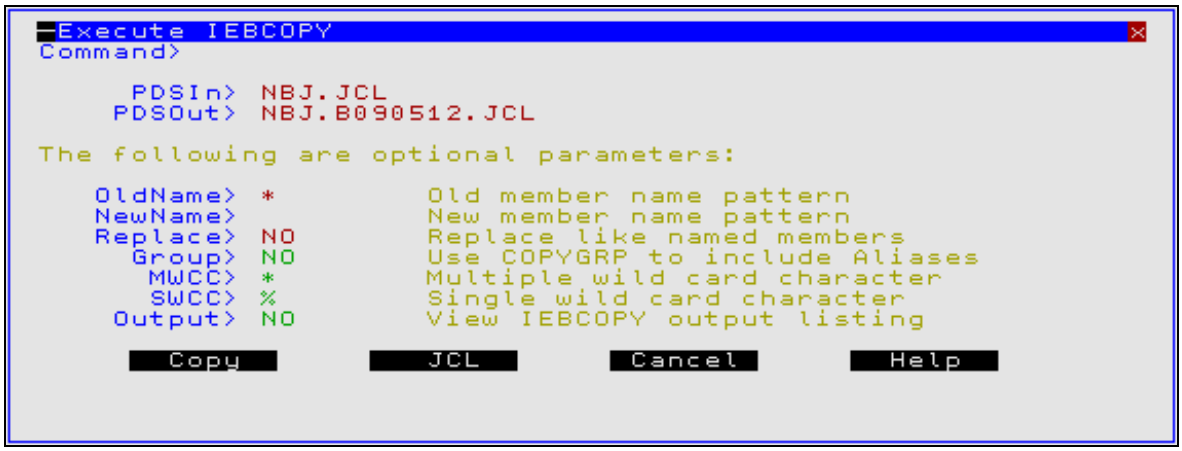


Figure 139. IEBCOPY Dialog window.

Panel Fields

PDSIn>
Specify the DSN of the source PDS(E) library. (This may be a LOAD Library.)

PDSOut>
Specify the DSN of the target PDS(E) library. This may be the same library DSN specified for PDSIn.

Default is the value specified on the last invocation of the IEBCOPY dialog. Otherwise, the default is the PDSIn value.

OldName>
Source library member name mask identifying members to be copied.

Multiple character and single character wild cards, defined by the MWCC and SWCC values (see below) may be used in the member mask.

If no value is specified (i.e. unset), the entire library will be copied. This is different to simply specifying wildcard "*" (asterisk) which copies all members individually. To copy all members of a library, the process is quicker if the value is blank.

If invoked via the "C" prefix command, default is the value of the Library List "Entry" field or is unset if a Dataset or Catalog List. Otherwise, the default is the value specified on the last invocation of the IEBCOPY dialog.

NewName>
Target member name. The member name specified in OldName will be renamed to this new name.

This field must be empty if a wildcard character is used in the OldName member mask. Wild cards are not supported for NewName.
Default is the OldName value.

Replace>
Enter "YES" or "NO" to indicate whether existing members in the target library are to be replaced if the source and target member names match.
Default is the value specified on the last invocation of the IEBCOPY dialog, otherwise "NO".

Group>
Enter "YES" or "NO" to indicate whether any defined ALIAS entries for the selected library members are to be copied also. Note that this also applies to Load Library members.

MWCC>
Specifies the Multiple Wild Card Character which represents zero or more characters in the OldName library member mask.
Default is "*" (asterisk).

SWCC>
Specifies the Single Wild Card Character which represents one character in the OldName library member mask.
Default is "%" (percent).

Output>
Enter "YES" or "NO" to indicate whether the IEBCOPY SYSPRINT output is to be displayed.
Default is "NO".

Favourite Datasets/Commands (=8.7)

Overview

The Favourite Datasets/Commands utility (FAV) enables users to specify a default project hierarchy and also assign file names and command streams to items of a numbered list. This utility was introduced in order to assist migratation from other productivity software that offer similar features.

Users may configure numbered items and later reference the file name or execute the command assigned to an item, simply by entering the item number. This offers users an interface to commonly accessed data sets which be used in addition to commands entered in the user's HOME command centre (CMX) file.

Favourite Datasets/Commands Panel

The Favourite Datasets/Commands panel window (ZZSFAV00) is an interactive panel window (window class WINWIPO0) and may be started via the following:

- Select option 7. 'Favourites' from the Utilities Menu.
- Select 'Favourites' from the Utilities menu in the CBLe main window menu bar.
- Enter command FAV on the command line of any window.

By default, field entries are populated with arguments and options that were entered the last time the Search for Favourite Datasets/Commands panel was used.

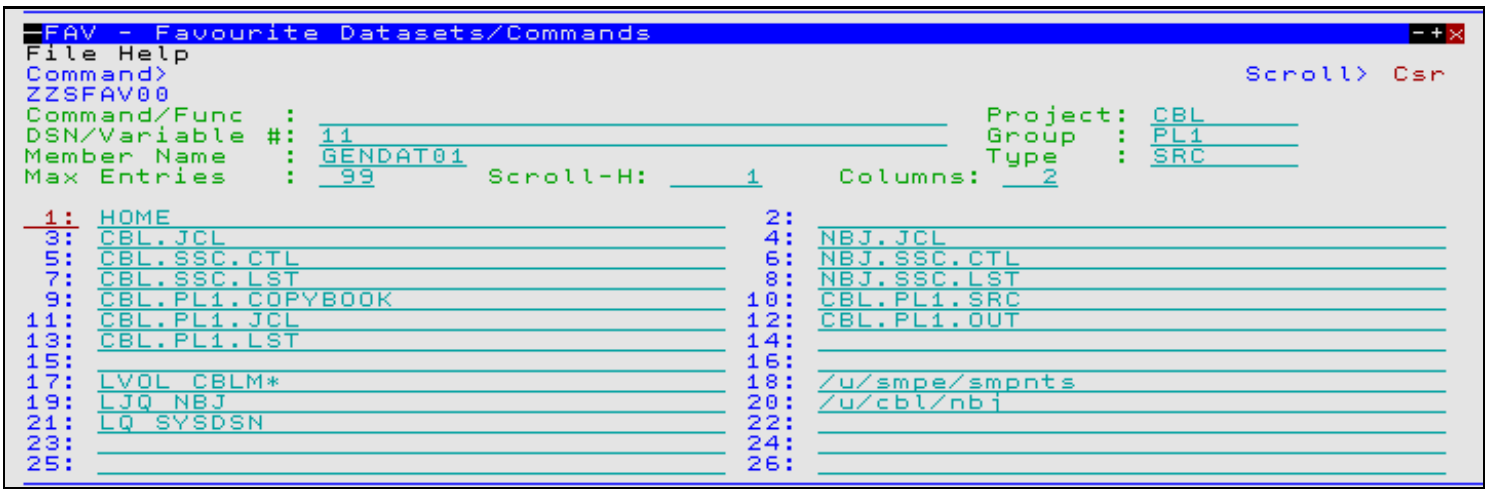


Figure 140. FAV - Favourite Datasets/Commands panel.

When <Enter> is pressed, the a command verb and a fileid parameter is constructed from one or more of the `DSN/Variable #`, `Member Name`, `Project`, `Group` and `Type` fields.

The fielid is determined based primarily on the contents of the `DSN/Variable #` field as follows:

Field Content	Fileid Determination
null	Use the contents of the Project, Group, Type and Member Name fields.
non-numeric	Use the contents of the DSN/Variable # and Member Name field.
numeric	Use the contents of the specified number list item and the Member Name field.

Note that, pressing <Enter> or, if configured, double-clicking the left mouse button on one of the numbered items, is equivalent to entering the item number in the **`DSN/Variable #`** field then pressing <Enter>.

Panel Input Fields

`Command/Func:`
: This input field (ZCOMMAND) allows the user to optionally enter a command to be executed using the contents of the **`DSN/Variable #`** field and, if a PDS/PDSE or GDG DSN, **`Member Name`** field as input to the command. If the contents of the **`DSN/Variable #`** field is numeric, referencing one of the numbered list items, then the contents of that list item is used as input to the command.

If, however, the contents of the list item starts with a recognised command verb, then the command entered in the **Command/Func:** field is ignored and the list item command stream is executed instead.

If left empty, then the command generated for this field on pressing <Enter> is determined as follows:

Command Executed	Condition
none	A command is already included as part of the specified list item number.
LA	The fileid is a single token (qualifier) containing no "." (dot/period) and no leading "/" (slash).
EDIT	The fileid has an MVS PDS/PDSE DSN with a member name or is an HFS file path.
LL	The fileid is an MVS PDS/PDSE DSN with no member name.
LD	The fileid is not an MVS PDS/PDSE DSN.

DSN/Variable #:

This input field (ZFAVNUM) optionally specifies a complete fileid, the DSN of a PDS/PDSE library or a list item number that references a fileid or command stream. If left empty, then a fileid is generated from the **Project**, **Group**, **Type** and **Member Name** fields.

Entering an invalid list item number in this field will return the following errors:

```
ZZSP025E Parameter n is invalid in command SaoGet n.
ZZSE043E RC=20 from: SAOSTRING
```

Member Name:

This input field (ZMEMBER) specifies a library member name to be included as part of the fileid.

For MVS systems only, where this field is not empty, the use of its contents in the resultant fileid is based on whether a member name has already been specified via the other fields used to resolve the fileid. i.e. If no member name is already identified within the constructed fileid, then the contents of the Member Name field are enclosed in "( )" (parentheses) and appended to the fileid.

For VSE and CMS, this member name is used only if the **DSN/Variable #** field is null, in which case the fileid is built from the Project, Group, Type and Member Name fields.

Project:
Group:
Type:

Input fields Project, Group and Type (ZPROJECT, ZGROUP, ZTYPE respectively) identify the default fileid tokens (qualifiers) to be used if the **DSN/Variable #** field is null.

For MVS, the Project, Group and Type fields represent the first three qualifiers of a DSN.

For CMS, the Project and Type fields represent the FileMode and FileType tokens respectively. The Group field is ignored.

For VSE, the Project, Group and Type fields represent a LIBR library name, sub-library name and member type respectively.

Max Entries:

This input field (ZMAXENTRIES) specifies the maximum number of list items to be displayed in the current Favourite Datasets/Commands panel.

This number of list items may be increased or decreased at any time while the panel is open, so adding list items to or removing list from the panel's display. A command stream or fileid assigned to a list entry that has been removed, is not lost and will be redisplayed if the Max Entries value is increased to include this list item. Since panel field values are saved as SELCOPYi User INI file variables, list item values can be redisplayed across SELCOPYi sessions.

Keeping the maximum number of list items low, reduces the amount of storage required to display the Favourite Datasets/Commands panel and also makes the list more managable.

Default number of entries is 99, the maximum value of this field is 999.

Scroll-H:

Input field Scroll-H (ZSCROLLH) is used for the horizontal scrolling of text in list items. It identifies the first text position currently in view in all list items. This value may be updated by the user, so scrolling the list item entry fields so that this text position is the first in view.

Horizontal scrolling of list item text is also achieved using <PF10> (left) and <PF11> (right) from anywhere within the panel display.

Default value is 1. (i.e. the start of the list item text.)

Columns:

Input field Columns (ZDISPLAYCOLS) specifies the number of list item columns to be displayed on a single line of the current Favourite Datasets/Commands panel.

Increasing this value will increase the number of list items visible in the panel but will reduce the length of text displayed in each list field.

Default value is 2 columns per line.

n:

A number of list item (FAV.n) input field entries in which to store commonly accessed fileids and CLI command streams.

The number of list item fields displayed in the panel is defined by the Max Entries input field up to a maximum of 999.

The display of list item entries may be scrolled to a specific list item by overtyping the first item number in the current display (highlighted with red underscore) with the required list item number.

The entire contents of an individual list item may be expanded using <PF2>, allowing edit of the text data. When the expanded data view is closed (using <PF3>) the list item displays the updated text.

From anywhere within the panel display, <PF10> and <PF11> will scroll the contents of **all** list item fields left and right respectively, whereas <PF7> and <PF8> will scroll the list items up and down respectively.

System Information Menu (=8.8)

The System Information Menu panel (ZZSGSYSM) is an interactive panel window opened on selection of option 8. in the SELCOPY/i Utilities Menu.

SELCOPY/i supports display of the status information about the environment in which it is running. With the exception of operating system information windows, system windows are usually only required for diagnostic purposes only.

Option 'About' provides information about the running version of SELCOPY/i including product release and component build levels. It also identifies the latest PTF level.

Options

1 Operating System	SYSI - Operating System Details
2 LPA	SYSLPA - List LPA Modules
3 Link List	SYSLL - List Link-List libraries
4 APF Libraries	SYSAPF - List APF authorised libraries
5 Tasks	SYST - Task List
6 Programs	SYSP - List loaded programs
7 SVC Status	SVC - Display status of the SELCOPY/i SVC
8 About	ABOUT - Display SELCOPY/i Release and Service Level

Operating System Window (=8.8.1)

The Operating System window may be opened via the following:

- Select option 1. 'Operating System' from the System Information Menu.
- Select 'Operating System' from the Utilities/System menu in the CBLe main window menu bar.
- Enter command SYSI on the command line of any window.

Note: Not implemented for VSE and CMS.

Access to this operating system information and also to LPA, LinkList, APFList, Tasks, Storage and Programs information windows, may be restricted if RACF (or equivalent) resource checking has been enabled for the SELCOPY/i and the user does not have read access to the named SYSTEM resource.

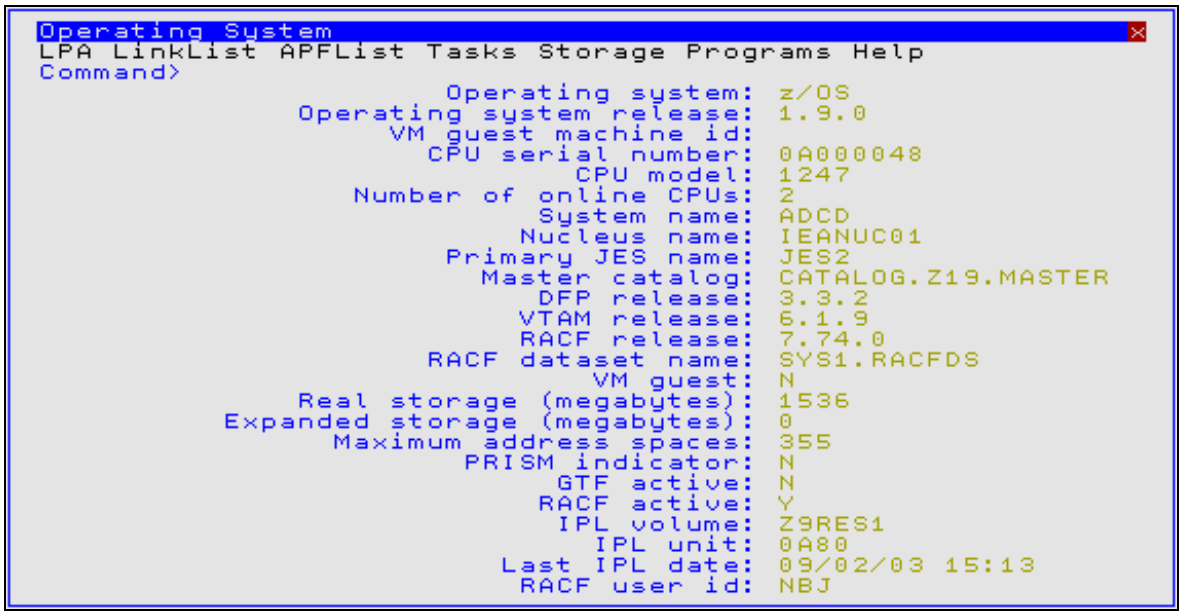


Figure 141. MVS Operating System window.

Menu Bar Items

LPA	Open LPA Modules Window
LinkList	Open Link List Window
APFList	Open APF List Window
Tasks	Open Task List Window

Storage	Open Allocated Storage Windows
Programs	Open Loaded Programs Window

LPA Modules Window (=8.8.2)

The LPA (Link Pack Area) Modules window may be opened via the following:

- Select option 2. 'LPA' from the System Information Menu.
- Select 'LPA' from the Operating System window menu.
- Enter command SYSLPA on the command line of any window.

The LPA Modules window is a List Window and supports the standard List window features. i.e. Field Descriptor Block, Edit and Selecting, Sorting and Filtering.

Note: Not valid for CMS and VSE.

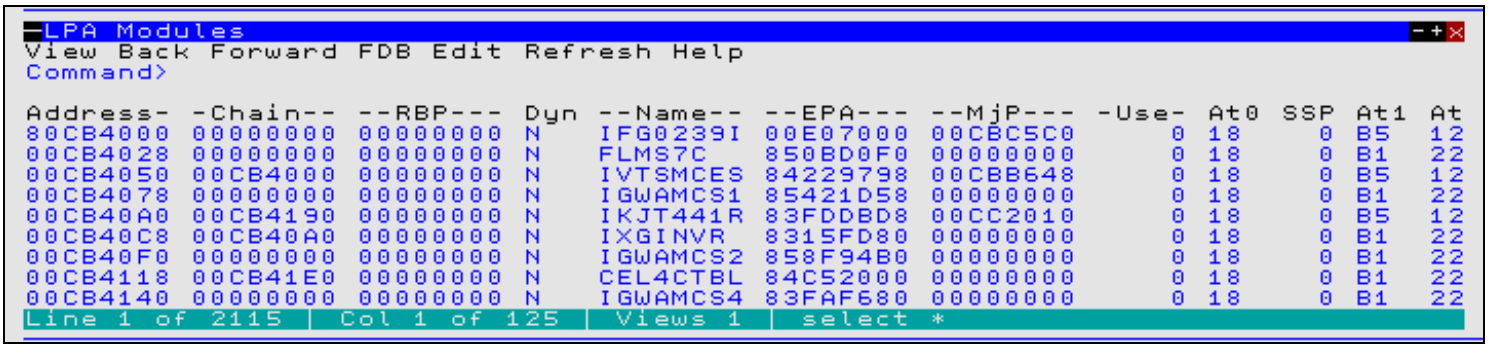


Figure 142. LPA Modules window.

Columns Displayed

Name	Type	Description
Address	Hex	LPDE or CDE element address
Chain	Hex	LPDE or CDE chain pointer
RBP	Hex	RB pointer
Dyn	BitFlag	Dynamic LPA
Name	Char	Module name
EPA	Hex	Entry point address
MjP	Hex	Major name pointer
Use	UInt	Use count
At0	Hex	Attributes 0
SSP	UInt	Storage subpool
At1	Hex	Attributes 1
At2	Hex	Attributes 2
At3	Hex	Attributes 3
At4	Hex	Attributes 4
AliasOf	Char	Aliased name
LoadedAt	Hex	Load point address
LenHex	Hex	Load module length
LenDec	UInt	Load module length

Link List Window (=8.8.3)

The Link List window may be opened via the following:

- Select option 3. 'Link List' from the System Information Menu.
- Select 'LinkList' from the Operating System window menu.
- Enter command SYSLL on the command line of any window.

The Link List window is a List Window and supports the standard List window features. i.e. Field Descriptor Block, Edit and Selecting, Sorting and Filtering.

The contents of the library is displayed in a List Library Members window on hitting <Enter> or, if configured, double-clicking the left mouse button, on that library's entry in the list.

Note: Not valid for CMS and VSE.

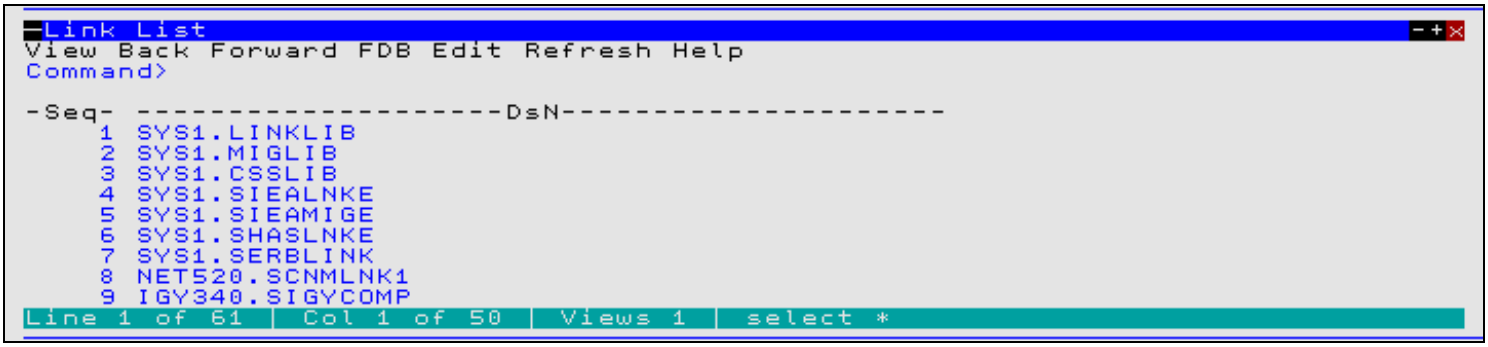


Figure 143. Link List window.

Columns Displayed

Name	Type	Description
Seq	UInt	Link list sequence number
DsN	Char	Link list library name

APF List Window (=8.8.4)

The APF (Authorised Program Facility) List window may be opened via the following:

- Select option 4. 'APF Libraries' from the System Information Menu.
- Select 'APFList' from the Operating System window menu.
- Enter command SYSAPF on the command line of any window.

The APF List window is a List Window and supports the standard List window features. i.e. Field Descriptor Block, Edit and Selecting, Sorting and Filtering.

The contents of the authorised library is displayed in a List Library Members window on hitting <Enter> or, if configured, double-clicking the left mouse button, on that library's entry in the list.

Note: Not valid for CMS and VSE.

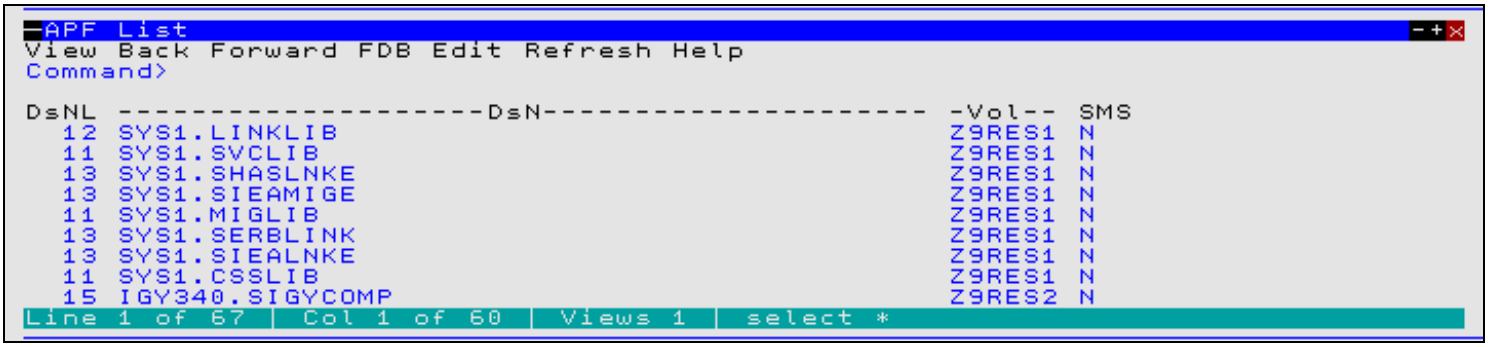


Figure 144. APF List window.

Columns Displayed

Name	Type	Description
DsNL	UInt	APF library name length
DsN	Char	APF library name
Vol	Char	Volume serial
SMS	BitFlag	SMS managed

Task List Window (=8.8.5)

The Task List window may be opened to display the active tasks in the local address space, via the following:

- Select option 5. 'Tasks' from the System Information Menu.
- Select 'Tasks' from the Operating System window menu.
- Enter command SYSTASK on the command line of any window.

The Task List window is a List Window and supports the standard List window features. i.e. Field Descriptor Block, Edit and Selecting, Sorting and Filtering.

Note: Not implemented for CMS and VSE.

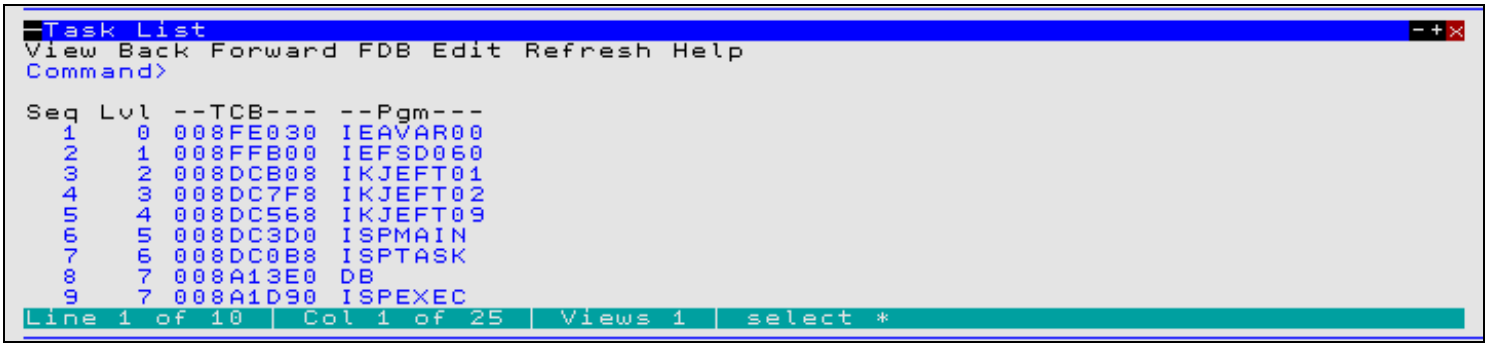


Figure 145. Task List window.

Columns Displayed

Name	Type	Description
Seq	UInt	Task sequence
Lvl	UInt	Task level
TCB	Hex	TCB address
Pgm	Char	Program name

Allocated Storage Windows

Allocated Storage and Unallocated Storage windows may be opened to display lists of areas of allocated, free and unallocated storage in the local address space.

Separate storage window lists are available for Private Area (PVT), Common Service Area (CSA), System Queue Area (SQA) and Local System Queue Area (LSQA) storage. These lists may be opened by selecting the required storage type from a pop-up menu displayed on selecting 'Storage' from the Operating System window menu.

The storage windows belong to the list window class and support standard list window features. i.e. Field Descriptor Block, Edit and Selecting, Sorting and Filtering.

Note: Not implemented for CMS and VSE.

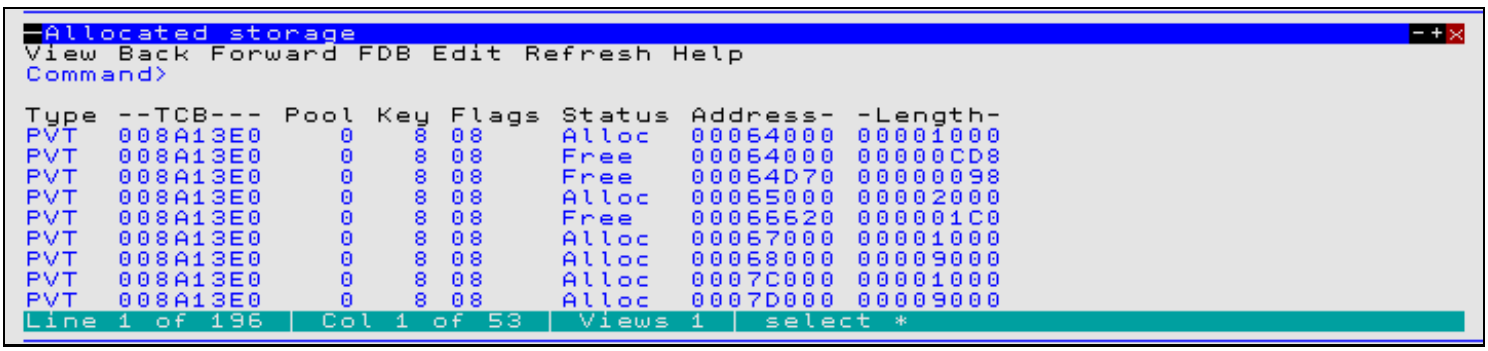


Figure 146. Allocated Private Storage window.

Columns Displayed

Name	Type	Description
Type	Char	Storage type
TCB	Hex	Owning TCB
Pool	UInt	Sub pool
Key	UInt	Sub pool storage key
Flags	Hex	Sub pool flags
Status	Char	Storage status
Address	Hex	Storage address
Length	Hex	Storage length

Loaded Programs Window (=8.8.6)

The Loaded Programs window may be opened to display programs that have been dynamically loaded into the local address space, via the following:

- Select option 6. 'Programs' from the System Information Menu.
- Select 'Programs' from the Operating System window menu.
- Enter command SYSPGM on the command line of any window.

The Loaded Programs window is a List Window and supports standard List window features. i.e. Field Descriptor Block, Edit and Selecting, Sorting and Filtering.

Note: Not implemented for CMS and VSE.

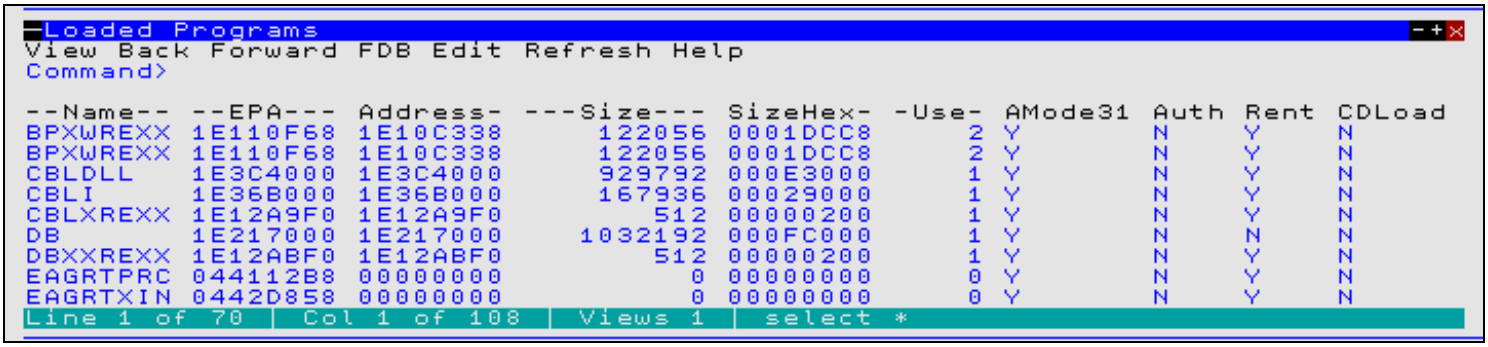


Figure 147. Loaded Programs window.

Columns Displayed

Name	Type	Description
Name	Char	Program name.
EPA	Hex	Entry point address.
Address	Hex	Load address.
Size	UInt	Load module size.
SizeHex	Hex	Load module size (hex).
Use	UInt	Use count.
AMode31	BitFlag	Program is AMODE 31.
Auth	BitFlag	Program is authorised.
Rent	BitFlag	Program is reuseable.
CDLoad	BitFlag	Program loaded with VSE CDLOAD.
Perm	BitFlag	CMS nucleus extension PERM attribute.
Sys	BitFlag	CMS nucleus extension SYSTEM attribute.
Service	BitFlag	CMS nucleus extension SERVICE attribute.
EndCmd	BitFlag	CMS nucleus extension ENDCMD attribute.
ImmCmd	BitFlag	CMS nucleus extension IMMCMD attribute.

SELCOPYi Storage Statistics Window

The Storage Statistics window may be opened via the following:

- Select 'SELCOPYi storage stats' from the Utilities/System menu in the CBLe main window menu bar.
- Enter command SYSSTOR on the command line of any window.

The Storage Statistics window displays storage being used by SELCOPYi at that moment in time. The values in each field will vary as windows are opened and closed.

The storage allocated by SELCOPYi is categorised internally as belonging to the Heap or the Stack.

Heap

Heap storage contains structures such as lists, control blocks, etc.

Each structure is an element within the heap and may persist beyond the life of the function that generated it. Each element exists within a fixed length heap storage block which itself may contain 1 or more elements. When an element is released by SELCOPYi, the area within the storage block occupied by that element, is freed.

If possible, SELCOPYi will utilise these free areas of storage for new elements. However, if an element is generated with a length that exceeds the available free area within existing storage blocks, a new storage block is allocated from main storage. Similarly, if all the elements within a storage block are freed, the block is released back to main storage.

Stack

The stack is a fixed area of storage that contains the dynamic storage areas associated with each function that has been called.

A function acquires dynamic storage for local variables from the stack when it is called and frees this storage when it returns control to its caller.

Thus, the amount of stack storage in use at any time depends on the number of levels of nested function calls and the amount of storage required by each function. This, in turn, depends on which facilities of SELCOPYi are in use.

The **Stack high water mark** represents the maximum amount of stack storage that has been in use in the current SELCOPYi session.

Lists

The List insert work area and List container are areas of storage associated with SELCOPYi listing facilities.

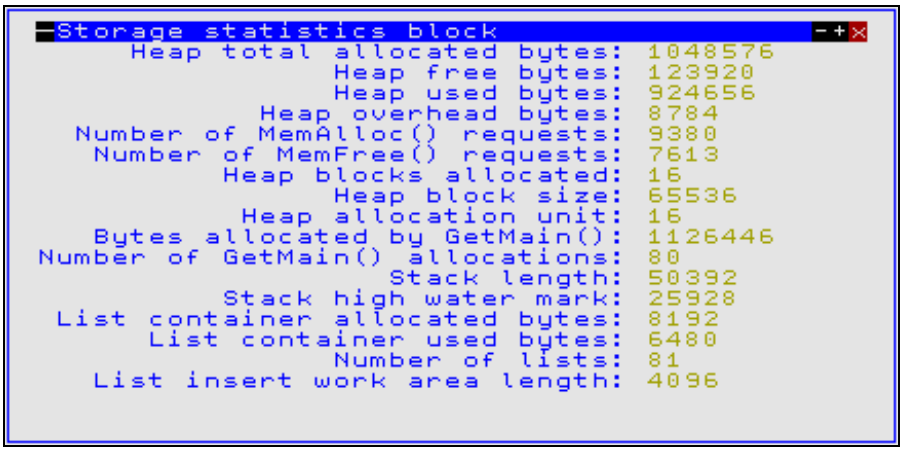


Figure 148. Storage Statistics window.

SELCOPYi Module List Window

The SELCOPYi Module List may be opened to display information on all modules that comprise SELCOPYi, via the following:

- Select 'SELCOPYi module list' from the Utilities/System menu in the CBLe main window menu bar.
- Enter command APE on the command line of any window.

The Module List window is a List Window and supports the standard List window features. i.e. Field Descriptor Block, Edit and Selecting, Sorting and Filtering.

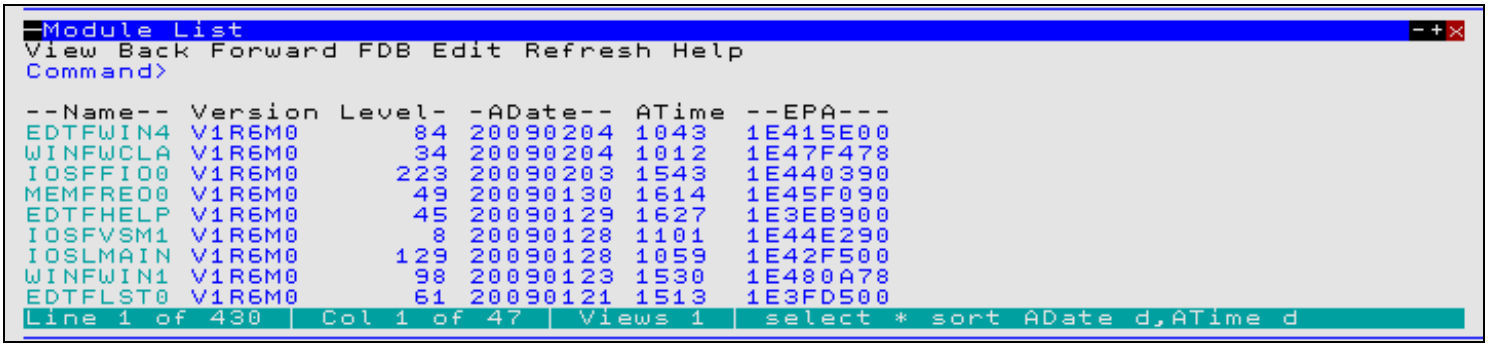


Figure 149. Module List window.

CBLVCAT SVC window (=8.8.7)

The CBLVCAT SVC window may be opened via the following:

- Select option 7. 'SVC Status' from the System Information Menu.
- Select 'SELCOPYi SVC' from the Utilities/System menu in the CBLe main window menu bar.
- Enter command SVC on the command line of any window.

The CBLVCAT SVC window displays information about the CBLVCAT SVC required to perform CBLVCAT LISTVCAT catalog listings.

Note: Not valid for CMS and VSE.

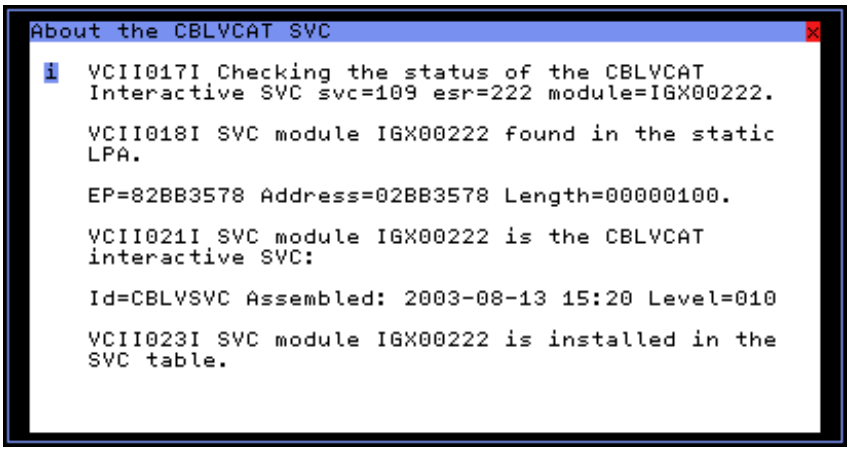


Figure 150. CBLVCAT SVC window.

CBLNAME Window

The CBLVCAT SVC window may be opened via the following:

- Select 'CBLNAME' from the Utilities/System menu in the CBLe main window menu bar.
- Enter command CBLNAME on the command line of any window.

The CBLNAME window is a storage display window containing the CBLNAME module loaded by SELCOPYi.

The CBLNAME storage display window does not display areas of storage outside the loaded CBLNAME module and the data may not be updated by the user.

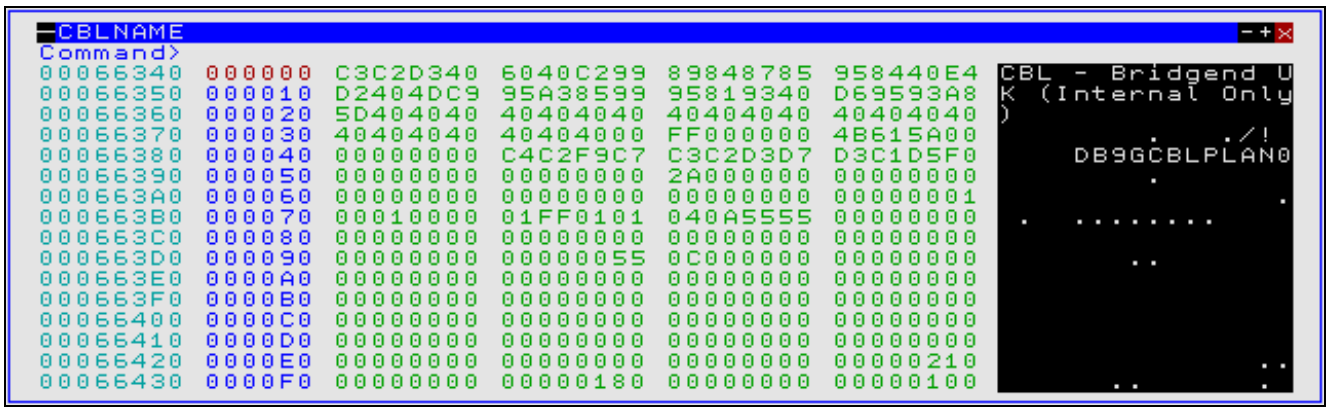


Figure 151. CBLNAME window.

About SELCOPYi (=8.8.8)

The About SELCOPYi window displays information relating to the release and maintenance (PTF) level of the executing SELCOPYi program, the *EXPIRY DATE* of the user's licence, the environment in which it is operating and the number of times the user has started SELCOPYi.

The window is modal and may be opened via the following:

- Select option 8. 'About' from the System Information Menu.
- Select 'About SELCOPYi' from the Help menu in the CBLe main window menu bar.
- Enter primary command **ABOUT** on the command line of any window.

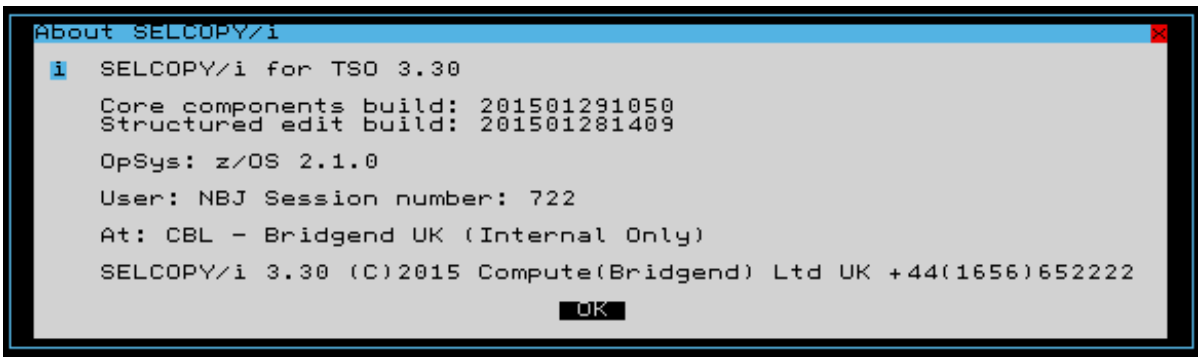


Figure 152. About SELCOPYi window.

File Search (=8.9)

The File Search window may be opened via the following:

- Select option 9. 'Search' from the Utilities Menu.
- Select 'File Search' from the Utilities menu in the CBLe main window menu bar.
- Enter command FS on the command line of any window.

The File Search window displays the lines in a PDS member (MVS), LIBR member (VSE) or CMS file that contain a given string.

For more advanced file search features, use the File Search/Update/Copy utility.

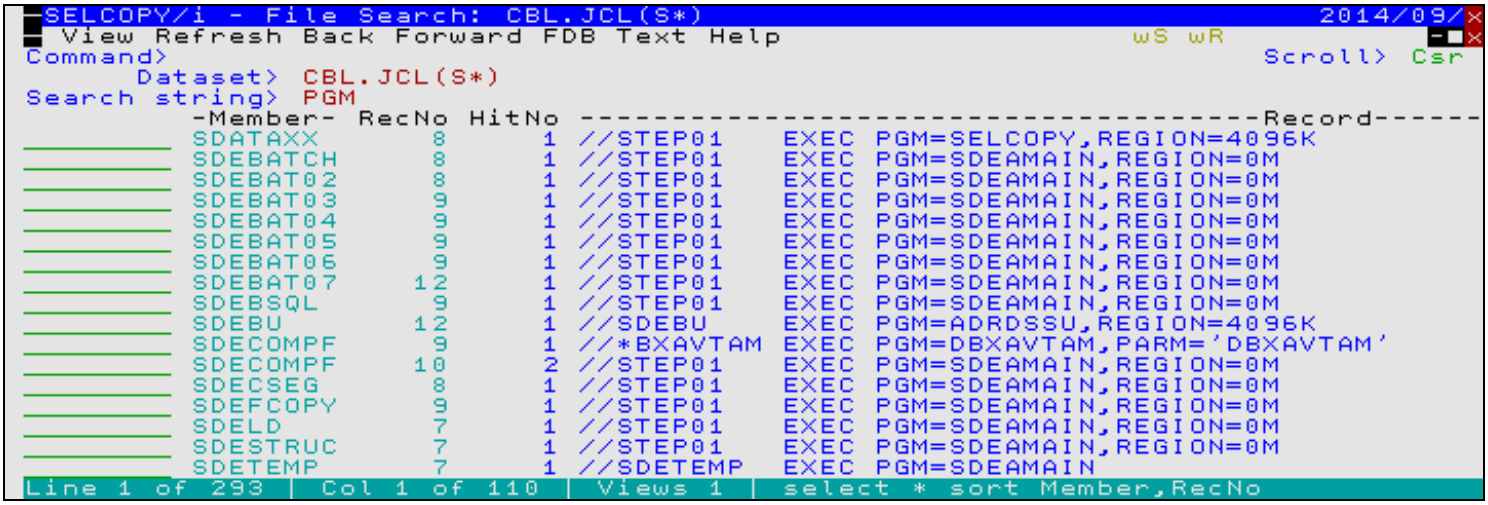


Figure 153. File Search window.

Panel Fields

`Dataset>`

◊ For MVS, the Dataset parameter is the DSN of a PDS(E) library to be searched which may optionally include a member name mask to identify a subset of members to be searched.

A member name mask supports the following wild cards:

`*` A single asterisk represents an entire member name or zero or more characters within a member name mask.

`%` A single percent sign represents exactly one character within a member name mask. Up to 8 percent signs can be specified in each member name mask.

If specified, the member name mask must immediately follow the PDS(E) DSN and be enclosed in "( )" (parentheses). A member name mask that is less than 8 characters in length and does not contain an "*" (asterisk) wild card will have a trailing "*" wild card automatically appended. e.g. To search all members of "CBP.PGMLIB" whose names start "CBLA":

```
CBL.PGMLIB(CBLA)
```

◊ For VSE, the Dataset parameter is the name of the LIBR library and sub-library to be searched. The sub-library name, member name and member type may include the "*" wild card to represent zero or more characters. e.g. To search all members of "OEM2.CBL" :

```
OEM2.CBL.*.*
```

◊ For CMS, the Dataset parameter is a CMS fileid mask in standard CMS format denoting the files to be searched. The file name, file type and file mode may each include the "*" wild card to represent zero or more characters. e.g. To search all "EXEC" file types with file name beginning "SS" on all accessed mini-disks.

```
SS* EXEC *
```

`Search string>`

The character search string.

The search string is **not** case sensitive and must be enclosed in single or double quotes if it includes blank characteres.

Prefix Line Commands

For **MVS** systems, the following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command E.
A	Open the Create Alias dialog window.
B	Open the CBLe text editor to to perform SDATA BROWSE on the entry.
C	Copy the entry.
CF	Open the Compare Files Panel for this entry, using the entry name as the New File field entry.
CL	Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
EX	Execute the entry. (Invokes the TSO command, EXECUTE, using the entry name as input.
F	Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry.
FS	Open the File Search window for the entry.
IC	Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.
J	Submit the entry to batch. Executes the CBLe CLI SUBMIT command using the entry name as input. (A CBLe frame window must be active for this operation to suceed.)
K	Delete (Kill) the entry without prompting for verification.
R	Rename the entry.
SD	Open the SDE BROWSE/EDIT Dialog Window to browse or edit the entry's data within a Structured Data Environment window view.
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBLe text editor to View (edit read/only) this entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

For **VSE** systems, the following prefix line commands are available:

Command	Description
<Dflt>	Prefix line command E.
D	Delete the entry. User will be prompted to verify the deletion.
E	Open the CBLe text editor to edit this entry.
FS	Open the File Search window to search the contents of this entry. Not supported for VSE LIBR library entries.
J	Submit the entry to batch. Executes the CBLe CLI SUBMIT command using the entry name as input. (A CBLe frame window must be active for this operation to suceed.)
K	Delete (Kill) the entry without prompting for verification.
L	LOCK the member.
R	Rename the entry.
U	UNLOCK the member. A member may only be unlocked by the user that locked it.
V	Open the CBLe text editor to View (edit read/only) this entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Columns Displayed

Name	Type	Description
Member	ALPair	Member
RecNo	Int	Record number
HitNo	Int	Hit number
Record	ALPair	File record

Search for Library Members (=8.10)

Overview

The Search for Library Members utility (LLX) provides a method of locating by name, one or more members within a number of PDS/PDSE libraries.

The utility calls the LL (ListLibrary) command repeatedly for each library with a DSN matching the library DSN mask(s) or referenced by a DDname or DDname concatenation of libraries. This will identify the existance of any members that match the specified member name mask(s) within that library.

The Search for Library Members utility executes in the foreground only. To execute in batch, a user can write a simple SELCOPY routine to read the libraries with parameter DIR and report the library directory entries that match the required member mask. See sample SELCOPY routine members (ZZI*) in sample library SZZSSAM1 and also *"SELCOPY Debug & Development"* for assistance in writing new SELCOPY routines.

The Search for Library Members report output is written to a temporary file and presented to the user in a SELCOPYi (CBLe) text edit view.

Unless "Quiet" option has been selected, the library member name search will pause and a popup window opened prompting the user for a decision to cancel or continue, with or without further prompts, if either of the following conditions are true:

1. No matching member names have been found in the first 10 libraries. The popup window proivides an opportunity to change the number of libraries to search before this popup is displayed again.
2. A single library contains at least 1000 matching member names. The popup window proivides an opportunity to change this matching member names threshold.

Search for Library Members Panel

The Search for Library Members utility panel window (ZZSLLX00) is an interactive panel window (window class WINWIPO0) and may be started via the following:

- Select option 10. 'Find Lib Members(s)' from the Utilities Menu.
- Select 'Search for Library members' from the Utilities menu.
- Execute the command LLX with no parameters from the command line of any window.

By default, field entries are populated with arguments and options that were entered the last time the Search for Library Members Utility panel was used.

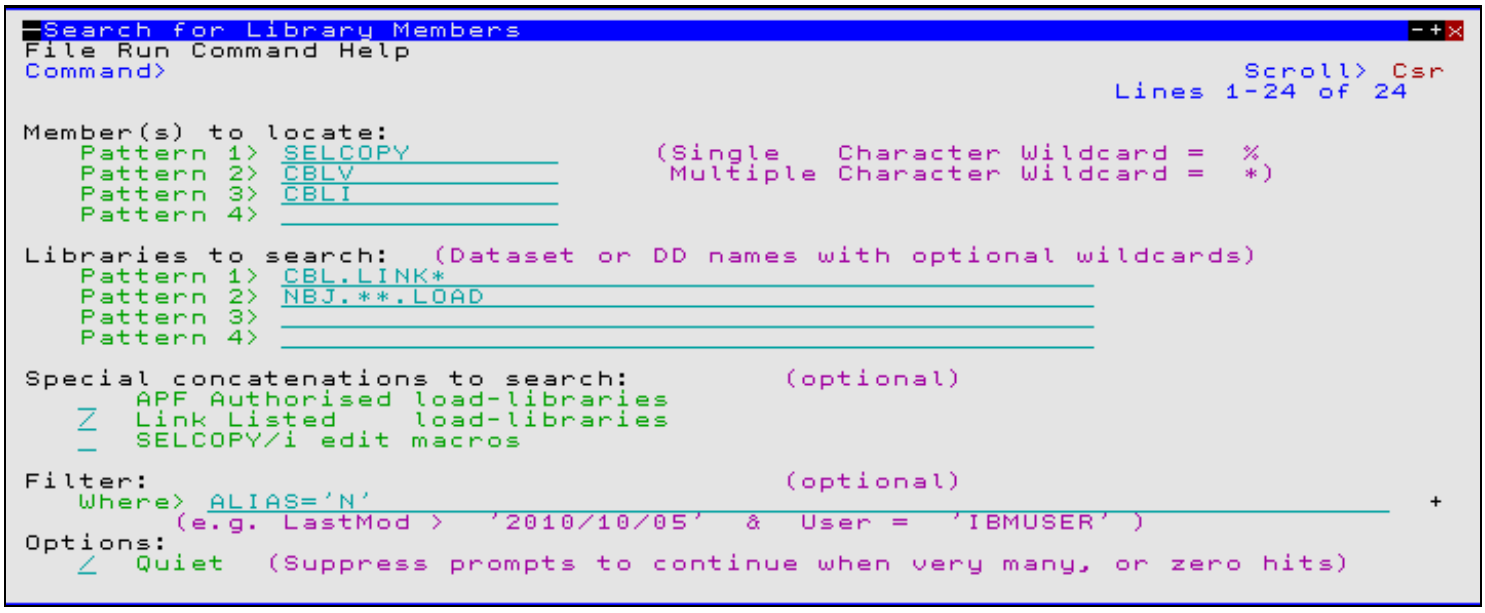


Figure 154. Search for Library Members Panel.

Having typed entries in the required panel fields, simply pressing the <Enter> key or, if configured, double-clicking the left mouse button will will action the library member name search in the foreground.

Menu Bar Items

Run
Run the library member name search in the foreground.

Command
Generate the LLX command line syntax for field entries specified by the user and display it in a temporary CMX file text edit view. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

Panel Input Fields

Members(s) to locate:
Pattern 1/2/3/4>
These input fields (MEMBER1, MEMBER2, MEMBER3 and MEMBER4) allow the user to provide up to 4 alternative member name masks to identify the member names to be located.

A member name mask supports the following wild cards:

- `*` A single asterisk represents an entire member name or zero or more characters within a member name mask.
- `%` A single percent sign represents exactly one character within a member name mask. Up to 8 percent signs can be specified in each member name mask.

If no member name masks are specified, then all libraries selected will be searched for all members. i.e. All members will be reported for each library.

Libraries to search:
Pattern 1/2/3/4>
These input fields (LIBRARY1, LIBRARY2, LIBRARY3 and LIBRARY4) allow the user to provide up to 4 alternative library DSN masks, library DDnames and/or library concatenation DDnames which identiy the PDS/PDSE libraries in which to search.

Note that **all** libraries referenced within a DDname data set concatenation will be searched.

A library DSN mask supports the following wild cards:

- `*` A single asterisk indicates that either a qualifier or one or more characters within a qualifier can occupy that position. An asterisk can precede or follow a set of characters.
- `**` A double asterisk indicates that zero or more qualifiers can occupy that position. A double asterisk cannot precede or follow any characters; it must be preceded or followed by either a dot or a blank.
- `%` A single percent sign indicates that exactly one character can occupy that position. (Up to 8 percent signs can be specified in each qualifier.)

If no library DSN masks or library DDnames are specified, then at least one of the special library concatenation fields (APF Authorised, Link Listed or SELCOPYi edit macros) must be selected.

APF Authorised load-libraries
This option field (APFCHK) indicates that all APF authorised load libraries are to be searched.

This list of libraries may be displayed using the APF List Window (command SYSAPF.)

Link Listed load-libraries
This option field (LNKCHK) indicates that all load libraries in the active Link List concatenation are to be searched.

This list of libraries may be displayed using the Link List Window (command SYSLL.)

SELCOPYi edit macros
This option field (MACCHK) indicates that all libraries in the user's current CBLe text editor macro path are to be searched.

This list of libraries may be displayed using the CBLe text edit command, QUERY MACROPATH.)

Where>
This input field (WHERE) specifies additional member name filter criteria. Members are reported as being found only if the information in the PDS/PDSE directory entry for that member also satisfies this additional criteria.

The syntax of a WHERE filter is described by the list window WHERE Clause which supports list field names as described by the MVS load library and non-load library lists' field descriptor block (FDB). See *"List Library Members"* for details of these field names, descriptions and their data types.

Beware that maintenance of a non-load library member's directory information is **not** enforced by the system. Therefore, its existance depends on the last application to write data to that member. Missing directory fields have default values: 0 if numeric (e.g. VV, MM); ' ' (blank) if character (e.g. User) and null if TimeDec (e.g. Created, LastMod.)

Quiet
This option field (QUIET) indicates that the user will not be prompted for a decision to continue the search when library or member name thresholds are encountered.

Search for Library Members Output

The output generated by the Search for Library Members utility is a temporary CMX command file which gets displayed automatically in a SELCOPYi (CBLe) text edit window view. It identifies each of the library names searched, the number of matching member names in the library, followed by a command to edit each member located in that library.

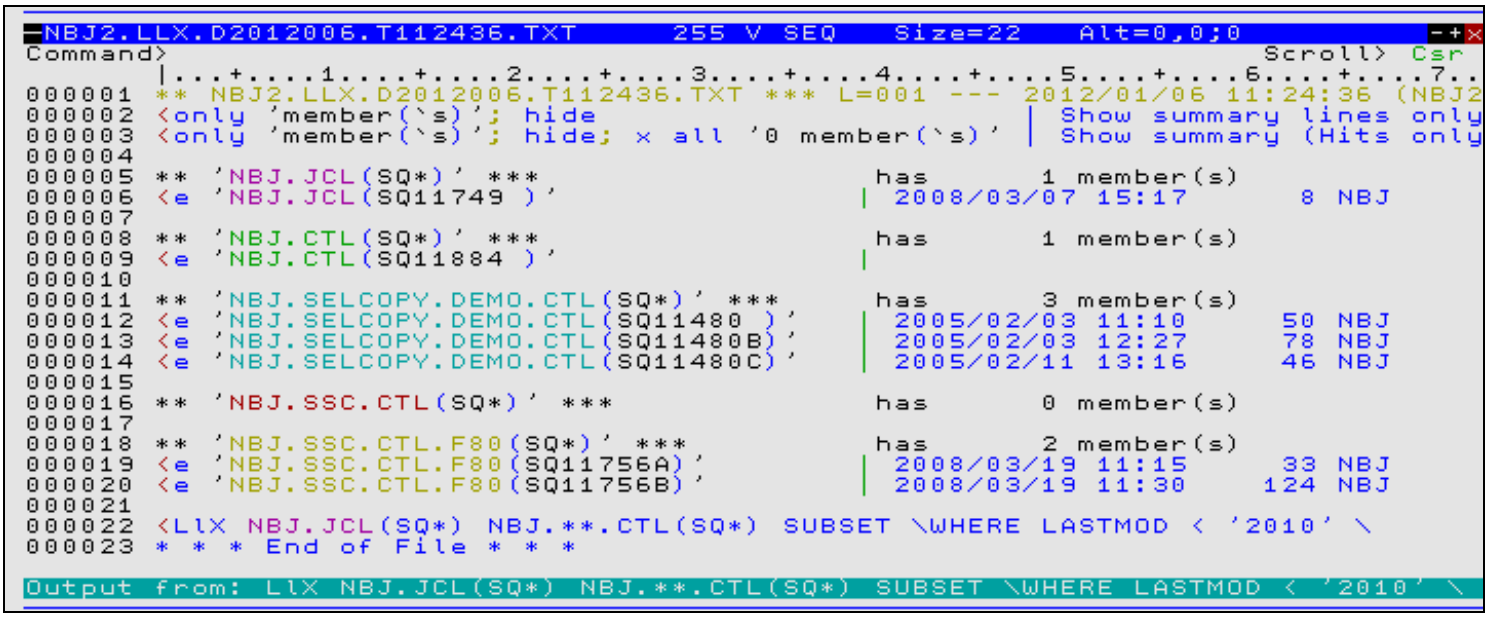


Figure 155. Search for Library Members Output.

The ONLY text edit macro invocations in lines 2 and 3 may be executed to filter edited lines to display all summary lines (library DSN and number of matching member names) or only summary lines for which at leasty one member name has been found.

The last line of the file is the LLX command line syntax generated by the Search for Library Members utility for the options provided.

Any command (prefixed by <) in this file may be executed using the ACTION facility simply by positioning the cursor on the required command and pressing the <F16> key.

Calendar Window (=8.13)

The Calendar window may be opened via the following:

- Select option 13. 'Calendar' from the Utilities Menu.
- Select 'Calendar' from the Utilities menu in the CBLe main window menu bar.
- Enter command CALendar on the command line of any window.

When opened, the calendar window shows the current month with today's date highlighted. Each day has the day of the month and the Julian day number displayed in a table.

You can scroll the calendar backwards and forwards by the month or the year or you can enter a specific year or month in the fields at the top of the window.

To scroll the calendar use the following commands:

Command	Default PF key	Description
SCROLL UP	PF7	Display the previous month.
SCROLL DOWN	PF8	Display the next month.
SCROLL LEFT	PF11	Display the current month in the previous year.
SCROLL RIGHT	PF12	Display the current month in the next year.

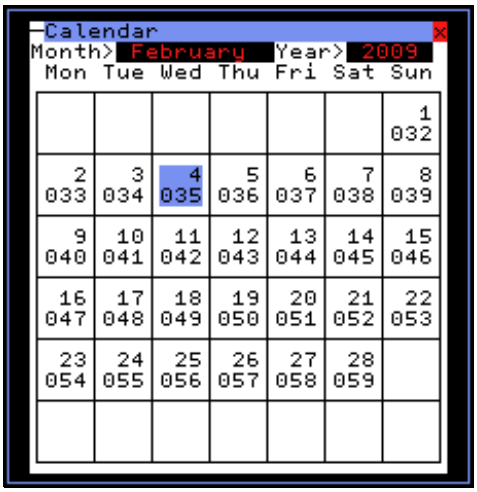


Figure 156. Calendar window.

Calculator Window (=8.14)

The Calculator window may be opened via the following:

- Select option 14. 'Calculator' from the Utilities Menu.
- Select 'Calculator' from the Utilities menu in the CBLe main window menu bar.
- Enter command CALC on the command line of any window.

The calculator window allows you to enter a calculation and displays the result of the calculation.

In fact the calculator is a REXX function interpreter. You enter a valid REXX expression and the calculator evaluates it. You are not restricted to numerical calculations. You can enter any valid REXX expression including for example the conversion functions.

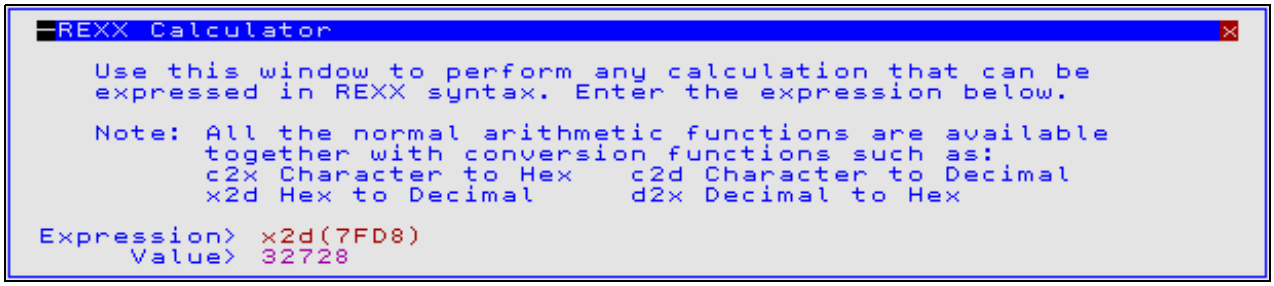


Figure 157. REXX Calculator window.

Create New Datasets Menu (=8.15)

The Create New Datasets Menu panel (ZZSGDEFN) is an interactive panel window opened on selection of option 15. 'Alloc/Define' from the Utilities Menu.

New files may be defined to the system from within the SELCOPY/i environment.

Note that "Copy" automatically invokes an Allocate Non-VSAM or Deine VSAM object panel to create a new output data set if required.

Options

1 Non-VSAM	**ALLOC**	Allocate new Sequential or PDS/PDSE library
2 KSDS	**AMSK**	Define new VSAM KSDS
3 ESDS	**AMSE**	Define new VSAM ESDS
4 RRDS	**AMSR**	Define new VSAM RRDS
5 LDS	**AMSL**	Define new VSAM LDS
6 ALIAS	**AMSA**	Defines new Catalog Alias
7 GDG	**AMSG**	Defines new GDG Base
8 Copy	**FC**	Copy an existing dataset

Allocate NonVSAM (=8.15.1)

The Allocate NonVSAM Dialog window may be opened via the following:

- Select 'Non-VSAM' from the Create New Datasets menu panel.
- Select 'Allocate NonVSAM' from the File menu in the CBLe main window menu bar.
- By any SELCOPY/i utility that requires allocation of an output data set.
- Enter the CBLe command ALLOCATE with no parameters on the command line of any window.
- Perform CBLe or SDE Edit using a new data set name, add some data and SAVE.
- Perform CBLe or SDE Edit of an existing non-VSAM data set, execute SET FILEID to assign a new DSN to the data in storage then SAVE.
  Note that the original data set is unchanged.

The Allocate nonVSAM window allows the user to supply characteristics for a new cataloged non-VSAM data set, then select the Allocate button or the Define menu item to action the allocation.

Fields within these dialogs represent the relevant TSO ALLOCATE or JCL DD statement parameters as appropriate for a new cataloged non-VSAM data set. Please refer to the *"TSO/E Command Reference"* and *"MVS JCL Reference"* for further information.

The **Model>** field allows the user to model the new entry's characteristics on an existing cataloged data set entry. On entering a non-VSAM data set name in the Model field and hitting <Enter>, all other fields are updated automatically to reflect the inherited values.

Note: Not implemented for CMS and VSE.

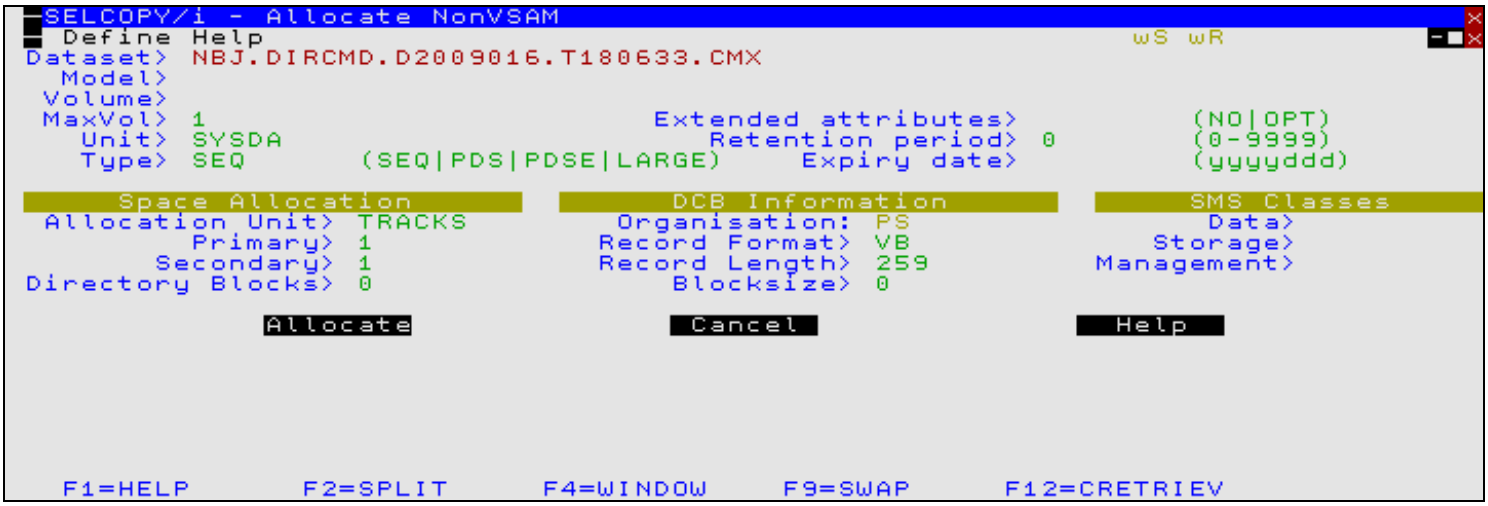


Figure 158. Allocate nonVSAM window.

Menu Bar Items

Define
Drop down menu containing the following items:

Foreground
Applicable to operation under TSO and CMS. Allocate the data set in the foreground (Control is temporarily passed to TSO or CMS).

Foreground+IEBCOPY (Not yet enabled)
As for Foreground but also copy data from the data set specified in the Model field.

Background (Not yet enabled)
Applicable to operation on VSE and MVS. A CBLe view is opened to edit a temporary job containing batch JCL to allocate/define the data set. The job may be submitted to the batch system using the CBLe command SUB.

Background+IEBCOPY (Not yet enabled)
As for Background but also include JCL to populate the data set with data from the data set specified in the Model field.

Help
Open the help window for data set allocation.

Define VSAM KSDS/ESDS/RRDS/LDS (=8.15.2/3/4/5)

The Define VSAM KSDS/ESDS/RRDS/LDS Dialog windows may be opened via the following:

- Select 'KSDS', 'ESDS', 'RRDS' or 'LDS' as appropriate from the Create New Datasets menu panel.
- Select 'Define KSDS', 'Define ESDS', 'Define RRDS' or 'Define LDS' as appropriate from the File menu in the CBLe main window menu bar.
- Enter line command AMSDIALOG with option KSDS, ESDS, RRDS or LDS on the command line of any window. Alternatively, use the synonyms AMSK, AMSE, AMSR or AMSL respectively.
- Perform CBLe or SDE Edit using a new data set name, execute SET DSORG KSDS/ESDS/RRDS/LDS, as required, add some data and SAVE.
- Perform CBLe or SDE Edit of an existing VSAM data set, execute SET FILEID to assign a new DSN to the data in storage then SAVE to save the data to a new VSAM data set of the same type.
  Note that the original VSAM data set is unchanged.

Define VSAM dialog windows allow the user to supply IDCAMS DEFINE characteristics for a new VSAM CLUSTER.

Select the appropriate menu bar item (see below) to define the new entry.

Fields within these dialogs represent the relevant IDCAMS DEFINE CLUSTER parameters as appropriate for the entry being defined. Please refer to *"DFSMS Access Method Services for Catalogs"* for further information.

The **Model>** field allows the user to model the new entry's characteristics on an existing catalog entry. On entering a VSAM data set name in the Model field and hitting <Enter>, all other fields are updated automatically to reflect the inherited values.

Note: Not implemented for CMS and VSE.

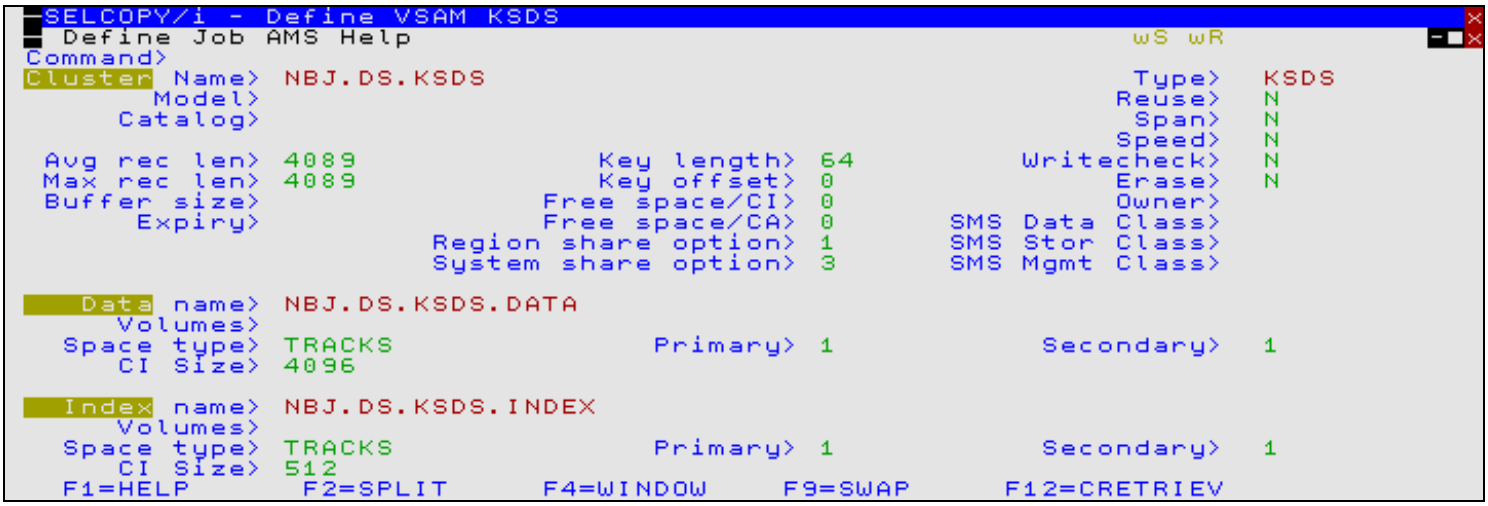


Figure 159. Define VSAM KSDS window.

Menu Bar Items

Define
Start the VSAM object definition. (Foreground)

Job
Creates and edits the IDCAMS DEFINE statement including job control ready for submission to batch (See CBLe command SUBMIT.)

AMS
Opens a CBLe edit view containing generated AMS command syntax to perform the IDCAMS DEFINE. Execute by placing the cursor on the first line of the command and hitting <F16> The command may be copied to the user's HOME command centre for future reference.

Help
Open the help window for VSAM elements definition.

Define GDG Base (=8.15.7)

The **Define GDG Base** panel (ZZSGAMSG) is an interactive panel window (window class WINWIPO0) and may be started via the following:

- Select option 7. 'GDG' from the Create New Datasets menu panel.
- Select 'Define GDG' from the File menu in the CBLe main window menu bar.
- Enter line command AMSDIALOG with option GDG on the command line of any window. Alternatively, use the synonym AMSG.

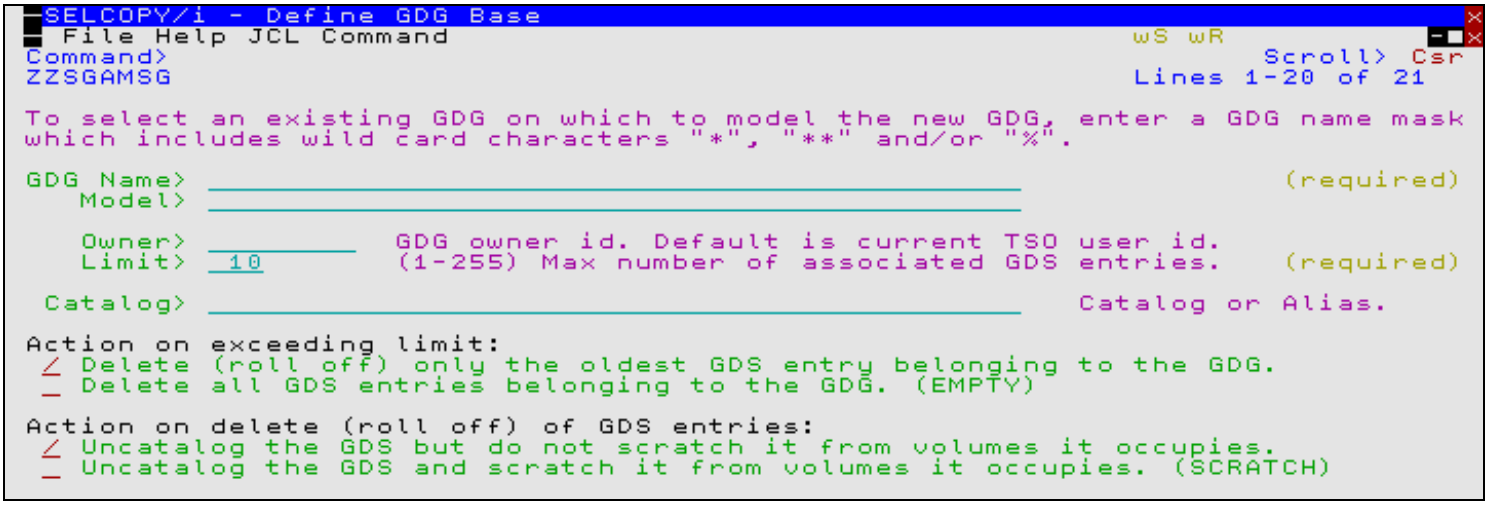


Figure 160. Define GDG Base Panel.

By default, field entries are populated with arguments and options that were entered the last time the panel was used.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button, will will action the GDG define using values entered by the user in the panel input fields.

Alternatively, the user may select an item from the menu bar.

Menu Bar Items

File
The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values entered in the panel so that they may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

JCL
Generate a JCL job stream that executes the **IDCAMS** program with input (SYSIN) containing the DEFINE GENERATIONDATAGROUP syntax generated for the specified panel field values.

The job stream is displayed in a temporary text edit view and may be submitted to batch using the SUBMIT command.

Command
Generate the AMS command line syntax for field entries specified by the user, and display it in a temporary CMX file text edit view. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

Panel Input Fields

GDG Name>
The name of a new GDG Base catalog entry.

Model>
The name of an existing GDG Base catalog entry on which to model attributes of the new GDG.

On entering a model GDG name, press the <Enter> key to update panel input fields to reflect equivalent values assigned to the model.

Wild card characters "%" (percent), representing a single character; "*" (asterisk), representing zero or more characters may be used in the Model GDG entry name to display a selectable list of matching GDG entries.

Owner>
Identfy the owner of the new GDG.

Limit>
Specifies the total number (1-255) of Generation Data Sets (GDS) that can be associated with the new GDG. A non-zero value in this field is mandatory.

Catalog>
The DSN or Alias which identifies the catalog in which the GDG will be defined. If omitted, the default is defined by the IDCAMS catalog search order.

Action on Exceeding Limit:
Specify the action to be taken when a GDS associated with the GDG is to be cataloged which will exceed the defined GDG limit.

`Delete (roll off) only the oldest GDS entry belonging to the GDG.`
Do not empty the GDG of all existing GDS associations, but remove (roll off) only the oldest GDS entry.

`Delete all GDS entries belonging to the GDG. (EMPTY)`
Empty the GDG of all existing GDS associations. i.e. Roll off all associated GDSs.

Action on delete (roll off) of GDS entries:
Specify the action to be taken when a GDS associated with the GDG is removed (rolled off).

`Uncatalog the GDS but do not scratch it from volumes it occupies.`
If the data set is a non-SMS managed data set it is not removed from any of the volumes it occupies. If the data set is an SMS-managed data set it is recataloged as a non-VSAM data set in rolled-off status, and is no longer associated with the GDG base. It is not deleted from any of the SMS-managed volumes it occupies.

`Uncatalog the GDS and scratch it from volumes it occupies. (SCRATCH)`
The GDS is deleted from all volumes it occupies when uncataloged from the GDG base, regardless of whether it is SMS-managed or not.

Generate XML (=8.16)

XML Generation Panels

The XML generation utility panel views (ZZSGXML0) are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select 'XML-Gen' (option 16) from the Utilities menu (=8.16)
- Execute primary command XMLGEN (XML) with no parameters from the command line of any window.

The **XML Generation** panel allows the user to produce an exportable copy of a structured dataset as extensible markup language (XML) text.

The utility reads an input structured dataset and writes an output text dataset consisting of XML tags and tag content. The tag names correspond to the field names of the copybook/structure applied to the input dataset and the tag content to the field values expressed in character format.

Where the input structure maps more than one record-type, the **VIEW** (F22) primary command may be issued to open the Select Record-Types panel and restrict the record-types processed.

The **SELECT** (F5) primary command may also be issued to open the Select Field-Names panel and restrict the fields selected from any given record-type.

The XMLGEN process may be executed in the foreground or as a batch job.

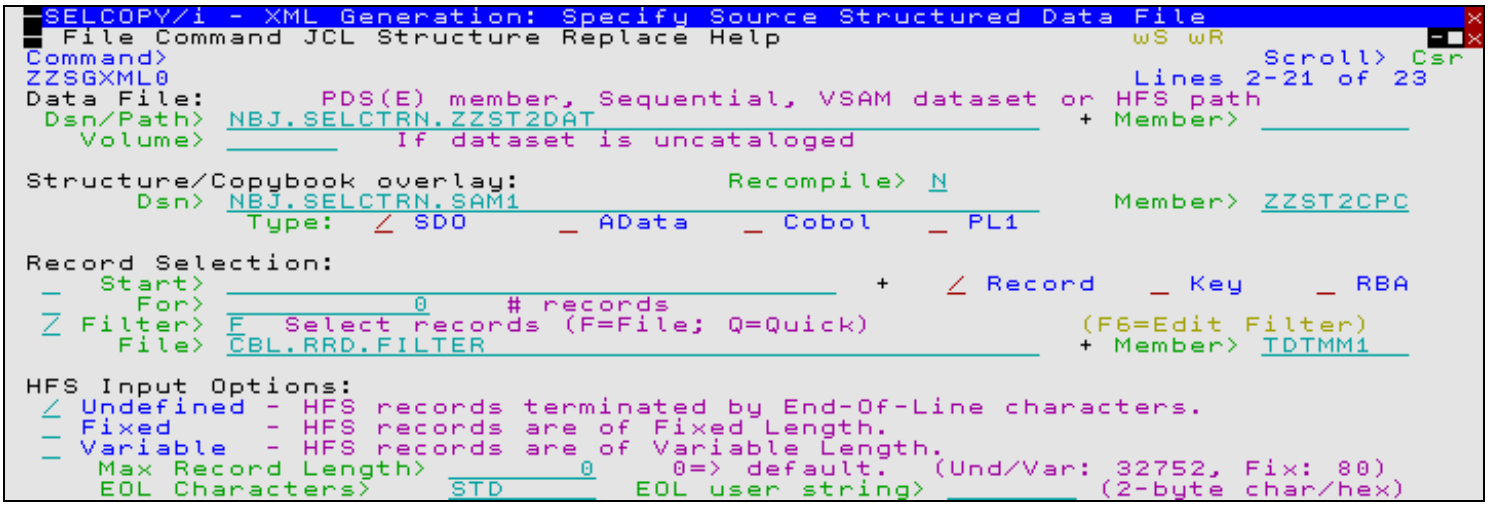


Figure 161. XML Generation Utility Panel.

Menu Bar Items

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values entered in the panel so that they may be redisplayed the next time the panel is opened.

Command
: Generate the XMLGEN command line syntax and display it in a temporary CMX file text edit view. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

The user has the opportunity to edit the command prior to its execution and/or copying it to the home (CMX) command centre for future reference and re-execution.

JCL
: Generate a JCL job stream that executes the **SDEAMAIN** program with input (SDEIN) containing the XMLGEN command generated for the specified panel field values.

The job stream is displayed in a temporary text edit view and may be submitted to batch using the SUBMIT command.

Structure
: Opens the Create Structure (SDO) Menu to generate a SELCOPY/i SDO from a source COBOL or PL1 Copybook or an XREF file.

Replace
Opens the COBOL Compiler Options panel to review and, if necessary, add COBOL REPLACE "From" and "To" pseudo-text values to be used in compiling a COBOL copybook.

Values enterd in this panel apply only to the current user. System wide COBOL REPLACE values may also have been entered in the SELCOPY/i Site INI file. (See the *"SELCOPY Product Suite Customisation Guide"* for details.)

Help
Display help for this panel view.

Panel Fields - Source Structured Data File

Input Data File:
Input fields which together identify a single existing, sequential or VSAM data set, GDG file generation, HFS file or PDS/PDSE library member from which XML is to be generated.

Name>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Volume>
Specifies the name of the input data set volume. This is required only if input is from an uncataloged data set.

Structure/Copybook overlay:
Defines fields which together specify a cataloged structure file (COBOL or PL1 Copybook, ADATA file or a SELCOPY/i SDO) used to format the input records. The structure may be a sequential data set or a PDS/PDSE library member.

These fields must be selected and contain valid entries for record data to be assigned tag names.

Dsn>
Identifies the fully qualified data set name of a sequential data set or PDS/PDSE library containing a COBOL or PL1 Copybook, ADATA file or SDO structure.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Type:
Indicate the type of structure (COBOL, PL1, ADATA or SDO).

Recompile>
If *Structure/Copybook overlay* refers to a COBOL, PL1 or ADATA source file (not an SDO) then a compile step must be performed in order to turn the source copybook into SELCOPYi's own internal structure (SDO) format. A temporary SDO will be created, lasting the duration of the SELCOPYi session, making subsequent reference to the same copybook during the session much faster.

If, however, the copybook (or any of its included components) is modified during the session, then a recompile of the source will normally be required.

For performance reasons **N** (meaning no recompile should occur) is the default.

Specify **Y** to force a copybook recompile each time the process is run. Alternatively type the primary command: **SD DROP *<copybook_name>***

Record Selection:
Fields which together identify criteria by which a subset of records from the file are selected for browse/edit.

Start>
If activated, the **Start>** field identifies the first record in the file at which XML generation will start. Records occurring sequentially before the start record will be excluded. If this field is not activated, records are selected beginning at record 1.

This input field may contain a record number, an RBA number (for ESDS input only), or a key string (for KSDS input only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

Record | Key | RBA
Identifies the type of start value specified in the **Start>** field. Enter "/" in the appropriate, mutually exclusive parameter field.

For>
If activated, the **For>** field specifies the maximum number of records within the file to be processed. If this field is not activated, records are selected beginning at start record and ending at the last record in the file.

Filter>
If activated, the **Filter>** field specifies options to either generate a new record filter or use an existing record filter file. A record filter will perform further subsetting on input file records selected for processing by the Start> and/or For> input fields.

Filter options are as follow:

Q	On executing the **FILTER** command (F6), the Quick Filter dialog panel will be opened in order to generate a temporary filter on the unformatted record data.
F	Use a permanent filter identified by the sequential data set or library member identified in the File> field. On executing the **FILTER** command (F6), the Create File Filter dialog panel will be opened in order to display the contents of an existing filter file or create and save a new filter file.

If option "F" is selected, then specifiaction of a filter fileid is mandatory.

File>
Identifies the fully qualified data set name of a sequential data set or PDS/PDSE library containing a record filter. Quotes are unnecessary but permitted.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the File> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

HFS Input Options:
Options and values that apply to HFS input files only.

Undefined | Fixed | Variable
Identify the format of input HFS records.
Undefined indicates that records are terminated by an End-of-Line (EOL) string.
Fixed indicates that all records are of a fixed length as defined by a specified LRECL.
Variable indicates that all records are of variable length as defined by a length field within the data.

Max Record Length>
Applicable to each of the record formats, this value defines the LRECL (maximum length) of input records. A record longer than this value will be chopped into multiple records.
A 0 (zero) value implies the default which is 32752 for Undefined and Variable record formats and 80 for Fixed record format.

EOL Characters>
Applicable to Undefined record format only, choose from one of the following EOL character combinations:

STD	-	Any standard line-end.
CR	X'0D'	Carriage Return.
LF	X'0A'	Line Feed.
NL	X'15'	New Line.
CRLF	X'0D0A'	Carriage Return + Line Feed.
LFCR	X'0A0D'	Line Feed + Carriage Return.
CRNL	X'0D15'	Carriage Return + New Line.
user	-	A 2-byte user string specified in **EOL user string>**

EOL user string>
Applicable only if **EOL Characters>** is set to **user**, this field specifies the user supplied 2-byte EOL string. It may be specified in character or hexadecimal notation. (e.g. '##', X'FFFF')

Var Length Field>
Applicable to Variable record format only, these fields identify the location of the record length fields within the data.

(Offset)
Offset of the record length field from the start of the record. Default is 0. (i.e. the length field is at the start of the record.)

(Length)
Length (number of bytes) of the record length field. Default is 2.

(Data Origin)
Offset into the record data at which the value in the record length field is to be applied. Default is 0. (i.e. the record length include the length field.)

Panel Fields - Output XML Text File

Output XML Text File:
Input fields which together identify a single output sequential, VSAM or PDS/PDSE library data set, HFS file or PDS/PDSE library member.

DSN/Path>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

If a DSN is specified for a data set that does not already exist, a prompt data set dialog will be opened to allocate the new output file.

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Volume>
Specifies the name of the output data set volume. This is required only if output is to an uncataloged data set.

Append to existing Output
Select this option if the generated records are to be appended to existing records in the output data set.

HFS Output Options:

EOL Characters>
Choose from one of the following End-Of-Line character combinations:

NL	X'15'	New Line.
CR	X'0D'	Carriage Return.
LF	X'0A'	Line Feed.
CRLF	X'0D0A'	Carriage Return + Line Feed.

Translation Options:

Non-printable chars>
Since XMLGEN output is supposed to be in a portable character format, this option is required to specify how non-printable characters are dealt with.

HEX
If a character field contains a non-printable character output the whole field in hex string format. For example a character field length 4 containing X'FFFFFFFF' would have its value represented as

```
X&apos;FFFFFFFF&apos;
```

If a non-printable character is found in a character field and this option is in effect the field XML tag will have the attribute **NONPRINT_CHAR="HEX"**.

ASIS
No special action is taken. All input bytes are copied to the output XML tag value. If a non-printable character is found in a character field and this option is in effect the field XML tag will have the attribute **NONPRINT_CHAR="ASIS"**.

SKIP
The field value is skipped. If a non-printable character is found in a character field and this option is in effect the field XML tag will have the attribute **NONPRINT_CHAR="SKIP"** and no content.

REPLACE
Each non-printable character in a character field is replaced with the specified value. If a non-printable character is found in a character field and this option is in effect the field XML tag will have the attribute **NONPRINT_CHAR="REPLACE"**. The default replace character is the period (full-stop) ".". The replacement character can be specified as:

with>

character
The actual replacement character. If no character is specified then period (.) is assumed. If the character is a lower case letter it will be changed to upper case.

'character' | *"character"*
The actual replacement character in single quotes (apostrophes) or double quotes. If the character is a lower case letter it will be translated to upper case.

c'*character*' | c"*character*"
The actual replacement character in single quotes (apostrophes) or double quotes with a **c** or **C** prefix. No case translation takes place.

x'*hex_value*' | x"*hex_value*"
The replacement character specified as a hexadecimal value.

HEX
This keyword does not represent a replacement character but requests that any substring of non-printable characters found in a character field is replaced with its value in hexadecimal format inside **<HEX> </HEX>** tags. For example a character field length 4 containing X'C1C2FFC3' would have its value represented as

```
AB<HEX>X&apos;FF&apos;</HEX>C
```

In this case the field XML tag wiill have the attribute **NONPRINT_CHAR="REPLACE_HEX"**.

XML special chars>
XML specifies 5 characters as of special syntactical significance. These characters are used to delimit XML constructs and must not appear as themselves in tag values. XML provides an escape sequence (character reference) which can be used to represent these special characters in tag values.

The XML special characters are:

Character	Name	Escape sequence
<	Less than	<
>	Greater than	>
'	Apostrophe	'
"	Double quote	"
&	Ampersand	&

This option provides a way of dealing with any of the XML special characters found in character data fields.

ESCAPE
If a character field contains an XML special character replace it with its XML escape sequence. For example a character field length 4 containing 'A<>B' would have its value represented as:

```
A&lt;&gt;B
```

If an XML special character is found in a character field and this option is in effect the field XML tag will have the attribute **SPECIAL_CHAR="ESCAPE"**.

HEX
If a character field contains an XML special character output the whole field in hex string format. For example a character field length 4 containing 'A<>B' would have its value represented as

```
X&apos;C14C6EC2&apos;
```

If an XML special character is found in a character field and this option is in effect the field XML tag will have the attribute **SPECIAL_CHAR="HEX"**.

CDATA
If a character field contains an XML special character output the whole field asis in an XML character data (CDATA) section. CDATA sections in an XML document represent unparsed character data. For example a character field length 4 containing 'A<>B' would have its value represented as

```
<![CDATA[A<>B]]>
```

If an XML special character is found in a character field and this option is in effect the field XML tag will have the attribute **SPECIAL_CHAR="CDATA"**.

REPLACE
Each XML special character in a character field is replaced with the specified value. If an XML special character is found in a character field and this option is in effect the field XML tag will have the attribute **SPECIAL_CHAR="REPLACE"**. The default replace character is the underscore "_". The replacement character can be specified as:

with>

character
The actual replacement character. If no character follows REPLACE then underscore (_) is assumed. If the character is a lower case letter it will be changed to upper case.

'character' | *"character"*
The actual replacement character in single quotes (apostrophes) or double quotes. If the character is a lower case letter it will be translated to upper case.

c'*character*' | c"*character*"
The actual replacement character in single quotes (apostrophes) or double quotes with a **c** or **C** prefix. No case translation takes place.

x'*hex_value*' | x"*hex_value*"
The replacement character specified as a hexadecimal value.

HEX
This keyword does not represent a replacement character but requests that any substring of XML special characters found in a character field is replaced with its value in hexadecimal format inside **<HEX> </HEX>** tags. For example a character field length 4 containing 'A<>B' would have its value represented as:

```
A<HEX>X&apos;4C6E&apos;</HEX>B
```

In this case the field XML tag wiill have the attribute **SPECIAL_CHAR="REPLACE_HEX"**.

Invalid data values>
Non-character fields in structured data files may have invalid values which cannot be converted to character format. For example, a field defined as containing packed decimal (COBOL COMP-3) data may not contain a valid packed decimal value. This option provides a way of specifying how such fields are represented in the XML output.

HEX
If a non-character field contains an invalid data value output the whole field in hex string format. For example a packed decimal field length 4 containing X'00000000' would have its value represented as

```
X&apos;00000000&apos;
```

If an invalid value is found in a non-character field and this option is in effect the field XML tag will have the attribute **INVALID_DATA="HEX"**.

SKIP
The field value is skipped. If an invalid data value is found in a non-character field and this option is in effect the field XML tag will have the attribute **INVALID_DATA="SKIP"** and no content.

REPLACE
The invalid field data value is replaced with the specified value. If an invalid field data value is found in a non-character field and this option is in effect the field XML tag will have the attribute **INVALID_DATA="REPLACE"**. The default replace character is the asterisk "*". The replacement character can be specified as:

with>
character
The actual replacement character. If no character follows REPLACE then asterisk (*) is assumed. If the character is a lower case letter it will be changed to upper case.

'character' | *"character"*
The actual replacement character in single quotes (apostrophes) or double quotes. If the character is a lower case letter it will be translated to upper case.

c'*character*' | c"*character*"
The actual replacement character in single quotes (apostrophes) or double quotes with a **c** or **C** prefix. No case translation takes place.

x'*hex_value*' | x"*hex_value*"
The replacement character specified as a hexadecimal value.

CCSID Conversion>
Since the purpose of XMLGEN is to produce a portable export version of the data in a z/OS mainframe structured data file, and the output is character data, the coded character set identifiers (CCSIDs) of the input, output and of the XMLGEN internal constants themselves are of significance.

Even if the input and output is coded in an EBCDIC CCSID, these may differ, and both may differ from the CCSID of the XMLGEN internal constants. Since some of the special characters used in XML have different code points in different EBCDIC CCSIDs (for example square brackets) these must be dealt with consistently to produce correct XML output.

XMLGEN uses the z/OS character conversion support supplied by IBM modules CUNLINFO (for obtaining CCSID information) and CUNLCNV (for character conversion from one CCSID to another).

The internal XMLGEN CCSID (that of the constants used to build the XML syntax) is CCSID 285 (EBCDIC, SBCS UNITED KINGDOM).

XMLGEN assumes a default CCSID as follows:

Interactive
When executed interactively XMLGEN uses as default input CCSID that of the user's 3270 terminal.

Batch
When executed in batch XMLGEN uses as default input CCSID the value of the INI file variable **SDE.CCSID**. This variable is set automatically to the user's 3270 terminal CCSID (if not already set) during an interactive session. It can also be set using the structured data SET CCSID command.

Avaliable options are as follow:

NONE
The XML output dataset is produced using the default CCSID and the input dataset character fields are assumed to be in the same CCSID. The internal XMLGEN constants are converted from internal CCSID 285 to the default CCSID.

ASCII
Convert the output to ASCII. This is equivalent to specifying **CONVERT TO 819**. CCSID 819 is ISO 8859-1 ASCII.

UNICODE
Convert the output to UNICODE (UTF-16). This is equivalent to specifying **CONVERT TO 1200**. CCSID 1200 is the IBM bigendian UTF-16 CCSID which is automatically transformaed to the most recent UTF-16 standard.

CONV
from>
from_ccsid
The input character data fields are converted from this CCSID. If not supplied the default input CCSID is used.

to>
to_ccsid
The CCSID of the output XML text dataset. Internal XMLGEN character literals and input character data fields (and HFS line end characters if used) are converted to this CCSID.

Tag Name Case>
This input field value controls whether XML tag names generated from the structure field names are to be upper cased.

DEFAULT
Use the default action as set by the global SDE option, NAMECASE.

TAGUPPER
Tag names are uppercased.

NOTAGUPPER
Tag names are not uppercased. Tags match the field names with no upper case translation.

Miscellaneous Options:

View option>
When XMLGEN is run interactively this option allows the user to request to view the output when the process completes.

BROWSE
Browse the output XML dataset.

EDIT
Edit the output XML dataset using the SELCOPY/i text editor.

NOVIEW
Do not view the output XML dataset. This option is forced when run in batch.

Indentation value>
Nested output XML tags corresponding to the hierarchy of group and elementary data fields in the input structure are indented by a default of one space for each data item level. This option allows the specification of a different indentation value.

Output all field redefinitions:
If the structure defined for the input dataset contains redefined fields this option controls whether the field redefinitions are output.

Output all unnamed (FILLER) fields:
If the structure defined for the input dataset contains unnamed or FILLER fields this option controls whether these fields are output.

Output all fields as elementary:
This option controls whether elements of a group field are output as children of their parent group tag. If activated then group field tags are not included and all elements are output at the top level within the record-type.

Split long XML records:
For each elementary input field XMLGEN builds one output record containing the field start tag, the field value (possibly with embedded HEX tags and special character escape sequences), and the field end tag. Depending on the options chosen and the nature of the input data, relatively long output records may result. If an output

record is longer than the allocated logical record length of the output dataset this option controls how XMLGEN deals with the long output record.

Activated
Split the output record breaking it up into as many logical records as necessary. Records are split at the logical record length irrespective of the record content.

Not activated
Do not split the output record. Rather than truncate the output record XMLGEN terminates with an error message. This is the default.

Suppress output comment block:
This option controls whether an XML style comment block is generated at the top of the output dataset. This contains information about the host operating system, the id of the creator of the output file and the creation date and time, and details of any character conversion performed on the output character data.

Primary Commands

The following primary commands are supported.

FILTER	Open the FILTER dialog for the specified filter type.
BC	Browse the input copybook..
BIF	Browse the input file..
BOF	Browse the output file.
VIEW	Select record-types.
SELECT	Select field-names.

Function Keys

In addition to the standard interactive panel key assignments for scrolling and navigation, the XMLGEN panel supports the following:

F6	FILTER	Open the FILTER dialog for the specified filter type.
F14	EXPAND	Expand an input/output field.
F16	BC	Browse the input copybook..
F19	BIF	Browse the input file.
F20	BOF	Browse the output file.
F22	VIEW	Select record-types.
F23	SELECT	Select field-names.

Select Record-Types Panel

The **Select Record Types** panel is displayed following execution of the **VIEW** command in the XML generation panel views.

This panel contains a list of all record types defined by an input structure/copybook presented to the user as an editable table. Any **USE WHEN** condition, used to determine whether record data fits the record type definition, is also displayed.

Records assigned a record type that is included in this list will be eligible for processing by the calling panel.

Use standard SELCOPY/i table editing techniques to exclude or delete record types from this list and so exclude records assigned these record types from being processed. For example, the following command may be executed to exclude all rows except those where the record type name begins with literal "**ABC-**":

```
WHERE   ViewRT >> 'ABC-'
```

Pressing <F3> to exit the panel, will also save the table of selected keyed record types and return to the calling panel.

Select Field Names Panel

Choose Record-Type

The **Select Field Names** panel is displayed following execution of the **SELECT** (or SEL) command from the XML or CSV generation panel views.

Choose each record type for which specific field columns are to be selected by entering 'S' against the record type in the **Sel** column or by positioning the cursor on the required record type then either pressing the <Enter> key or, if configured, double-click the left mouse button. To deselect the record type key field definition, remove the 'S' against its entry in the 'Sel' column.

For each chosen record type, the Select from Field Names List panel is opened displaying a list of fields comprising that record type. The list of field names should be edited so that only the required key fields are diplayed. The order in which these fields occur in this list will be the order in which they are processed.

On return from the selectable field list, the **Fields Selected** column will be updated to indicate the number of fields included for compare.

Pressing <F3> to exit the panel, will also save the table of selected record types for which selected fields will be processed, and return to the calling panel.

Select Field Names List

The **Select from Field Names List** panel is displayed for each record type selected from the Select Field Names panel.

A list of field names, defined by the selected record type, is presented to the user as an editable table. Standard SELCOPY/i table editing techniques should be used to exclude and re-order the fields so that only fields which are to be processed are displayed in the order in which they are to be processed.

Only included field name entries are processed, This has the benefit that a field may easily be included again later if necessary. For example, the following commands may be executed to filter (include) specific table rows:

```
WHERE (length(strip(SelectFld),'T') > 5)  and  (#3 = 'BN')
```
Exclude all rows except those where the length of the Field Name entry is greater than 5 and the Field Picture Type is "BN". Note that the "Field Pic Type" column is field reference number 3.

```
MORE SelectLev < 3
```
Include previously excluded entries where the field level is 1 or 2. (Entries that are already included will remain included.)

The order in which the field names occur is the order in which the fields will be handled.

Pressing <F3> to exit the panel, will also save the table of selected field names and return to the calling panel.

Generate CSV (=8.17)

CSV Generation Panels

The CSV generation utility panel views (ZZSGCSV0) are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select 'CSV-Gen' (option 17) from the Utilities menu (=8.17)
- Execute primary command CSVGEN (CSV) with no parameters from the command line of any window.

The **CSV Generation** panel allows the user to produce an exportable copy of a structured dataset as comma separated variable (CSV) text.

Where the input structure maps more than one record-type, the user will be prompted to select a single record-type, with records belonging to all other record-types being bypassed.

The **SELECT** (F5) primary command may also be issued to open the Select Field-Names panel and restrict the fields selected from any given record-type.

The CSVGEN process may be executed in the foreground or as a batch job.

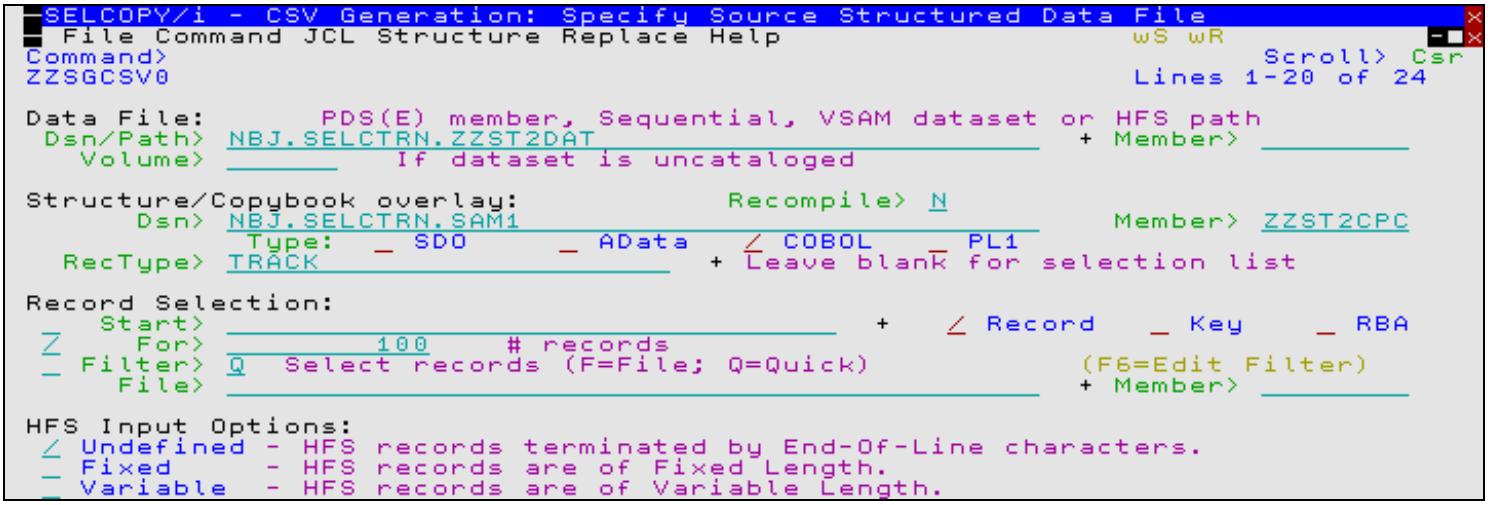


Figure 162. CSV Generation Utility Panel.

Menu Bar Items

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values entered in the panel so that they may be redisplayed the next time the panel is opened.

Command
: Generate the CSVGEN command line syntax and display it in a temporary CMX file text edit view. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

The user has the opportunity to edit the command prior to its execution and/or copying it to the home (CMX) command centre for future reference and re-execution.

JCL
: Generate a JCL job stream that executes the **SDEAMAIN** program with input (SDEIN) containing the CSVGEN command generated for the specified panel field values.

The job stream is displayed in a temporary text edit view and may be submitted to batch using the SUBMIT command.

Structure
: Opens the Create Structure (SDO) Menu to generate a SELCOPY/i SDO from a source COBOL or PL1 Copybook or an XREF file.

Replace
: Opens the COBOL Compiler Options panel to review and, if necessary, add COBOL REPLACE "From" and "To" pseudo-text values to be used in compiling a COBOL copybook.

Values enterd in this panel apply only to the current user. System wide COBOL REPLACE values may also have been entered in the SELCOPY/i Site INI file. (See the *"SELCOPY Product Suite Customisation Guide"* for details.)

Help
Display help for this panel view.

Panel Fields - Source Structured Data File

Input Data File:
Input fields which together identify a single existing, sequential or VSAM data set, GDG file generation, HFS file or PDS/PDSE library member from which CSV is to be generated.

Name>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Volume>
Specifies the name of the input data set volume. This is required only if input is from an uncataloged data set.

Structure/Copybook overlay:
Defines fields which together specify a cataloged structure file (COBOL or PL1 Copybook, ADATA file or a SELCOPY/i SDO) used to format the input records. The structure may be a sequential data set or a PDS/PDSE library member.

These fields must be selected and contain valid entries for record data to be assigned tag names.

Dsn>
Identifies the fully qualified data set name of a sequential data set or PDS/PDSE library containing a COBOL or PL1 Copybook, ADATA file or SDO structure.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Type:
Indicate the type of structure (COBOL, PL1, ADATA or SDO).

Recompile>
If *Structure/Copybook overlay* refers to a COBOL, PL1 or ADATA source file (not an SDO) then a compile step must be performed in order to turn the source copybook into SELCOPYi's own internal structure (SDO) format. A temporary SDO will be created, lasting the duration of the SELCOPYi session, making subsequent reference to the same copybook during the session much faster.

If, however, the copybook (or any of its included components) is modified during the session, then a recompile of the source will normally be required.

For performance reasons **N** (meaning no recompile should occur) is the default.

Specify **Y** to force a copybook recompile each time the process is run. Alternatively type the primary command: **SD DROP *<copybook_name>***

RecType>
Identifies the name of a record-type record mapping defined within the structure. CSV will be generated only for records of this record-type.

The SDE: Select Record-Type panel will be automatically opened to display a selectable list of record-types if the structure contains multiple record types. Otherwise the single record-type will be inserted automatically.

Record Selection:
Fields which together identify criteria by which a subset of records from the file are selected for browse/edit.

Start>
If activated, the **Start>** field identifies the first record in the file at which CSV generation will start. Records occurring sequentially before the start record will be excluded. If this field is not activated, records are selected beginning at record 1.

This input field may contain a record number, an RBA number (for ESDS input only), or a key string (for KSDS input only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

Record | Key | RBA
Identifies the type of start value specified in the **Start>** field. Enter "/" in the appropriate, mutually exclusive parameter field.

For>
If activated, the **For>** field specifies the maximum number of records within the file to be processed. If this field is not activated, records are selected beginning at start record and ending at the last record in the file.

Filter>
If activated, the **Filter>** field specifies options to either generate a new record filter or use an existing record filter file. A record filter will perform further subsetting on input file records selected for processing by the Start> and/or For> input fields.

Filter options are as follow:

Q	On executing the **FILTER** command (F6), the Quick Filter dialog panel will be opened in order to generate a temporary filter on the unformatted record data.
F	Use a permanent filter identified by the sequential data set or library member identified in the File> field. On executing the **FILTER** command (F6), the Create File Filter dialog panel will be opened in order to display the contents of an existing filter file or create and save a new filter file.

If option "F" is selected, then specification of a filter fileid is mandatory.

File>
Identifies the fully qualified data set name of a sequential data set or PDS/PDSE library containing a record filter. Quotes are unnecessary but permitted.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the File> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

HFS Input Options:
Options and values that apply to HFS input files only.

Undefined | Fixed | Variable
Identify the format of input HFS records.
Undefined indicates that records are terminated by an End-of-Line (EOL) string.
Fixed indicates that all records are of a fixed length as defined by a specified LRECL.
Variable indicates that all records are of variable length as defined by a length field within the data.

Max Record Length>
Applicable to each of the record formats, this value defines the LRECL (maximum length) of input records. A record longer than this value will be chopped into multiple records.
A 0 (zero) value implies the default which is 32752 for Undefined and Variable record formats and 80 for Fixed record format.

EOL Characters>
Applicable to Undefined record format only, choose from one of the following EOL character combinations:

STD	-	Any standard line-end.
CR	X'0D'	Carriage Return.
LF	X'0A'	Line Feed.
NL	X'15'	New Line.
CRLF	X'0D0A'	Carriage Return + Line Feed.
LFCR	X'0A0D'	Line Feed + Carriage Return.
CRNL	X'0D15'	Carriage Return + New Line.
user	-	A 2-byte user string specified in **EOL user string>**

EOL user string>
Applicable only if **EOL Characters>** is set to **user**, this field specifies the user supplied 2-byte EOL string. It may be specified in character or hexadecimal notation. (e.g. '##', X'FFFF')

Var Length Field>
Applicable to Variable record format only, these fields identify the location of the record length fields within the data.

(Offset)
Offset of the record length field from the start of the record. Default is 0. (i.e. the length field is at the start of the record.)

(Length)
Length (number of bytes) of the record length field. Default is 2.

(Data Origin)
Offset into the record data at which the value in the record length field is to be applied. Default is 0. (i.e. the record length include the length field.)

Panel Fields - Output CSV Text File

Output CSV Text File:
Input fields which together identify a single output sequential, VSAM or PDS/PDSE library data set, HFS file or PDS/PDSE library member.

DSN/Path>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

If a DSN is specified for a data set that does not already exist, a prompt data set dialog will be opened to allocate the new output file.

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Volume>
Specifies the name of the output data set volume. This is required only if output is to an uncataloged data set.

Append to existing Output
Select this option if the generated records are to be appended to existing records in the output data set.

HFS Output Options:

EOL Characters>
Choose from one of the following End-Of-Line character combinations:

NL	X'15'	New Line.
CR	X'0D'	Carriage Return.
LF	X'0A'	Line Feed.
CRLF	X'0D0A'	Carriage Return + Line Feed.

Options:

Separator character>
By default CSVGEN produces **comma** separated variables, but this option allows the user to specify any other character as the variable separator. The option may be specified as a single quoted or unquoted character literal, or as a hex value using **X'nn'** notation.

Quoted strings>
The **QUOTE** option controls when variable values are to be enclosed in double-quotes.

CHARacter	Quote character fields values only (default).
ALL	Quote all field values.
REQuired	Quote only if required i.e. if value contains a double-quote or the separator character.

CCSID Conversion>
Since the purpose of CSVGEN is to produce a portable export version of the data in a z/OS mainframe structured data file, and the output is character data, the coded character set identifiers (CCSIDs) of the input, output and of the CSVGEN internal constants themselves are of significance.

Even if the input and output is coded in an EBCDIC CCSID, these may differ, and both may differ from the CCSID of the CSVGEN internal constants. Since some of the special characters used in CSV have different code points in different EBCDIC CCSIDs (for example square brackets) these must be dealt with consistently to produce correct CSV output.

CSVGEN uses the z/OS character conversion support supplied by IBM modules CUNLINFO (for obtaining CCSID information) and CUNLCNV (for character conversion from one CCSID to another).

The internal CSVGEN CCSID (that of the constants used to build the CSV syntax) is CCSID 285 (EBCDIC, SBCS UNITED KINGDOM).

CSVGEN assumes a default CCSID as follows:

Interactive
When executed interactively CSVGEN uses as default input CCSID that of the user's 3270 terminal.

Batch
When executed in batch CSVGEN uses as default input CCSID the value of the INI file variable **SDE.CCSID**. This variable is set automatically to the user's 3270 terminal CCSID (if not already set) during an interactive session. It can also be set using the structured data SET CCSID command.

Avaliable options are as follows:

NONE
The CSV output dataset is produced using the default CCSID and the input dataset character fields are assumed to be in the same CCSID. The internal CSVGEN constants are converted from internal CCSID 285 to the default CCSID.

ASCII
Convert the output to ASCII. This is equivalent to specifying **CONVERT TO 819**. CCSID 819 is ISO 8859-1 ASCII.

UNICODE
Convert the output to UNICODE (UTF-16). This is equivalent to specifying **CONVERT TO 1200**. CCSID 1200 is the IBM bigendian UTF-16 CCSID which is automatically transformaed to the most recent UTF-16 standard.

CONV
from>
from_ccsid
The input character data fields are converted from this CCSID. If not supplied the default input CCSID is used.

to>
to_ccsid
The CCSID of the output CSV text dataset. Internal CSVGEN character literals and input character data fields (and HFS line end characters if used) are converted to this CCSID.

View option>
When CSVGEN is run interactively this option allows the user to request to view the output when the process completes.

BROWSE
Browse the output CSV dataset.

EDIT
Edit the output CSV dataset using the SELCOPY/i text editor.

NOVIEW
Do not view the output CSV dataset. This option is forced when run in batch.

Suppress output column headers record
This option controls whether a CSV record containing the original field names is generated as the first output record.

Strip trailing blanks
This option controls controls whether trailing blanks are to be stripped from each variable. This option is particularly relevant to fixed length character fields.

Primary Commands

The following primary commands are supported.

FILTER	Open the FILTER dialog for the specified filter type.
BC	Browse the input copybook..
BIF	Browse the input file..
BOF	Browse the output file.
SELECT	Select field-names.

Function Keys

In addition to the standard interactive panel key assignments for scrolling and navigation, the CSVGEN panel supports the following:

F5	SELECT	Select field-names.
F6	FILTER	Open the FILTER dialog for the specified filter type.
F14	EXPAND	Expand an input/output field.
F16	BC	Browse the input copybook..
F19	BIF	Browse the input file.
F20	BOF	Browse the output file.

Generate JSON (=8.18)

JSON Generation Panels

The JSON generation utility panel views (ZZSGJSN0) are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select 'JSON-Gen' (option 18) from the Utilities menu (=8.18)
- Execute primary command JSONGEN (JSON) with no parameters from the command line of any window.

The **JSON Generation** panel allows the user to produce an exportable copy of a structured dataset as JavaScript Object Notation (JSON) text.

The utility reads an input structured dataset and writes an output text dataset consisting of JSON {*name*:*value*} pairs. The JSON *names* correspond to the field names of the copybook/structure applied to the input dataset and the *values* to the field values expressed in character format.

Where the input structure maps more than one record-type, the **VIEW** (F22) primary command may be issued to open the Select Record-Types panel and restrict the record-types processed.

The **SELECT** (F5) primary command may also be issued to open the Select Field-Names panel and restrict the fields selected from any given record-type.

The JSONGEN process may be executed in the foreground or as a batch job.

Menu Bar Items

`File`
: The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values entered in the panel so that they may be redisplayed the next time the panel is opened.

`Command`
: Generate the JSONGEN command line syntax and display it in a temporary CMX file text edit view. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

  The user has the opportunity to edit the command prior to its execution and/or copying it to the home (CMX) command centre for future reference and re-execution.

`JCL`
: Generate a JCL job stream that executes the **SDEAMAIN** program with input (SDEIN) containing the JSONGEN command generated for the specified panel field values.

  The job stream is displayed in a temporary text edit view and may be submitted to batch using the SUBMIT command.

`Structure`
: Opens the Create Structure (SDO) Menu to generate a SELCOPY/i SDO from a source COBOL or PL1 Copybook or an XREF file.

`Replace`
: Opens the COBOL Compiler Options panel to review and, if necessary, add COBOL REPLACE "From" and "To" pseudo-text values to be used in compiling a COBOL copybook.

  Values entered in this panel apply only to the current user. System wide COBOL REPLACE values may also have been entered in the SELCOPYi Site INI file. (See the *"SELCOPY Product Suite Customisation Guide"* for details.)

`Help`
: Display help for this panel view.

Panel Fields - Source Structured Data File

`Input Data File:`
: Input fields which together identify a single existing, sequential or VSAM data set, GDG file generation, HFS file or PDS/PDSE library member from which JSON output is to be generated.

  `Name>`
  : Identifies the fully qualified data set name or an absolute or relative HFS file path.

    A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Volume>
Specifies the name of the input data set volume. This is required only if input is from an uncataloged data set.

Structure/Copybook overlay:
Defines fields which together specify a cataloged structure file (COBOL or PL1 Copybook, ADATA file or a SELCOPY/i SDO) used to format the input records. The structure may be a sequential data set or a PDS/PDSE library member.

These fields must be selected and contain valid entries for record data to be assigned tag names.

Dsn>
Identifies the fully qualified data set name of a sequential data set or PDS/PDSE library containing a COBOL or PL1 Copybook, ADATA file or SDO structure.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Type:
Indicate the type of structure (COBOL, PL1, ADATA or SDO).

Recompile>
If *Structure/Copybook overlay* refers to a COBOL, PL1 or ADATA source file (not an SDO) then a compile step must be performed in order to turn the source copybook into SELCOPYi's own internal structure (SDO) format. A temporary SDO will be created, lasting the duration of the SELCOPYi session, making subsequent reference to the same copybook during the session much faster.

If, however, the copybook (or any of its included components) is modified during the session, then a recompile of the source will normally be required.

For performance reasons **N** (meaning no recompile should occur) is the default.

Specify **Y** to force a copybook recompile each time the process is run. Alternatively type the primary command: **SD DROP *<copybook_name>***

Record Selection:
Fields which together identify criteria by which a subset of records from the file are selected for browse/edit.

Start>
If activated, the **Start>** field identifies the first record in the file at which JSON generation will start. Records occurring sequentially before the start record will be excluded. If this field is not activated, records are selected beginning at record 1.

This input field may contain a record number, an RBA number (for ESDS input only), or a key string (for KSDS input only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

Record | Key | RBA
Identifies the type of start value specified in the **Start>** field. Enter "/" in the appropriate, mutually exclusive parameter field.

For>
If activated, the **For>** field specifies the maximum number of records within the file to be processed. If this field is not activated, records are selected beginning at start record and ending at the last record in the file.

Filter>
If activated, the **Filter>** field specifies options to either generate a new record filter or use an existing record filter file. A record filter will perform further subsetting on input file records selected for processing by the Start> and/or For> input fields.

Filter options are as follow:

Q	On executing the **FILTER** command (F6), the Quick Filter dialog panel will be opened in order to generate a temporary filter on the unformatted record data.
F	Use a permanent filter identified by the sequential data set or library member identified in the File> field. On executing the **FILTER** command (F6), the Create File Filter dialog panel will be opened in order to display the contents of an existing filter file or create and save a new filter file.

If option "F" is selected, then specifiaction of a filter fileid is mandatory.

File>
Identifies the fully qualified data set name of a sequential data set or PDS/PDSE library containing a record filter. Quotes are unnecessary but permitted.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the File> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

HFS Input Options:
Options and values that apply to HFS input files only.

Undefined | Fixed | Variable
Identify the format of input HFS records.
Undefined indicates that records are terminated by an End-of-Line (EOL) string.
Fixed indicates that all records are of a fixed length as defined by a specified LRECL.
Variable indicates that all records are of variable length as defined by a length field within the data.

Max Record Length>
Applicable to each of the record formats, this value defines the LRECL (maximum length) of input records. A record longer than this value will be chopped into multiple records.
A 0 (zero) value implies the default which is 32752 for Undefined and Variable record formats and 80 for Fixed record format.

EOL Characters>
Applicable to Undefined record format only, choose from one of the following EOL character combinations:

STD	-	Any standard line-end.
CR	X'0D'	Carriage Return.
LF	X'0A'	Line Feed.
NL	X'15'	New Line.
CRLF	X'0D0A'	Carriage Return + Line Feed.
LFCR	X'0A0D'	Line Feed + Carriage Return.
CRNL	X'0D15'	Carriage Return + New Line.
user	-	A 2-byte user string specified in **EOL user string>**

EOL user string>
Applicable only if **EOL Characters>** is set to **user**, this field specifies the user supplied 2-byte EOL string. It may be specified in character or hexadecimal notation. (e.g. '##', X'FFFF')

Var Length Field>
Applicable to Variable record format only, these fields identify the location of the record length fields within the data.

(Offset)
Offset of the record length field from the start of the record. Default is 0. (i.e. the length field is at the start of the record.)
(Length)
Length (number of bytes) of the record length field. Default is 2.

(Data Origin)
Offset into the record data at which the value in the record length field is to be applied. Default is 0. (i.e. the record length include the length field.)

Panel Fields - Output JSON Text File

Output JSON Text File:
Input fields which together identify a single output sequential, VSAM or PDS/PDSE library data set, HFS file or PDS/PDSE library member.

DSN/Path>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

If a DSN is specified for a data set that does not already exist, a prompt data set dialog will be opened to allocate the new output file.

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Volume>
Specifies the name of the output data set volume. This is required only if output is to an uncataloged data set.

Append to existing Output
Select this option if the generated records are to be appended to existing records in the output data set.

HFS Output Options:

EOL Characters>
Choose from one of the following End-Of-Line character combinations:

NL	X'15'	New Line.
CR	X'0D'	Carriage Return.
LF	X'0A'	Line Feed.
CRLF	X'0D0A'	Carriage Return + Line Feed.

Translation Options:

CCSID Conversion>
Since the purpose of JSONGEN is to produce a portable (data-interchange format) version of the data in a z/OS mainframe structured data file, and the output is character data, the coded character set identifiers (CCSIDs) of the input, output and of the JSONGEN internal constants themselves are of significance.

Even if the input and output is coded in an EBCDIC CCSID, these may differ, and both may differ from the CCSID of the JSONGEN command's internal constants. Since some of the special characters used in JSON have different code points in different EBCDIC CCSIDs (for example quotation marks) these must be dealt with consistently to produce correct JSON output.

JSONGEN uses the z/OS character conversion support supplied by IBM modules CUNLINFO (for obtaining CCSID information) and CUNLCNV (for character conversion from one CCSID to another).

The internal JSONGEN CCSID (that of the constants used to build the JSON syntax) is CCSID 285 (EBCDIC, SBCS UNITED KINGDOM).

JSONGEN assumes a default CCSID as follows:

Interactive
When executed interactively JSONGEN uses as default input CCSID that of the user's 3270 terminal.

Batch
When executed in batch JSONGEN uses as default input CCSID the value of the INI file variable **SDE.CCSID**. This variable is set automatically to the user's 3270 terminal CCSID (if not already set) during an interactive session. It can also be set using the structured data SET CCSID command.

Avaliable options are as follow:

NONE
The JSON output dataset is produced using the default CCSID and the input dataset character fields are assumed to be in the same CCSID. The internal JSONGEN constants are converted from internal CCSID 285 to the default CCSID.

ASCII
Convert the output to ASCII. This is equivalent to specifying **CONVERT TO 819**. CCSID 819 is ISO 8859-1 ASCII.

UNICODE
Convert the output to UNICODE (UTF-16). This is equivalent to specifying **CONVERT TO 1200**. CCSID 1200 is the IBM big-endian UTF-16 CCSID which is automatically transformaed to the most recent UTF-16 standard.

CONV

from>

from_ccsid

The input character data fields are converted from this CCSID. If not supplied the default input CCSID is used.

to>

to_ccsid

The CCSID of the output JSON text dataset. Internal JSONGEN character literals and input character data fields (and HFS line end characters if used) are converted to this CCSID.

Tag Name Case>

This input field value controls whether JSON tag names generated from the structure field names are to be upper cased.

DEFAULT

Use the default action as set by the global SDE option, NAMECASE.

TAGUPPER

Tag names are uppercased.

NOTAGUPPER

Tag names are not uppercased. Tags match the field names with no upper case translation.

Miscellaneous Options:

View option>

When JSONGEN is run interactively this option allows the user to request to view the output when the process completes.

BROWSE

Browse the output JSON dataset.

EDIT

Edit the output JSON dataset using the SELCOPY/i text editor.

NOVIEW

Do not view the output JSON dataset. This option is forced when run in batch.

Indentation value>

Nested output JSON names corresponding to the hierarchy of group and elementary data fields in the input structure are indented by a default of one space for each data item level. This option allows the specification of a different indentation value.

Output all field redefinitions:

If the structure defined for the input dataset contains redefined fields this option controls whether the field redefinitions are output.

Split long JSON records

For each elementary input field JSONGEN builds one output record containing the field name and value. Depending on the nature of the input data, relatively long output records may result. If an output record is longer than the allocated logical record length of the output dataset, this option controls how JSONGEN deals with the long output record.

Activated

Split the output record breaking it up into as many logical records as necessary. Records are split at the logical record length irrespective of the record content.

Not activated

Do not split the output record. Rather than truncate the output record JSONGEN terminates with an error message. This is the default.

Primary Commands

The following primary commands are supported.

FILTER	Open the FILTER dialog for the specified filter type.
BC	Browse the input copybook..
BIF	Browse the input file..
BOF	Browse the output file.
VIEW	Select record-types.
SELECT	Select field-names.

Function Keys

In addition to the standard interactive panel key assignments for scrolling and navigation, the JSONGEN panel supports the following:

F6	FILTER	Open the FILTER dialog for the specified filter type.
F14	EXPAND	Expand an input/output field.
F16	BC	Browse the input copybook..
F19	BIF	Browse the input file.
F20	BOF	Browse the output file.
F22	VIEW	Select record-types.
F23	SELECT	Select field-names.

Merge Datasets

Merge Datasets Panel

The **Merge Datasets** panel may be displayed by selecting option 19 from the Utilities menu (=8.19) or by entering the **MERGE** primary command without parameters.

This panel allows the user to combine a number of existing datsets that are already sorted on a common "key" field into a single output dataset, with the records written in ascending key sequence.

The Merge Datasets process may be executed in the foreground or as a batch job.

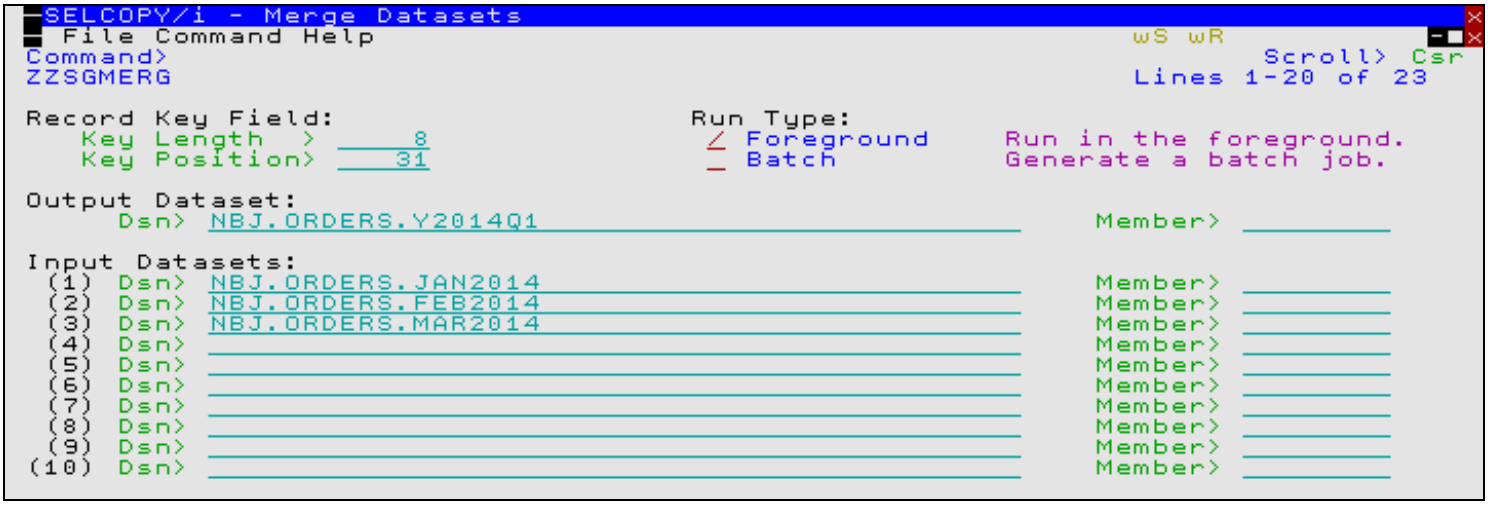


Figure 163. SELCOPYi - Merge Datasets

Menu Bar Items

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values entered in the panel so that they may be redisplayed the next time the panel is opened.

Command
: Generate the Merge Datasets command line syntax and display it in a temporary CMX file text edit view. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

The user has the opportunity to edit the command prior to its execution and/or copying it to the home (CMX) command centre for future reference and re-execution.

Help
: Display help for this panel view.

Panel Input Fields

Key Length>
: The length of the common **"key"** field within every input record, on which all datasets are assumed to be pre-sorted into ascending (character) sequence.

Key Position>
: The position within each input record of the common **"key"** field, on which all datasets are assumed to be pre-sorted into ascending (character) sequence.

Run Type: Batch
: A batch job executing **PGM=SELCOPY** will be generated and presented in a Text-Edit window. Return Code 1 from the submitted batch job alerts to the presence of out of sequence input records.

Run Type: Foreground
: Run the dataset merge process in the foreground. On completion a message will be displayed indicating the number of input records processed. An additional message will be displayed if any out of sequence records were encountered.

Output Dataset:
: Input fields which together identify a single sequential or VSAM file, GDG file generation or PDS/PDSE library member to which all input records will be copied in ascending key sequence.

Dsn>
Identifies the fully qualified data set name.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the name of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Input Datasets:
Up to 10 input datasets may be specified, each identifing a single sequential or VSAM file, GDG file generation or PDS/PDSE library member assumed to be pre-sorted into ascending (character) key sequence.

Dsn>
Identifies the fully qualified data set name.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the name of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Primary Commands

The following primary commands are supported.

OUTPUT (O)	Browse the output dataset.
I1	Browse input dataset 1.
I2	Browse input dataset 2.
I3	Browse input dataset 3.
I4	Browse input dataset 4.
I5	Browse input dataset 5.
I6	Browse input dataset 6.
I7	Browse input dataset 7.
I8	Browse input dataset 8.
I9	Browse input dataset 9.
I10	Browse input dataset 10.

Function Keys

In addition to the standard interactive panel key assignments for scrolling and navigation, the Merge Datasets panel supports the following:

F5	SELECT	Select field-names.
F6	FILTER	Open the FILTER dialog for the specified filter type.
F14	EXPAND	Expand an input/output field.
F16	BC	Browse the input copybook..
F19	BIF	Browse the input file.
F20	BOF	Browse the output file.

List/Delete PDSE v2 Orphaned Member Generations (=8.17)

The **List/Delete PDSE v2 Orphaned Member Generations** panel may be displayed by selecting option 20 from the Utilities menu (=8.20) or by entering the **GENORPH** primary command without parameters.

This panel allows the user to identify and optionally to automatically delete orphaned member generations for a given library.

Orphans are those members that no longer have a generation zero (it has been deleted or renamed).

The **GENORPH** process may be executed in the foreground or as a batch job.

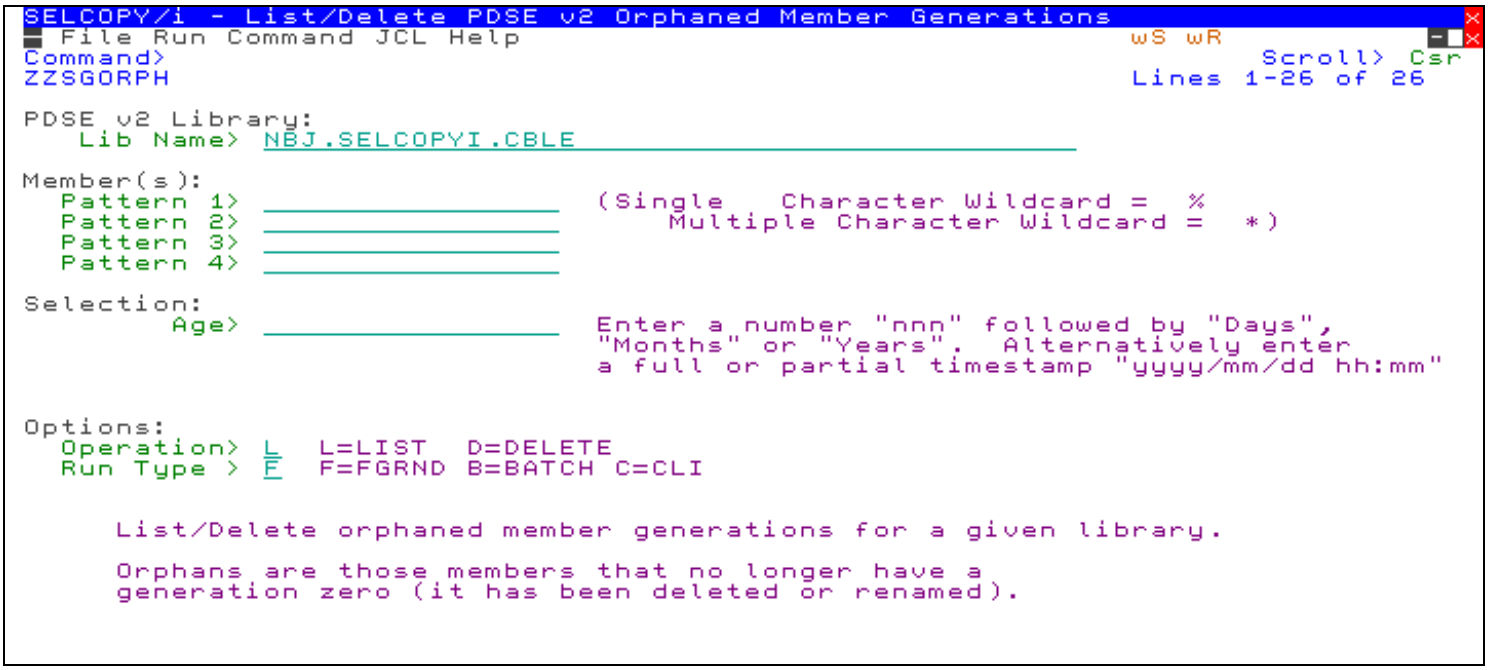


Figure 164. SELCOPYi - List/Delete PDSE V2 Orphaned Member Generations

Menu Bar Items

File
The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values entered in the panel so that they may be redisplayed the next time the panel is opened.

Run
Run the GENORPH process in the foreground.

Command
Generate the GENORPH command line syntax and display it in a temporary CMX file text edit view. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

The user has the opportunity to edit the command prior to its execution and/or copying it to the home (CMX) command centre for future reference and re-execution.

Command
Generate the JCL required to run the GENORPH process in batch.

Help
Display help for this panel view.

Panel Input Fields

DSN/Path>
Identifies the fully qualified data set name of a **PDSE version 2** library defined with **MAXGENS > 1** from which orphaned members will be identified.

Dataset names beginning with "." (dot) will be treated as though they actually begin with your own userid/hlq.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Members(s):
Pattern 1/2/3/4>

These input fields (Pattern 1/2/3/4) allow the user to provide up to 4 alternative member name masks in order to restrict the the identification of orphaned members.

A member name mask supports the following wild cards:

*	A single asterisk represents an entire member name or zero or more characters within a member name mask.
%	A single percent sign represents exactly one character within a member name mask. Up to 8 percent signs can be specified in each member name mask.

If no member name masks are specified, then all orphaned members will be identified.

Age>

Use this option to select only member generations that are older than the specified age.

The "Last Modified" timestamp for each member generation will be checked against this value and any that are later than this will be ignored.

A date relative to the current day may be supplied as **nnn** or **-nnn** where nnn is a number of days (default), months or years. Follow nnn with **MONTHS (M)** or **YEARS (Y)** to override the unit default of **DAYS (D)**

e.g. If today were **2016/03/05** then **5** would be treated as **2016/02/29** (leap year).

Alternatively you may specify latest timestamp for selection in **yyyy/mm/dd hh:mm** format, which may be specified in as much detail as necessary.
e.g. **"2018/09"** will be treated as **"2018/09/99 99:99"**

Operation> L=LIST | D=DELETE

L indicates that orphaned members will be displayed in a list-window if running interactively under SELCOPYi, or have their name/generation printed if running in batch (SDEAMAIN).

From a SELCOPYi list window individual members may be viewed or deleted using standard line-commands. To delete a block of members specify //**D** on the first and // on the last line of the block.

D indicates that the identified orphaned members are to be **deleted**.

Run Type> F=FGRND | B=BATCH | C=CLI

F indicates that immediate foreground execution is required as soon as the **ENTER** key is pressed.

B indicates that JCL should be produced for submission to batch.

C indicates that command line interface should be produced. The GENORPH primary command is displayed in a Text-Edit window in a format suitable for execution using the ACTION key (Shift-F4) ready to be copied into your HOME file (=4).

Data Set Information

The Data Set Information panels (DSINFONV) are interactive panel windows.

Detailed information may be displayed for DB2 tables and all types of file objects. For file objects, the information is initially displayed using data set information panels that have different contents depending on the type and organisation of file object selected.

Furthermore, the information may also be presented in a formatted, HTML report containing useful hyperlinks and which is automatically displayed in a SELCOPYi help window. This is the default for DB2 table information.

Dataset and HFS File Objects

A file object's information panel may be displayed using either of the following methods:

1. From a file list, enter prefix command "I" against the required file name.
2. From a Text Editor or Data Editor view of a file, enter "DSI" with no parameters to display information for the current file.
3. Enter primary command "**DSI *fileid***" where *fileid* is the name of the required data set or GDG Base catalog entry.

Depending on the data set type, information may span several panel views.

Use the <Enter> key to progress through the panel view pages and primary command **BACK** (F3 default) to redisplay the previous panel view.

DB2 Table Objects

A DB2 table object's information report may be displayed using either of the following methods:

1. From a DB2 table list, enter prefix command "I" against the required table name.
2. From a Data Editor view of a DB2 table, enter "DSI" or "INFO" with no parameters to display information for the current table.
3. Enter primary command "**DSI DB2(*ssn*) *table***" where *ssn* is the local DB2 sub-system and *table* is the 1, 2 or 3 part DB2 table name. If "(*ssn*)" is not specified, the current subsystem default is used.

Dataset Information - Non-VSAM

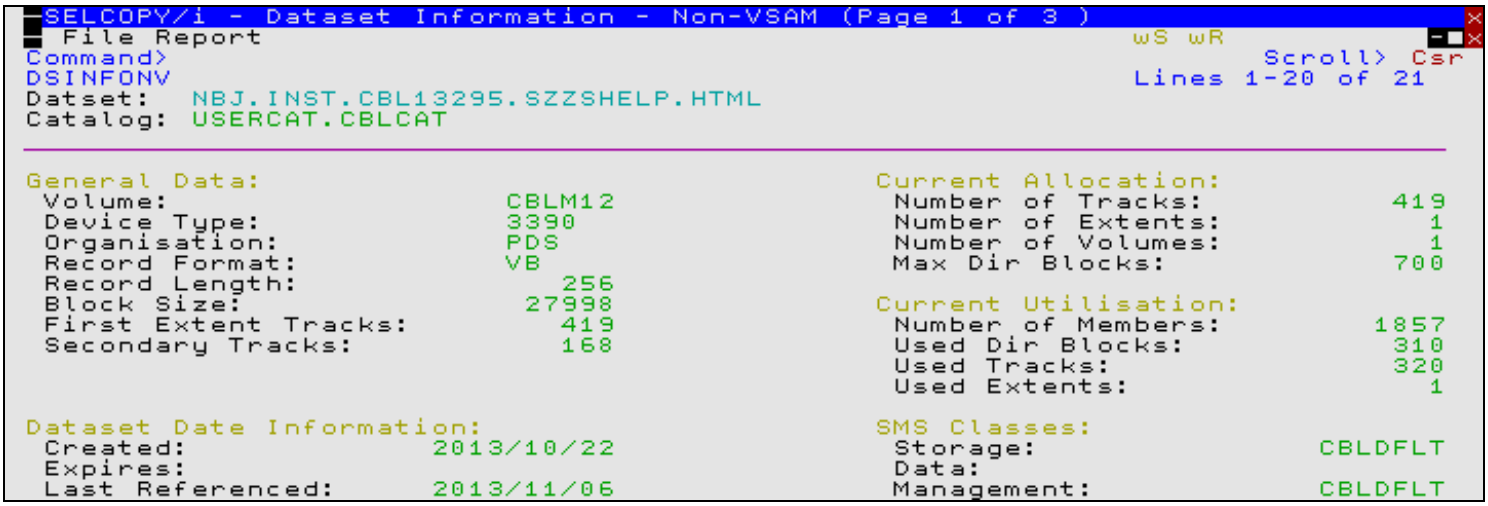


Figure 165. Data Set Information Panel - PDS.

Menu Bar Items

The following menu bar items may be displayed depending on the type and organisation of the file.

File

The File drop-down menu contains the single item "Exit" which simply closes the panel window.

Report
Display all information obtained for the file in a single formatted report. This report is generated using hyper-text markup language (HTML) and is diplayed in SELCOPY/i's HTML browser. The browser includes the menu item "Text" which will display the report in plain EBCDIC text in a text edit view. This text output may subsequently be saved to DASD and printed.

RecordCount
Executes a foregroud pass of the file to provide an accurate account of the number of records, the lengths of the longest and shortest records and the average record length.

Panel Fields - Dataset Information Non-VSAM (Page 1)

Dataset: | GDS: | GDG:
Fully qualified name of the selected sequential or PDS/PDSE library data set, GDG data set or GDG Base catalog entry.

Base GDG:
Applicable only to GDS data sets, identifies the name of the GDG to which the GDS belongs.

Catalog:
DSN of the catalog in which the data set or GDG base is cataloged.

Owner:
Applicable to GDG Base catalog entries only, this field identifies the GDG owner id.

Generations: (GDG only)
Maximum:
Reports the total number (1-255) of Generation Data Sets (GDS) that can be associated with the GDG.

Active:
Reports the number of currently active Generation Data Sets (GDS) associated with the GDG.

Action at Limit: (GDG and GDS only)
Roll Off:
Identifies the action taken when a new GDS is generated which results in the maximum defined number of GDS threshold being exceeded. **OLDEST** indicates that the oldest generation GDS is deleted, **ALL** indicates that all associated GDS are deleted.

Volume Scratch:
Identifies the action taken when a GDS entry is deleted. **YES** indicates that the GDS will be scratch from the volume as well as being uncataloged. **NO** indicates that the GDS will remain on the volume but will be uncataloged.

General Data:
Volume:
Reports the volume id of the first or only volume on which the data set is saved.
If the dataset spans more than one volume, then the volume ID is followed by a "+" (plus) symbol.

Device Type:
Reports the volume device type. (e.g. 3390)

Organisation:
Reports the data set organisation.

Record Format:
Reports the defined data set record format.

Record Length:
Reports the defined maximum record length.

Block Size:
For data sets of blocked record format, reports the allocated block size.

First Extent Blocks/Cylinders/Tracks:
Reports the number of space allocation units (blocks, cylinders or tracks) that constitute the first extent allocated on the volume. This is usually equivalent to the defined Primary Allocation value, however, the sytem may use up to 3 extents to perform a primary allocation.

Secondary Blocks/Cylinders/Tracks:
Reports the number of space allocation units (blocks, cylinders or tracks) that constitute the defined Secondary Allocation value.

Sripe Count:
Displayed only for extended data sets containing data that is striped across a number of volumes, this field reports the number of stripe volumes.

Dataset Date Information:
Created:
Reports the date on which the data set or GDG was defined.

Expires:
Reports the date on which the data set expires. This field is not applicable to GDG base entries.

Last Referenced:
Reports the date on which the data set was last opened. This field is not applicable to GDG base entries.

Active: (GDG only)
Lists the data set names of the associated GDS data sets. Press <Enter> on any of these entries to open a data set information panel for the individual GDS.

Current Allocation:

Number of Tracks:
Reports the total number of allocated tracks.

Number of Extents:
Reports the total number of allocated extents.

Number of Volumes:
Reports the total number of volumes on which space has been allocated. This number does not include candidate volumes on which space has not yet been allocated for the data set.

Max Dir Blocks:
Applicable only to PDS data sets, reports the number of allocated directory blocks. Each directory block is 256 bytes in length.

Maximum Pages:
Applicable only to PDSE data sets, reports the number of PDSE 4KB pages allocated to the data set.

Maximum Generations:
Applicable only to GDS data sets, reports the maximum number of GDG generation data sets that may be associated with the GDG before GDS deletion occurs.

Current Utilisation:

Used Tracks:
Reports the number of tracks used.

Used Extents:
Reports the number of extents used.

Number of Members:
Applicable only to PDS and PDSE data sets, reports the number of library members.

Used Dir Blocks:
Applicable only to PDS data sets, reports the number of used directory blocks.

Used Pages:
Applicable only to PDSE data sets, reports the number of PDSE 4KB pages used.

Percent Used Pages:
Applicable only to PDSE data sets, reports the number of PDSE 4KB pages used as a percentage of the maximum pages value.

SMS Classes:

Storage:
Reports the SMS storage class assigned to the data set.

Data:
Reports the SMS data class assigned to the data set.

Management:
Reports the SMS management class assigned to the data set.

Panel Fields - Dataset Information Non-VSAM (Page 2)

This display lists the volumes allocated to both the data set.

Details including extent information may viewed for each volume by placing the cursor on the volume name then pressing the <Enter> key.

Volumes:
Total number of volumes allocated to the data set.

Panel Fields - Dataset Information Non-VSAM (Page 3-*n*)

The remaining pages in the sequence display volume information, one page for each volume assigned to the data set.

Dataset:
The fully qualified data set name.

Volume:
The volume sequence number for the data set.

Volume Name:
The volume id.

Device Type:
The volume disk device type (e.g. 3390) followed by the device type code and one of the following in parentheses.

Candidate
The volume is a candidate for storing data set extents.

Primary
The volume is a volume on which data records are stored.

Volume Definition:
Blocks per Track:
The number of physical records that can be written on a track on the volume.

Bytes per Track:
The total number of bytes per track. Note that not bytes may be used for data storage.

First Extent Blocks/Cylinders/Tracks:
Reports the number of space allocation units (blocks, cylinders or tracks) that constitute the first extent allocated on the volume. This is usually equivalent to the defined Primary Allocation value, however, the sytem may use up to 3 extents to perform a primary allocation.

Secondary Blocks/Cylinders/Tracks:
Reports the number of space allocation units (blocks, cylinders or tracks) that constitute the defined Secondary Allocation value.

Dataset Date Information:
Created:
Reports the date on which the data set was defined.

Expires:
Reports the date on which the data set expires.

Last Referenced:
Reports the date on which the data set was last opened.

Extents:
The total number of extents on this volume followed by a table containing information for each extent.

Seq:
The extent sequence number.

Tracks:
The number of tracks in the extent.

Low Cyl:
The cylinder number containing the first track of the extent.

Low Track:
The number of the first track of the extent.

High Cyl:
The cylinder number containing the last track of the extent.

High Track:
The number of the last track of the extent.

Tracks Alloc:
The number of tracks allocated for the data set on this volume.

Tracks Used:
The number of allocated tracks containing data for the data set on this volume.

Dataset Information - VSAM

The Dataset Information window may be opened via the following:

- Enter the primary command DSI.
- Enter the line-command "I" where supported from a List type window.

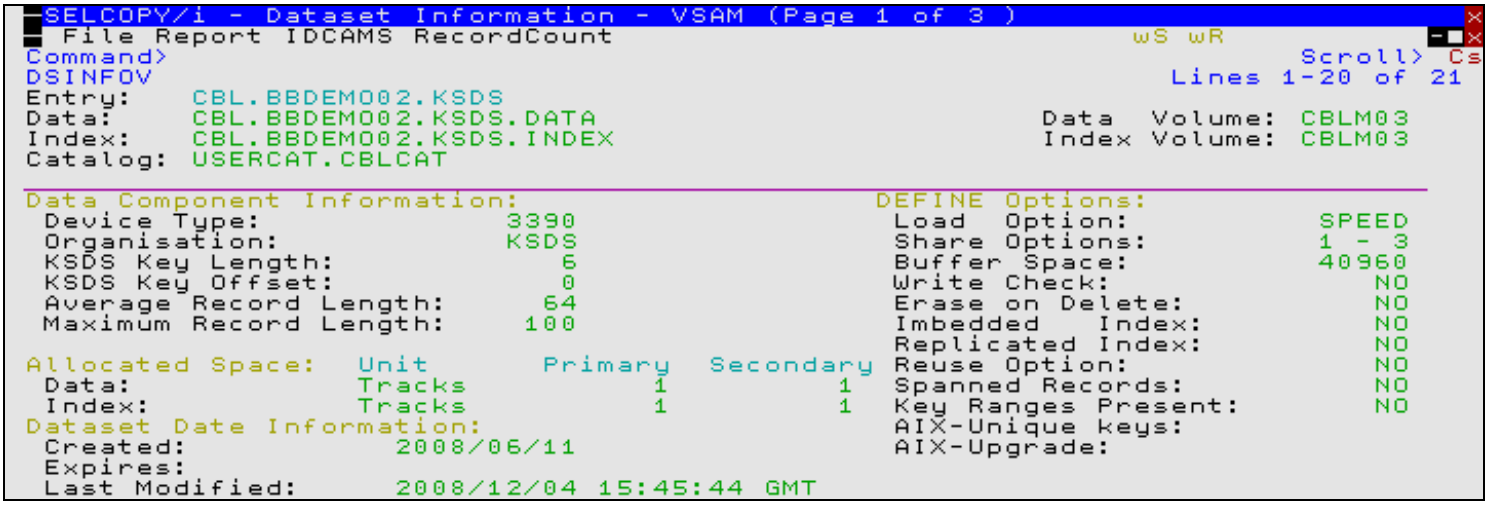


Figure 166. Data Set Information Panel - KSDS.

Menu Bar Items

The following menu bar items may be displayed depending on the type and organisation of the file.

File
The File drop-down menu contains the single item "Exit" which simply closes the panel window.

Report
Display all information obtained for the file in a single formatted report. This report is generated using hyper-text markup language (HTML) and is diplayed in SELCOPY/i's HTML browser. The browser includes the menu item "Text" which will display the report in plain EBCDIC text in a text edit view. This text output may subsequently be saved to DASD and printed.

IDCAMS
Opens the "Execute IDCAMS" utility window to generate an LISTCAT report for the file.

RecordCount
Executes a foregroud pass of the file to provide an accurate account of the number of records, the lengths of the longest and shortest records and the average record length.

Panel Fields - Dataset Information VSAM (Page 1)

Entry:
Fully qualified name of the VSAM cluster, alternate index or path.

Data:
Fully qualified name of the VSAM data component.

Index:
Fully qualified name of the VSAM index component.

Catalog:
Catalog name.

Data Volume:
Name of the volume containing the data component. If the data is defined across multiple volumes then a **"+"** (plus sign) is displayed following the name of the first volume.

Index Volume:
Name of the volume containing the index component. If the index is defined across multiple volumes then a **"+"** (plus sign) is displayed following the name of the first volume.

Data Component Information:
Device Type:
Disk device type on which the data component is stored.

Organisation:
Type of VSAM entry - KSDS, ESDS, RRDS, VRDS (Variable-length RRDS), AIX or PATH, followed by EXTENDED and/or COMP (compressed) if appropriate.

AIX/KSDS/VRDS Key Length:
Length of key for a Alternate-Index (AIX), Key Sequence Dataset (KSDS) or Variable-length Relative record Dataset (VRDS). For VRDS files the key is always length 4 (representing the file record number) and does not form part of the record presented by VSAM to the user.

AIX/KSDS/VRDS Key Offset:
Key location (relative to 0) for a Alternate-Index (AIX), Key Sequence Dataset (KSDS) or Variable-length Relative record Dataset (VRDS). For VRDS files the key is always at offset 0, length 4 (representing the file record number) and does not form part of the record presented by VSAM to the user.

Average Record Length:
Defined average record length.
To determine the dataset's actual average record length, select *RecordCount* from the menu-bar.

Maximum Record Length:
Defined maximum record length.
To determine the length of the longest actual dataset record, select *RecordCount* from the menu-bar.

Dataset Date Information:

Created:
Date on which the datset was defined.

Expires:
Date on which the dataset expires.

Last Modifed:
Date and time at which the dataset was most recently modified.

DEFINE Options:

Load Option:
VSAM define option (either SPEED or RECOVERY) that specifies whether the data component's control areas are to be preformatted during initial load.

SPEED
DATA CA's are not preformatted. If the initial load is unsuccessful, you must load the data set again from the beginning because VSAM cannot determine the location of your last correctly written record.

RECOVERY
DATA CA's are preformatted. If the initial load is unsuccessful, VSAM can determine the location of the last record written during the load process.

Share Options:
Describes how the dataset is be shared among users, with the first number indicating how it is shared across regions, and the second how it is shared across systems.

Cross Region

1 The data set can be shared by any number of users for read processing, or the data set can be accessed by only one user for read and write processing. VSAM ensures complete data integrity for the data set. This setting does not allow any non-RLS access when the data set is already open for VSAM RLS or DFSMStvs processing. A VSAM RLS or DFSMStvs open will fail with this option if the data set is already open for any processing.

2 The data set can be accessed by any number of users for read processing, and it can also be accessed by one user for write processing. It is the user's responsibility to provide read integrity. VSAM ensures write integrity by obtaining exclusive control for a control interval while it is being updated. A VSAM RLS or DFSMStvs open is not allowed while the data set is open for non-RLS output. If the data set has already been opened for VSAM RLS or DFSMStvs processing, a non-RLS open for input is allowed; a non-RLS open for output fails. If the data set is opened for input in non-RLS mode, a VSAM RLS or DFSMStvs open is allowed.

3 The data set can be fully shared by any number of users. Each user is responsible for maintaining both read and write integrity for the data the program accesses. This setting does not allow any non-RLS access when the data set is already open for VSAM RLS or DFSMStvs processing. If the data set is opened for input in non-RLS mode, a VSAM RLS or DFSMStvs open is allowed.

4 The data set can be fully shared by any number of users. For each request, VSAM refreshes the buffers used for direct processing. This setting does not allow any non-RLS access when the data set is already open for VSAM RLS or DFSMStvs processing. If the data set is opened for input in non-RLS mode, a VSAM RLS or DFSMStvs open is allowed. As for share option 3, each user is responsible for maintaining both read and write integrity for the data the program accesses.

Cross System

1 Reserved.

2 Reserved.

3 Specifies that the data set can be fully shared. With this option, each user is responsible for maintaining both read and write integrity for the data that user's program accesses.

4 Indicates that the data set can be fully shared. For each request, VSAM refreshes the buffers used for direct processing.

Buffer Space:
Minimum space required for buffers.

Write Check:
Indicates whether the cluster or component is to be checked by a machine action called write check when a record is written into it

Erase on Delete:
Indicates whether data records are to be erased when the entry is deleted from the catalog.

Imbedded Index:
Indicates whether the cluster was defined with the obsolete IMBED parameter, specifying that the index is stored in the data control area.

Replicated Index:
Indicates whether the cluster was defined with the obsolete RELICATE parameter.

Reuse Option:
Indicates whether the cluster was defined with the REUSE parameter, meaning it can be reloaded.

Spanned Records:
Indicates that logical data records can be longer than a control interval and may span multiple CIs.

Key Ranges Present:
Indicates whether the cluster was defined with the obsolete RELICATE parameter. Detail of any defined key ranges may be viewed by selecting **Report** from the menu-bar.

AIX-Unique Keys:
Indicates whether more than one data record (in the base cluster) can contain the same key value for the alternate index.

AIX-Upgrade:
Indicates whether the alternate index is to be automatically kept up to date when its base cluster is modified.

Panel Fields - Dataset Information VSAM (Page 2)

Entry:
Fully qualified name of the VSAM cluster, alternate index or path.

Owner:
The dataset's owner id.

SMS Classes:
- **Storage:** The name of the SMS Storage class assigned to the cluster.
- **Data:** The name of the SMS Data class assigned to the cluster.
- **Management:** The name of the SMS Management class assigned to the cluster.

Current Allocation:
- **Allocated Tracks:** Total number of tracks allocated to the data component.
- **Allocated Extents:** Total number of extents allocated to the data component on all volumes.
- **Volumes:** Total number of volumes allocated to the data component.

KSDS Index Allocation:
- **Allocated Tracks:** Total number of tracks allocated to the index component.
- **Number of records:** Total number of records in the index component.

Current Utilisation:
- **Used Data-Space:** Estimated percentage used, calculated as the High Allocated RBA, divided by the High Used RBA minus Free-Bytes.
- **Used Extents:** Total number of extents used across all volumes.

Total records:
Total number of records in the data component.

Deleted records:
The number of records that have been deleted from the data component.

Inserted records:
The number of records that have been inserted into the data component.

For a KSDS, the number of records that have been inserted into the data component before the last record. Records originally loaded and records added to the end are not included in this value.

For RRDS/VRDS, this is the number of records inserted into available slots. The number of records originally loaded are included in this statistic.

Updated records:
The number of records in the data component that have been updated.

This value does not reflect those records that were deleted, but a record that is updated and then deleted is counted in this value.

Control Area Information:

Physical Record Size:
The number of bytes that VSAM uses for a physical record in the data component.

Records per Track:
The number of physical records that VSAM can write on a track on the volume.

Tracks per CA:
The number of track per Control-Area.

Retrieved records:
The number of records that have been retrieved from the data component.

Max record number:
Identifies the highest possible valid relative record number, for a relative record data set.

Number of CA Splits:
The number of times a Control-Area was split. i.e. half the data records in the Control-Area were written into a new Control-Area and then deleted from the old Control-Area.

Control Interval Information:

Size -Data: / -Index:
The number of bytes in a Control-Interval for the Data and Index components.

Number of CIs per CA:
The number of Control-Intervals per Control-Area.

Number of free CIs per CA:
The number of free Control-Intervals per Control-Area.

Percentage free CIs per CA:
The percentage of Control-Intervals kept free per Control-Area to prevent subsequent CI-splits from causing a CA-split.

Percentage free bytes per CI:
Percentage of space to be left free in a Control-Interval for subsequent processing.

Number of CI splits:
The number of times a Control-INterval was split. i.e. half the data records in the Control-Interval were written into a new Control-Interval and then deleted from the old Control-Interval.

Panel Fields - Dataset Information VSAM (Page 3)

This display lists the volumes allocated to both the DATA and INDEX components of the dataset.

Details including extent information may viewed for each volume by placing the cursor on the volume name then pressing the ENTER key.

Panel Fields - Dataset Information VSAM (Page 4)

This display lists the dataset's related entries i.e. each path and/or alternate index

Panel Fields - Dataset Information VSAM (Page 5-*n*)

The following pages display information for each of the DATA and INDEX volumes in turn.

Data: / Index:
Fully qualified name of the DATA or INDEX component as appropriate.

Volume:
The volume's sequence number.

Volume Name:
The volume name.

Device Type:
The volume disk device type e.g. 3390, followed by the device type code and one of the following in brackets.

CANDIDATE
The volume is a candidate for storing the data or index component.

CAND-SPACE
The volume is a candidate for storing the data or index component, and it has a primary extent preallocated (the data set was defined with a guaranteed-space storage class).

OVERFLOW
The volume is an overflow volume on which data records in a key range are stored. The KEYRANGE begins on another (PRIME) volume.

PRIME
The volume is the first volume on which data records in a key range are stored.

Volume Definition:
Blocks per Track:
The number of physical records that VSAM can write on a track on the volume.

Tracks per CA:
The number of tracks in a control area for the component being listed. (This value is computed when the entry is defined. This value reflects the optimum size of the control area for the given device and the nature of the entry, whether indexed, nonindexed, or numbered.) For a key-sequenced data set with the imbedded attribute, this value includes the sequence set track.

Physical Record Size:
The number of bytes that VSAM uses for a physical record in the data or index component.

Volume Statistics:
High Used RBA:
The highest RBA (plus 1) within allocated space that actually contains data component, its key range, the index component, or the sequence set records of a key range. (The RBA of the next completely unused control interval.)

High Allocated RBA:
The highest RBA (plus 1) available within allocated space to store data component, its key range, the index component, or the sequence set records of a key range.

High Key CI RBA:
RBA of the data CI containing the dataset's high key.

Extents:
The total number of extents on this volume.

Seq:
The extent sequence number.

Tracks:
The number of tracks in the extent.

Low Cyl:
The extent's low cylinder number.

Low Track:
The extent's low track number.

High Cyl:
The extent's high cylinder number.

High Track:
The extent's high track number.

Low RBA:
The extent's low relative byte address.

High RBA:
The extent's high relative byte address.

DB2 Table Information

The DB2 Table Information report window may be opened via the following:

- Enter the primary command DSI.
- If the current window is a Data Editor view, enter primary command INFO.
- Enter the line-command "I" where supported in a DB2 object list window.

The report is divided into the following:

1. General Table Information
2. Table Statistics
3. Table Columns
4. Column Attributes
5. Table Indexes
6. Parent Tables
7. Dependent Child Tables

General Table Information

`Table:`
The 2 or 3 part table name identifying the schema and name of the DB2 table. If the selected table belongs to a remote DB2 server, the table name includes a location qualifier prefix.

`Type:`
The type of table reference (as defined in the SYSTABLES DB2 catalog table). Possible values are: **Alias**, **Clone Table**, **Global Temporary**, **Materialised Query**, **Implicit XML**, **Base Table**, **View** and **Auxiliary table**.

`In Database:`
Database name to which the table belongs.

`Tablespace:`
Name of the table space in which the table is defined.

`Table columns:`
Number of table columns defined to this table. This report field title links directly to the Table Columns report output.

`Primary key columns:`
Number of columns that constitute the primary key constraint defined for this table. Primary key column names and sequence are identified in the table of columns in the Table Columns report output.

`Table indexes:`
Number of indexes defined for this table. This report field title links directly to the Table Indexes report output.

`Referential constraints:`
Number of referential (parent key/foriegn key) constraints defined on columns in this table. This report field title links directly to the Parent Tables report output.

`Unique constraints:`
Number of unique constraints (excluding primary key constraints) defined for this table.

`Check constraints:`
Number of check constraints defined for this table.

`Parent tables:`
Number of parent tables for which columns within this table that constitute a foreign key, have a parent key relationship. A referential constraint is defined on this table for each parent key to foreign key relationship. This report field title links directly to the Parent Tables report output.

`Child tables:`
Number of dependent tables for which this table is a parent. Referential constraints defined on dependent tables nominate this table as the parent key table. This report field title links directly to the Dependent Child Tables report output.

`Database object id:`
DB2 internal object id for the database to which this table is defined.

`Table object id:`
DB2 internal object id for this table.

`Encoding scheme:`
Character data encoding scheme for this table. (EBCDIC, ASCII or UNICODE)

`Table access audit:`
Type of access to this table for which auditing is performed. (All, Changes or None)

`Edit procedure name:`
Name of the program used to process row data change, load and retrieval operations for this table.

`Validation procedure name:`
Name of the program used to validate row data before load, insert, update or delete of a row belonging to this table.

Label:
Descriptive label of maximum length 30 that has been assigned to this table via an SQL LABEL statement.

Remarks:
Comment string of maximum length 762 that has been assigned to this table via an SQL COMMENT statement.

Created by:
Primary authorisation id of the user who created this table.

Created time stamp:
Date and time at which this table was created.

Table Statistics

RUNSTATS time stamp:
The date and time at which RUNSTATS last updated the statistics information for this table.

Number of rows:
Number of rows in this table when RUNSTATS was last executed.

Pages used:
Number of pages used by this table when RUNSTATS was last executed.

DASD kilobytes:
Number of KB of DASD storage used by this table when RUNSTATS was last executed.

Table Columns

Displays a table containing one row for each column defined to this DB2 table. The information displayed for each column is as follows:

Column Number
The sequence number of the column within the table.

Prime Key Seq
The sequence number of the column within the defined primary key. If the column is not part of the primary key constraint, this field is blank.

Name
The column name.

Type
The built-in data type of data belonging to the column name.

Length or Precision
For columns of DECIMAL data type, this is the precision (total number of decimal digits) of the decimal number. For all other data types, this is the length of the column data.

Scale
Applicable to columns of DECIMAL data type only, this is the number of decimal fraction digits (i.e. number of digits to the right of the decimal point.) For all other data types, this field is 0 (zero).

Type Code
The DB2 internal data type id of the column.

Null
Indicator or whether or not the column supports a NULL value. (Yes or No)

Column Attributes

This section contains a more detailed description of the table column attributes than that displayed in the table of columns under the Table Columns report output.

For each column defined to the table, there is a sub-header "Column *column-name*" followed by information specific to that column.

If the column is assigned a distinct (non built-in) data-type, then the following statement follows the column name sub-header:

```
Column column_name has a user defined DISTINCT data type named
schema.distinct_type which is based on the internal data type datatype.
```

The remainder of the column attributes report follows.

Column Sequence:
The sequence number of the column within the table.

Prime key sequence:
The sequence number of the column within the defined primary key. If the column is not part of the primary key constraint, this field is omitted.

Data type:
The built-in data type of data belonging to the column name.

Distinct type code (hex):
Displayed only if the Data type is DISTINCT, this is the DB2 internal code for the defined distinct type assigned to this column name.

Distinct type:
Displayed only if the Data type is DISTINCT, this is the schema and name of the distinct type assigned to this column name.

Source data type:
Displayed only if the Data type is DISTINCT, this is the source built-in data type of the distinct type assigned to this column name.

Length
Displayed only if the Data type is **not** DECIMAL, this is the length of the column data.

Precision,Scale
Displayed only if the Data type (or Source data type) is DECIMAL, this is the precision (total number of decimal digits) and scale (number of fraction digits) for values in this column

Type code (dec):
Displayed only if the Data type is **not** DISTINCT, this is the DB2 internal code (in decimal) of the built-in data type assigned to this column name.

Type code (hex):
Displayed only if the Data type is **not** DISTINCT, this is the DB2 internal code (in hex) of the built-in data type assigned to this column name.

Source type code (dec):
Displayed only if the Data type is DISTINCT, this is the DB2 internal code (in decimal) of the source built-in data type of the distinct type assigned to this column name.

Source type code (hex):
Displayed only if the Data type is DISTINCT, this is the DB2 internal code (in hex) of the source built-in data type of the distinct type assigned to this column name.

In prime key:
Indicates whether or not the column forms part of the primary key constraint. (Yes or No)

In foreign key:
Indicates whether or not the column forms part of a foreign key in a referential constraint. (Yes or No)

In unique key:
Indicates whether or not the column forms part of a unique index key definition. (Yes or No)

In non-unique key:
Indicates whether or not the column forms part of a non-unique index key definition. (Yes or No)

Can contain nulls:
Indicates whether or not the column can contain null values. (Yes or No)

Can be updated:
Indicates whether or not the column can be updated. (Yes or No)

A column can not be updated if it is derived from a function or expression, has data type ROWID or the table is a read-only table view definition.

Has check constraint:
Indicates whether or not values entered in the column are verified by a check constraint. (Yes or No)

Hidden from SELECT *:
Indicates whether or not the column is hidden when the generic SQL query, SELECT *, is performed. (Yes or No)

CCSID:
For columns of character or graphic data types, displays the CCSID associated with string data otherwise 0 (zero).

Character subtype:
Displayed only for columns of character data types, this is the sub-type assigned to character string data. (SBCS, Mixed or Bit)

Has default value:
Indicates whether or not the column has a default value. (Yes or No)

Has field procedure:
Indicates whether or not the column has a defined field procedure for encoding and decoding column values. (Yes or No)

Label:
Descriptive label of maximum length 30 that has been assigned to this table column via an SQL LABEL statement.

Remarks:
Comment string of maximum length 762 that has been assigned to this table column via an SQL COMMENT statement.

Default value type:
Displayed only for columns that have a default value, this is the single character default value type code as specified in the DB2 catalog table SYSCOLUMNS.

Default type description
Displayed only for columns that have a default value, this statement identifies the default value type as interpreted from the default value code. Possible types are:

- ◊ Unrecognised default type
- ◊ ROWID data type GENERATED ALWAYS
- ◊ Default value is defined by column data type
- ◊ ROWID data type GENERATED BY DEFAULT
- ◊ FOR EACH ROW ON UPDATE GENERATED ALWAYS
- ◊ FOR EACH ROW ON UPDATE GENERATED BY DEFAULT
- ◊ Column defined as IDENTITY GENERATED ALWAYS
- ◊ Column defined as IDENTITY GENERATED BY DEFAULT
- ◊ Column defined for implicit DOCID for XML data
- ◊ Column defined AS SECURITY LABEL
- ◊ Column has no default value
- ◊ Default value is SQL authorisation id
- ◊ Default value is SESSION_USER special register
- ◊ Default value is Null
- ◊ Default value is string constant:
- ◊ Default value is floating point constant:
- ◊ Default value is decimal constant:
- ◊ Default value is integer constant:
- ◊ Default value is hexadecimal string:
- ◊ Default value is UNICODE string:
- ◊ Default value is character string:
- ◊ Default value is graphic string:
- ◊ Default value is decimal floating point constant:

Where the default value description ends with ":" (colon), the following line displays the defined default value in quotation marks.

Indexes on Column

If the column is part of a defined index key, a table follows that contains one row for each index defined on the column. The information displayed for each index is as follows:

Index Name
The name of the index for which the column constitutes part of the index key.

Index Type
The type of index. Possible values are: **Primary**, **Unique**, **Non-unique**, **Unique constraint**, **Unique not null**, **Unique parent**, **Unique ROWID** and **Unique XML values**.

Position
Position (sequence number) of the column and number of columns within the index key.

Ordering
Ordering of column values in the index key. (Ascending or Descending)

Foreign Keys Including Column

If the column is part of a foreign key defined by a referential constraint on the DB2 table, a table follows that contains one row for each index defined on the column. The information displayed for each index is as follows:

Parent Table
The schema and name of the parent key table.

Constraint Name
The name of the referential constraint definition that identifies the column as being part of the forign key.

Position
Position (sequence number) of the column and number of columns within the foreign key.

Delete Rule
The delete rule defined for this referential constraint. Possible values are: **No action**, **Cascade**, **Set null** and **Restrict**.

The delete rule determines the action to be taken when a row of the parent table is deleted which breaks the referential constraint enforced on a dependent table.

Enforced by DB2
Indicates whether or not the referential constraint is enforced by DB2 during normal operations (such as insert, update, or delete) and that it is guaranteed to be correct. (Yes or No)

Table Indexes

If one or more indexes have been defined on the table, this section displays a table containing one row for each defined index. The information displayed for each column is as follows:

Index Name
The schema and name of the index.

Index Type
The type of index. Possible values are: **Primary**, **Unique**, **Non-unique**, **Unique constraint**, **Unique not null**, **Unique parent**, **Unique ROWID** and **Unique XML values**.

Cols
Number of table columns that constitute the index key.

Seq
For each index key column belonging to the index entry, this field displays the sequence number of the column within the index key.

Column Name
For each index key column belonging to the index entry, this field displays the name of the column within the index key.

Ord
For each index key column belonging to the index entry, this field displays the ordering sequence of values in the column within the index key. ("A" - ascending or "D" - descending)

Parent Tables

If one or more referential constraints have been defined on the table, this section displays a table containing one row for each defined referential constraint. The information displayed for each column is as follows:

Parent Table
Parent key table and other options specified when the referential constraint was defined. The parent table entry contains the following report fields:

Table Name:
Schema and name of the parent key table.

Contraint:
Name of the referential constraint.

Delete Rule:
The delete rule defined for this referential constraint. Possible values are: **No action**, **Cascade**, **Set null** and **Restrict**.

Enforced:
Indicates whether or not the referential constraint is enforced by DB2 during normal operations. (Yes or No)

Foreign Key of Parent
This column is split into 3 sub-columns which identify the sequence number, name and parent column name of each column that constitutes a foreign key defintion in this table.

Seq
For each foreign/parent key column relationship belonging to the referential constraint entry, this field displays the sequence number of the column within the foreign/parent key.

Column Name
For each foreign/parent key column relationship belonging to the referential constraint entry, this field displays the name of the foreign key column in this table.

Parent Column Name
For each foreign/parent key column relationship belonging to the referential constraint entry, this field displays the name of the related parent key column in the parent table.

Dependent Child Tables

If one or more referential constraints exist on other DB2 tables that nominate this table as the parent key table, then this section displays a table containing one row for each defined referential constraint. The information displayed for each column is as follows:

Child Table
Child (dependent) table and other options specified when the referential constraint was defined. The child table entry contains the following report fields:

Table Name:
Schema and name of the child (foreign key) table.

Contraint:
Name of the referential constraint.

Delete Rule:
The delete rule defined for this referential constraint. Possible values are: **No action**, **Cascade**, **Set null** and **Restrict**.

Enforced:
Indicates whether or not the referential constraint is enforced by DB2 during normal operations. (Yes or No)

Foreign Key in Child
This column is split into 3 sub-columns which identify the sequence number, parent key column name and child (foreign key) column name of each column that constitutes a foreign key defintion in the child table.

Seq
For each foreign/parent key column relationship belonging to the referential constraint entry, this field displays the sequence number of the column within the foreign/parent key.

Column Name
For each foreign/parent key column relationship belonging to the referential constraint entry, this field displays the name of the parent key column in this table.

Child Column Name
For each foreign/parent key column relationship belonging to the referential constraint entry, this field displays the name of the related foreign key column in the child (dependent) table.

Create Structure (SDO) Menu (=9)

Create Structure Menu Panel

The Create Structure (SDO) Menu (ZZSGSDO0) is an interactive panel window, opened on selection of option 9. in the SELCOPY/i Primary option menu.

SELCOPY/i supports browse, edit, compare, search, update and remap of structured data records. To accurately format these records, the SELCOPY/i structured data editor uses an internally defined structure (SDO) which may be pre-generated from any combination of the following sources:

1. COBOL copybook.
2. PL1 copybook (include file).
3. COBOL or PL1 ADATA output file.
4. SELCOPY/i SDO structure file.
5. SDE CREATE STRUCTURE Record Type Definition syntax.

Additionally, the ZZSXREF utility may be used to generate a SELCOPY/i SDO from an XREF file.

This panel allows the user to select the source from which the Structured Data Edit (SDE) structure object (SDO) is to be created. Enter the relevant option number or position the cursor on the required option and press <Enter> or, if configured, double-click the left mouse button.

Menu Bar Items

File
The File drop-down menu contains the single item, Exit, to close the panel.

Help
Open the general help for the Create Structure (SDO) Menu option menu panel.

Options

1 Copybook	**SDO**	Create a structure from COBOL or PL1 copybook(s)
2 XREF	**XREF**	Create a structure from an existing XREF file
3 Layout	**LAYOUT**	Display record layouts defined by an SDO/copybook
4 Associations	**STRUCT**	Manage data file/copybook mapping associations
5 Loaded	**LLS**	List loaded structures
6 DB2		Create a DB2 structure (inc SELECT/WHERE/ORDER)
7 DCL		Create DCL/EQU for SELCOPY/SLC batch programs

Create Structure from Copybook(s)

The Create Structure from Assembler, COBOL or PL1 copybooks panels assist with generation of a SELCOPYi structure definition file using one or more Assembler DSECT, COBOL GROUP and/or PL1 STRUCTURE definitions. Each group/structure generates a single record type (RTO) mapping within the SDO.

Create Structure from COBOL/PL1 copybook(s) Panel

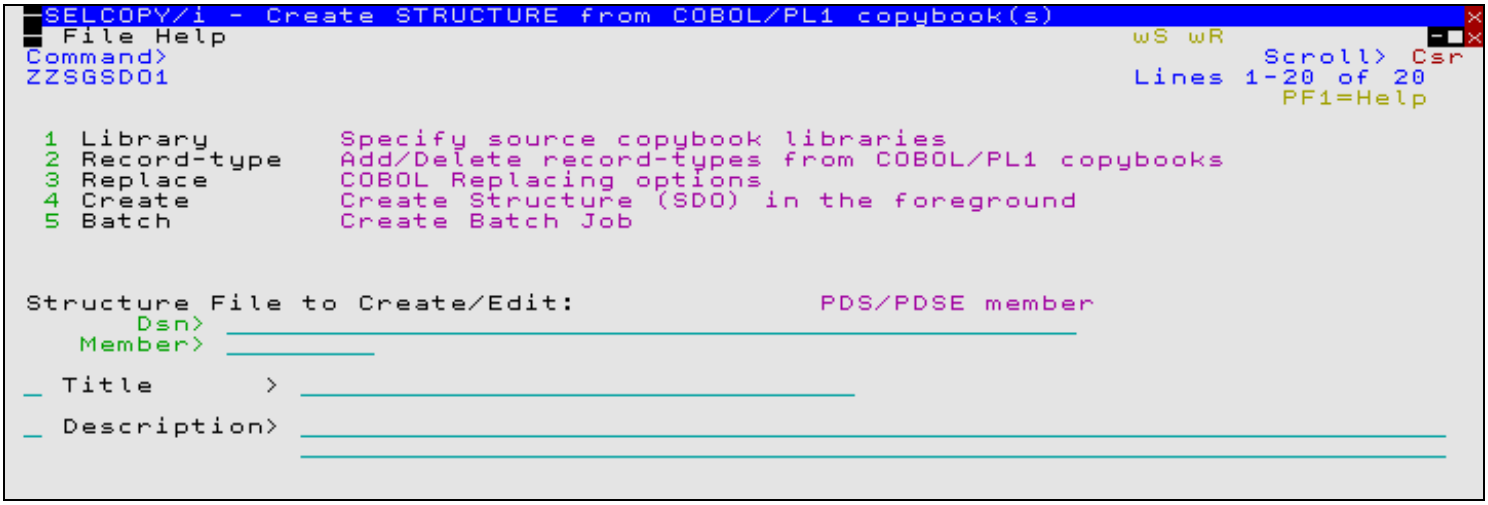


Figure 167. SELCOPYi - Create Structure from COBOL/PL1 copybook(s).

The Create Structure from COBOL/PL1 copybook(s) panel (ZZSGSDO1) is an interactive panel window, opened on selection of option 1. in the Create Structure (SDO) Menu.

Optional field entries must be activated by entering "/" in the preceding field if their values are to be included.

Options

1 Library	Specify source copybook libraries
2 Record-type	Add/Delete record-types from COBOL/PL1 copybooks
3 Replace	COBOL Replacing options
4 Create	Create Structure (SDO) in the foreground
5 Batch	Create Batch Job

Create

Create the SDO structure file in the SELCOPYi foreground using the current field values.

Batch

Generate a JCL job stream that executes the **SDEAMAIN** program with input (SDEIN) containing the SDE CREATE STRUCTURE command generated for the specified panel field values.

The job stream is displayed in a temporary text edit view and may be submitted to batch using the SUBMIT command.

Panel Input Fields

Structure File to Create/Edit:

Fields that identify the SELCOPYi structure (SDO) data set or PDS/PDSE member name. If the specified data set or PDS/PDSE member is an existing SDO which was generated **using this panel**, then panel options and field values will be re-populated using equivalent fields from the SDO.

Dsn>

Fully qualified name of a sequential data set or PDS/PDSE library. Quotes are permitted but unnecessary.
A selectable list of data sets will be presented if wildcard character "*" occurs anywhere within the specified DSN.

Member>

If DSN is a PDS/PDSE library, specifies the SDO member name.
A selectable list of members will be presented if wildcard character "*" occurs anywhere within the specified member name or the member name is left blank.

This parameter field corresponds to CREATE STRUCTURE parameter *struct-name*.

Title>
If activated, specifies an up to 30 character title for the SDO.
This parameter field corresponds to CREATE STRUCTURE parameter TITLE *sdo_title*.

Description>
If activated, specifies an up to 124 character description of the SDO.
This parameter field corresponds to CREATE STRUCTURE parameter DESCRIPTION *sdo_description*.

Create Structure - Copybook Library List

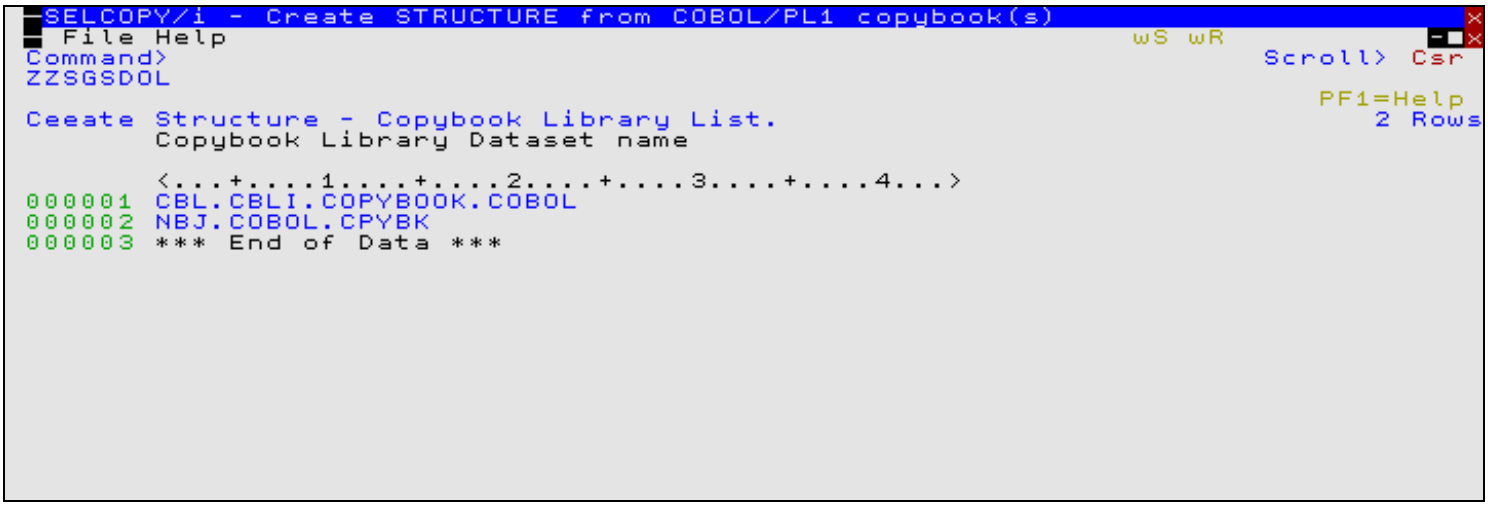


Figure 168. SELCOPYi - Create Structure - Copybook Library List.

The Create Structure - Copybook Library List panel (ZZSGSDOL) is an interactive panel window, opened on selection of option 1. in the Create Structure from COBOL/PL1 copybook(s) panel.

Standard SELCOPYi table editing techniques should be used to add a table row entry for each required copybook source library DSN.

The table identifies library data sets and the order in which they are to be searched. The library search chain is used to locate each copybook source member specified in the Create Structure - Define Record Types panel.

Pressing <PF3> to exit the panel, will also save the table of libraries and return to the Create Structure from COBOL/PL1 copybook(s) panel.

These library names corresponds to CREATE STRUCTURE parameter LIBRARY(*search_lib* ...).

Create Structure - Define Record-Types

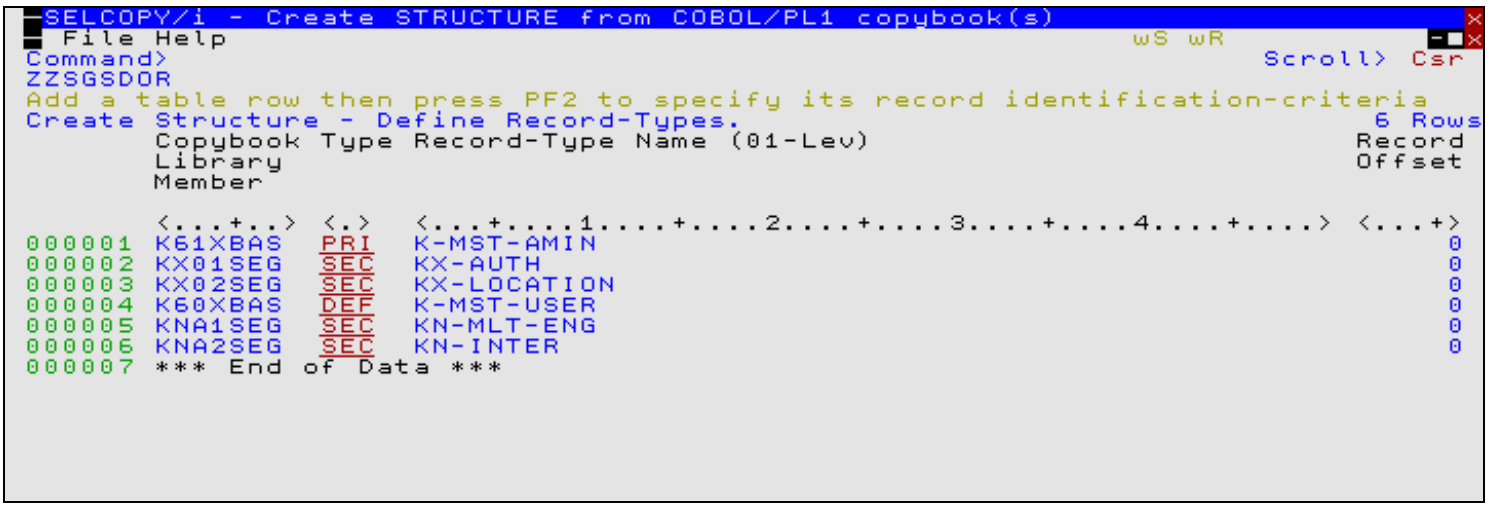


Figure 169. SELCOPYi - Create Structure - Define Record-Types.

The Create Structure - Define Record-Types panel (ZZSGSDOR) is an interactive panel window, opened on selection of option 2. in the Create Structure from COBOL/PL1 copybook(s) panel.

This panel is used to specify the source member and record type name for each record type definition (RTO) in the generated SDO.

Standard SELCOPYi table editing techniques should be used to add a table row entry for each required record type definition.

The specified record-type name must be the name of an Assembler DSECT, COBOL GROUP field or PL1 STRUCTURE defined in the source copybook member. If the copybook member contains more than one group/structure field, each required to define a separate RTO, then the copybook member may be specified in more than one table row.

A record type definition must be defined as being either a primary segment (PRI), mapping an entire logical record or the first segment of a segmented logical record, or a secondary segment, mapping second and/or subsequent segments of a segmented logical record. At least one default (DEF) primary segment record type must also be defined.

Having inserted a table row, the user can display the single record view of the row using the ZOOM command (assigned to <PF2> by default.) The zoom view is required in order to identify the source member as being a PL1 copybook or ADATA output and also to to specify record type identification (USE WHEN) criteria.

Pressing <PF3> to exit the zoomed view of the panel, will update the record-type definition table row and return to the multi-record view of the table. Pressing <PF3> again to exit the Define Record-Types panel, will save the table of record type definitions and return to the Create Structure from COBOL/PL1 copybook(s) panel.

Fields in this table corresponds to CREATE STRUCTURE parameter RECORD (**Record Type Def**).

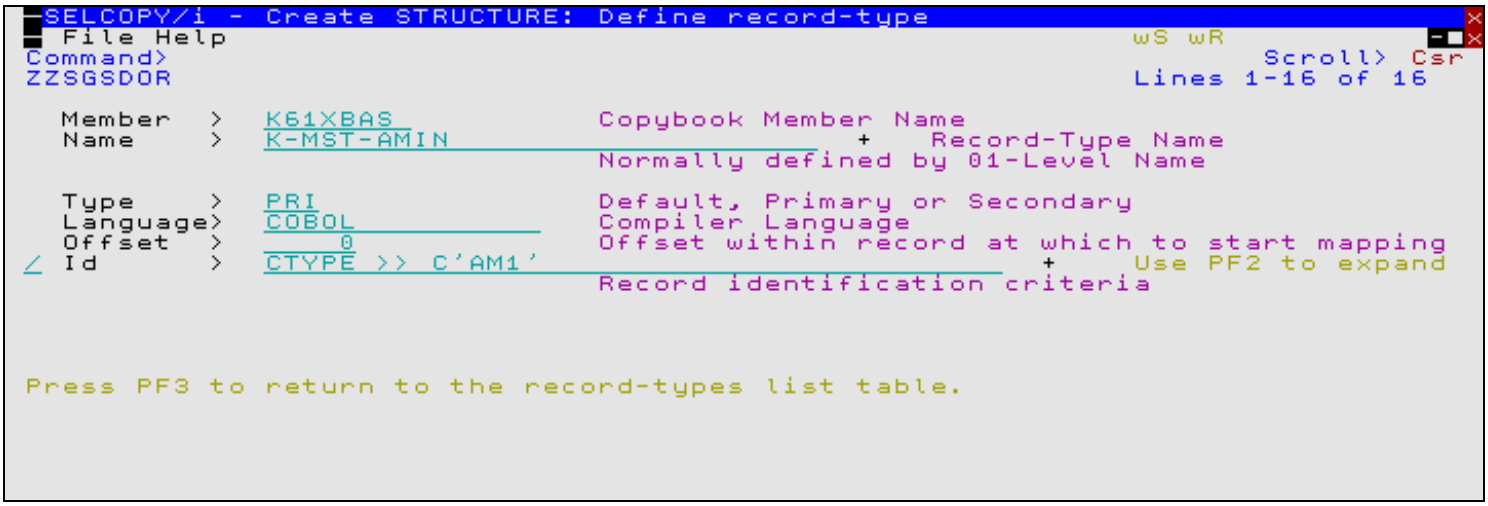


Figure 170. SELCOPYi - Create Structure - Define Record-Types Zoomed View.

Panel Input Fields

Member>
Name of an Assembler, COBOL or PL1, copybook or ADATA member belonging to a library in the defined library search chain. Note that, if the member name exists in more than one library in the search chain, then the first occurrence will be used.

This field corresponds to CREATE STRUCTURE parameter SOURCE *member_name*.

Name>
Name of the record type to be generated in the SDO. The specified name must match the name of an Assembler DSECT, COBOL GROUP or PL1 STRUCTURE defined in the source member.

This field corresponds to CREATE STRUCTURE parameter NAME *record_type*.

Type>
Specifies the type (**DEF**, **PRI** or **SEC**) of the record-type to be defined.

PRI indicates that the record type maps an entire logical record or the first (primary) segment of a segmented logical record.
SEC indicates that the record type definition maps second and/or subsequent (secondary) segments of a segmented logical record.
DEF indicates that the record type is a default primary record type definition. One and only one DEF record type must be defined in the SDO.

This field corresponds to CREATE STRUCTURE parameters PRIMARY, SECONDARY and DEFAULT.

Language>
Specifies the format of the source mamber as an Assembler (**ASM**) DSECT, **COBOL** copybook, **PL1** include file or **ADATA** (Assembler, COBOL or PL1) output file.

Note that it is possible to combine Assembler, COBOL, PL1 and ADATA source members in the same SDO generation.

This field corresponds to CREATE STRUCTURE parameter SOURCE ASM|COBOL|PL1|ADATA.

Offset>

Optional positive (or negative) numeric value specifying the offset into (or before) the record/segment data at which the record type mapping will begin.

This field corresponds to CREATE STRUCTURE parameter OFFSET <+|-> *n_bytes*.

Id>

If activated, specifies arguments to a USE IF expression to be saved in the record type definition. The USE IF expression identifies the criteria, based on record data, for which this record type will be applied.

For non-segmented records or primary record segments (type DEF or PRI), the USE WHEN condition references data within the current record or segment only. However, a secondary segment (SEC) condition may also reference data in the primary segment or previous secondary segments.

Furthermore, for all segment types (DEF, PRI and SEC), the record data may be referenced using unformatted record positions or formatted field names. See information on SDE expressions with particular reference to field value terms and built-in functions.

When the selection criteria involves testing fields in unformatted, non-segmented record data, the SDE expression should simply use function SUBSTR() on the record field RECORD. e.g. To assign a record type based on a character string "A11" at position 11 of the record data, specify the following:

```
USE IF  SUBSTR(RECORD,11,3) = 'A11'
```

To test fields in the record data as if already formatted by the record type structure you are attempting to assign, simply reference the required field names in the SDE expression. e.g. To assign a record type based on values in 2 numeric fields, SEQUENCE_ID and CUST_REF, both defined within the record type structure...

```
USE IF  ( SEQUENCE_ID > 301   AND   CUST_REF = 10233 )
```

Note that fields SEQUENCE_ID and CUST_REF may have been defined as packed decimal, integer, floating point, etc. If the record data is invalid for the assigned data type, the expression will return a "false" condition.

This field corresponds to CREATE STRUCTURE parameter USE WHEN *expression*.

Create Structure from XREF File

Create Structure from XREF File Panel

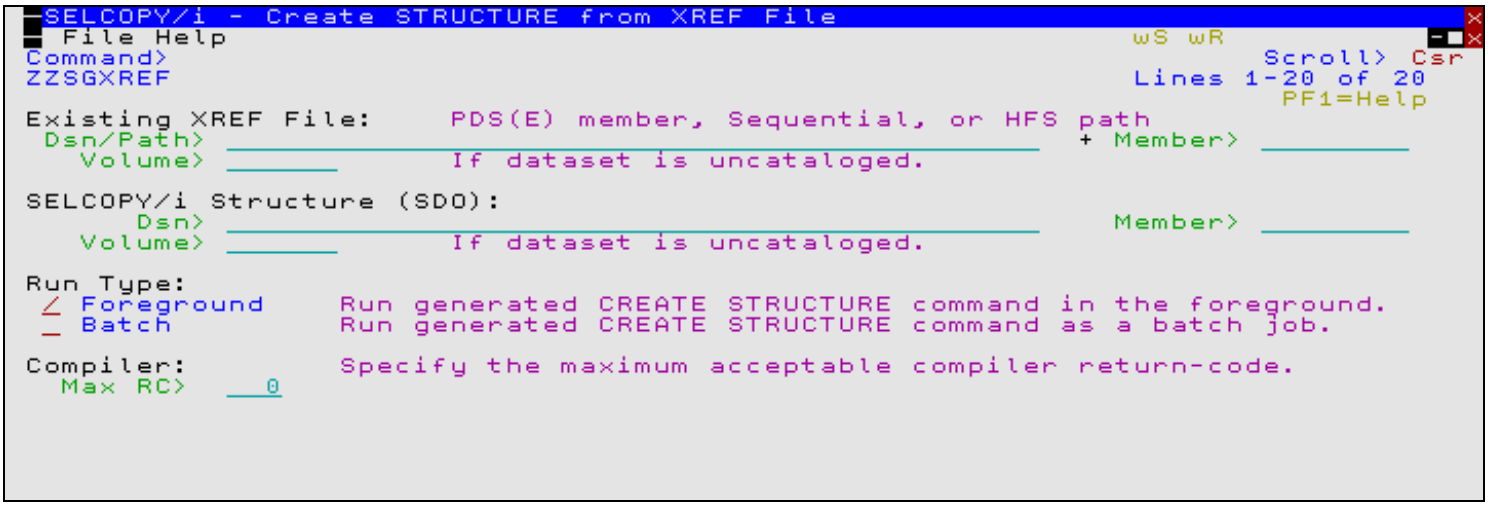


Figure 171. SELCOPYi - Create Structure from XREF File.

The Create Structure from XREF File panel (ZZSGXREF) is an interactive panel window, opened on selection of option 2. in the Create Structure (SDO) Menu. This panel provides an interface to the ZZSXREF utility to convert XREF files to SELCOPYi structure definition file (SDO).

Panel Input Fields

Existing XREF File:
Fields that identify the source XREF data set, HFS file or PDS/PDSE member.

Dsn/Path>
An absolute or relative HFS Path name or the fully qualified name of a sequential data set or PDS/PDSE library. Quotes are permitted but unnecessary.

A selectable list of data set names or HFS files will be displayed as appropriate if either wild card character "%" (percent) or "*" (asterisk), both representing zero or more characters, is specified. If a volume id exists in the Volume field, then a list of selectable data sets will be restricted to those contained in that volume's VTOC.

Member>
If DSN is a PDS/PDSE library, specifies the XREF member name.

A selectable list of members will be presented if wildcard character "*" or "%" occurs anywhere within the specified member name or the member name is left blank.

Volume>
Specifies a volume serial id mask for an uncataloged XREF file. (Not applicable to HFS files.)

SELCOPYi Structure (SDO):
Fields that identify the SELCOPYi structure (SDO) data set or PDS/PDSE member to be generated.

Beware that selecting an existing sequential data set or member will overwrite all existing data in that file.

Dsn>
Fully qualified name of a sequential data set or PDS/PDSE library. Quotes are permitted but unnecessary.

A selectable list of data sets will be displayed if either wild card character "%" (percent) or "*" (asterisk), both representing zero or more characters, is specified.

Member>
If DSN is an existing PDS/PDSE library, this field specifies the SDO member name.

A selectable list of members will be presented if wildcard character "*" or "%" occurs anywhere within the specified member name or the member name is left blank.

Volume>
Specifies a volume serial id mask for an uncataloged SDO.

Run Type>

Specifies whether the conversion utility is to be executed in the SELCOPYi **Foreground** or will generate a JCL **Batch** job as a temporary file displayed in a CBLe text edit view.

The batch job runs SDEAMAIN to execute CREATE STRUCTURE syntax and may be submitted to batch using the SUBMIT command.

Compiler:

SELCOPYi requires a COBOL or PL1 compilation ADATA output file in order to generate an SDO structure file. Therefore, the COBOL or PL1 compiler is invoked to compile all copybook files specified in the XREF file.

Max RC>

Specifies the maximum acceptable return code that may be returned by the COBOL or PL1 compilers in order for SELCOPYi to continue generating the SDO from the resulting ADATA file.

Display Record Layout

Display Record Layout Panel

The Display Record Layout panel (ZZSGLAYO) is an interactive panel window, opened on selection of option 3. in the Create Structure (SDO) Menu. This panel provides an interface to the SDE LAYOUT command used to display the layout of all records within a record structure (SDO, COBOL/PL1 Copybook or COBOL/PL1 ADATA) file.

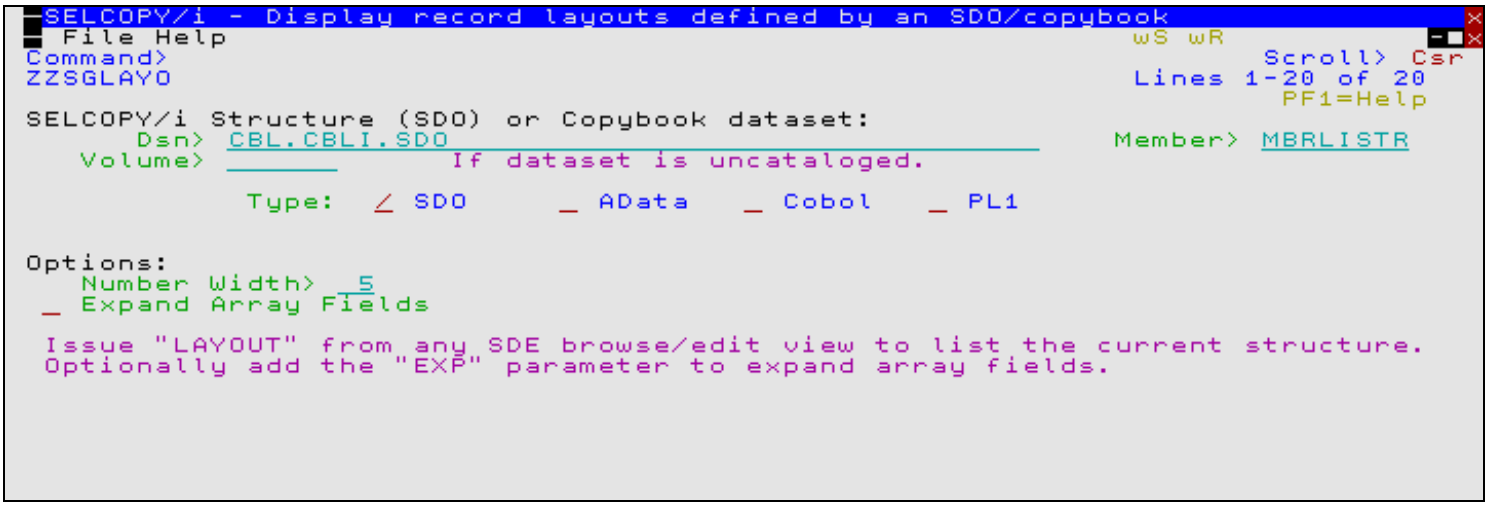


Figure 172. SELCOPYi - Display Record Layout Panel.

Panel Input Fields

SELCOPYi Structure (SDO) or Copybook dataset:
Fields that identify the source structure (SDO, Copybook or ADATA) data set or PDS/PDSE member that contain one or more record structure definitions.

Dsn>
Fully qualified name of a sequential data set or PDS/PDSE library. Quotes are permitted but unnecessary.

A selectable list of data sets will be displayed if either wild card character "%" (percent) or "*" (asterisk), both representing zero or more characters, is specified. If a volume id exists in the Volume field, then a list of selectable data sets will be restricted to those contained in that volume's VTOC.

Member>
If DSN is an existing PDS/PDSE library, this field specifies the member name.

A selectable list of members will be presented if wildcard character "*" or "%" occurs anywhere within the specified member name or the member name is left blank.

Volume>
Specifies a volume serial id mask for an uncataloged data set.

Type:
Specifies the format (SDO, **COBOL** or **PL1** Copybook, COBOL or PL1 **ADATA** output) of the record structure definition file.

Number Width>
Specifies the displayed width of numeric columns: **RefNo**, **Start**, **End** and **Length** in the layout list output.

The minimum width of these field is 5 characters and so this value need only be increased if it is known that the decimal display of these field values exceeds this width.

Expand Array Fields
This option field determines whether or not array (OCCURS) fields are expanded to display every repeating instance of a field within that array.

Display Record Layout Output

The structure layout is displayed in a list window.

For each field, the nested level number of that field is displayed to the left of the field name in the **Name** column. Furthermore, indentation occurs within the **Name** column for each nested level within a Group field.

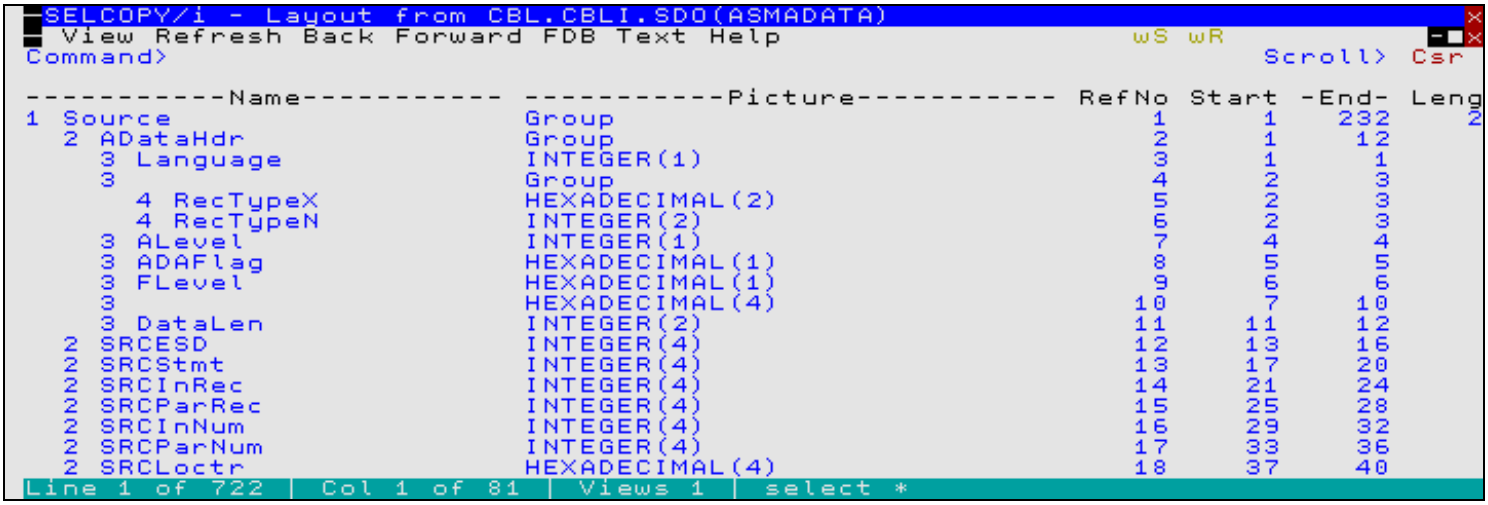


Figure 173. SELCOPYi - Display Record Layout Output.

Columns Displayed

Name	Type	Description
Name	ALPair	Field level and name
Picture	ALPair	Field picture and data type
RefNo	Int	Field reference number
Start	Int	Field start position
End	Int	Field end position
Length	Int	Field length

File to Copybook Associations (=9.4)

The Data File to Copybook Associations table edit view is a Data Editor edit view of member ZZSDSUSE in the user's SELCOPYi table library (TLIB). It may be opened via the following:

1. Select option 4. "Assiciations" in the Create Structure (SDO) Menu. (=9.4)
2. Enter primary command **EDIT** at the Manage Copybook Associations settings panel (=0.4.6).

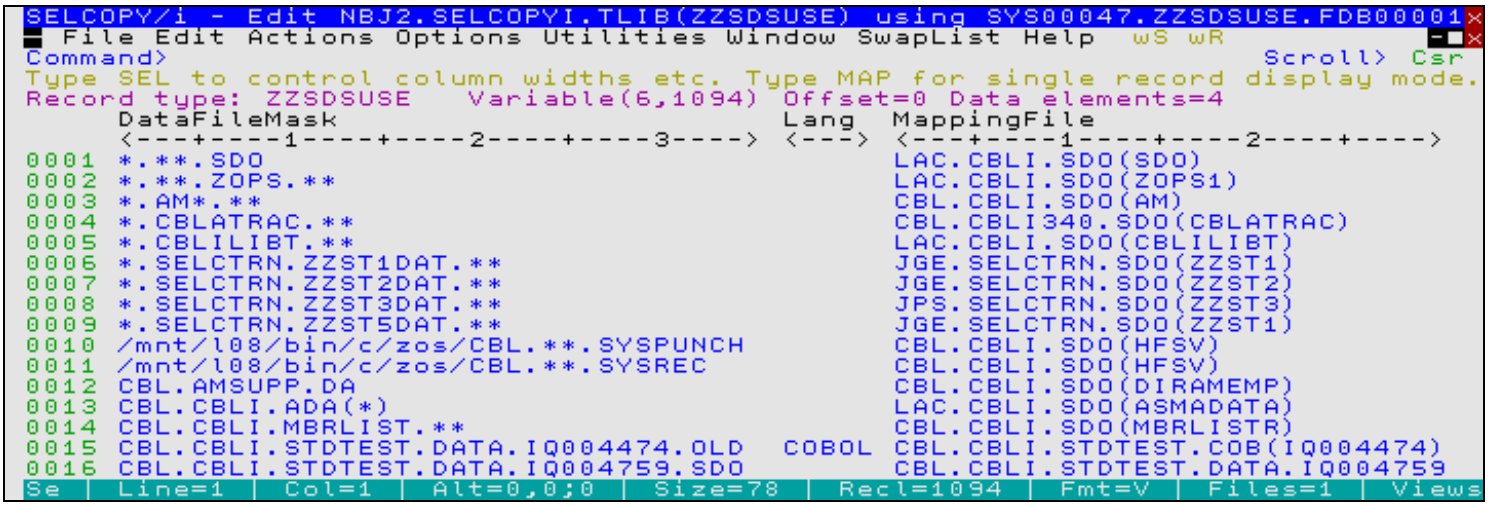


Figure 174. SELCOPYi - Data File/Copybook Mapping Associations.

Each entry in the Data File to Copybook Associations table edit view contains a data fileid mask together with the name of an associated SELCOPYi structure, COBOL or PL1 Copy book or HLASM DSECT that maps record data within files whose names match the fileid mask.

When the Data Editor is opened to browse or edit a file without specifying an associated structure, entries in this table are automatically scanned to find the first fileid mask that matches for the file's name. If successful, the associated structure name is then used to map the file's data. If no match is found, the file is displayed without an associated structure.

Table entries may be inserted, re-ordered (moved), deleted and updated using standard Data Editor primary and line commands.

Generic associations may be defined by speciying standard dataset/member name wildcard characters "*" (asterisk) or "%" (percent) in the DataFileMask column entry.

List Loaded Structures (=9.5)

List Tables Panel

The List Loaded Structures panel (ZZSGLLS0) is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

1. Select option 5. "Loaded" in the Create Structure (SDO) Menu. (=9.5)
2. Enter primary command LLS at any command line.

This panel may be used to list the file mapping structures (SDOs) that SELCOPYi currently has loaded into memory. These typically include structures used to map files, DB2 tables, output from the File-Search or Compare utilities etc that are open in other SELCOPYi windows.
In other words, structures that are currently in use.

It will also include any temporary structures that are created when direct reference to a COBOL/PL1 copybook is made on Data-Edit/Browse. These temporary structures are deliberately kept in memory for the duration of the SELCOPYi session (to save recompilation from the original source), but may be dropped using this panel if memory is getting tight, or if a a recompile is desired following an update to the original source.

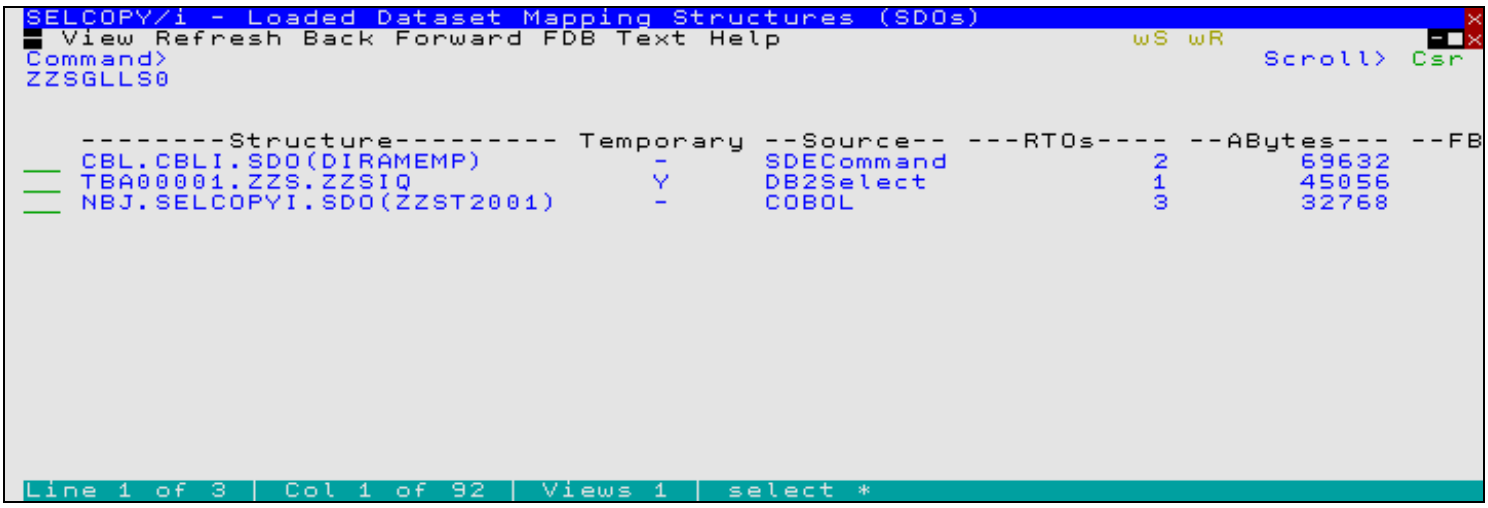


Figure 175. SELCOPYi - Loaded Dataset Mapping Structures (SDOs).

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command L.
D	DROP the structure from memory.
L	Issue the LAYOUT command for this structure.
LS	Issue the LS command for this structure.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 by default.

Columns Displayed

Name	Type	Description
Structure	VChar	Structure name
Temporary	BitFlag	Temporary structure
Source	Enum	Source
RTOs	Int	Number of record types
ABytes	Int	Allocated bytes
FBytes	Int	Free bytes
Flag1	Hex	Flag1

Generate SELCOPY DCL/EQU Statements (=9.7)

The Generate SELCOPY DCL/EQU Statements facility provides a method of generating SELCOPY DCL or EQU operations from a SELCOPYi SDO structure or from Assembler DSECT, COBOL GROUP and/or PL1 STRUCTURE definitions.

Generate DCL/EQU Statements from a Copybook Mapping Panel

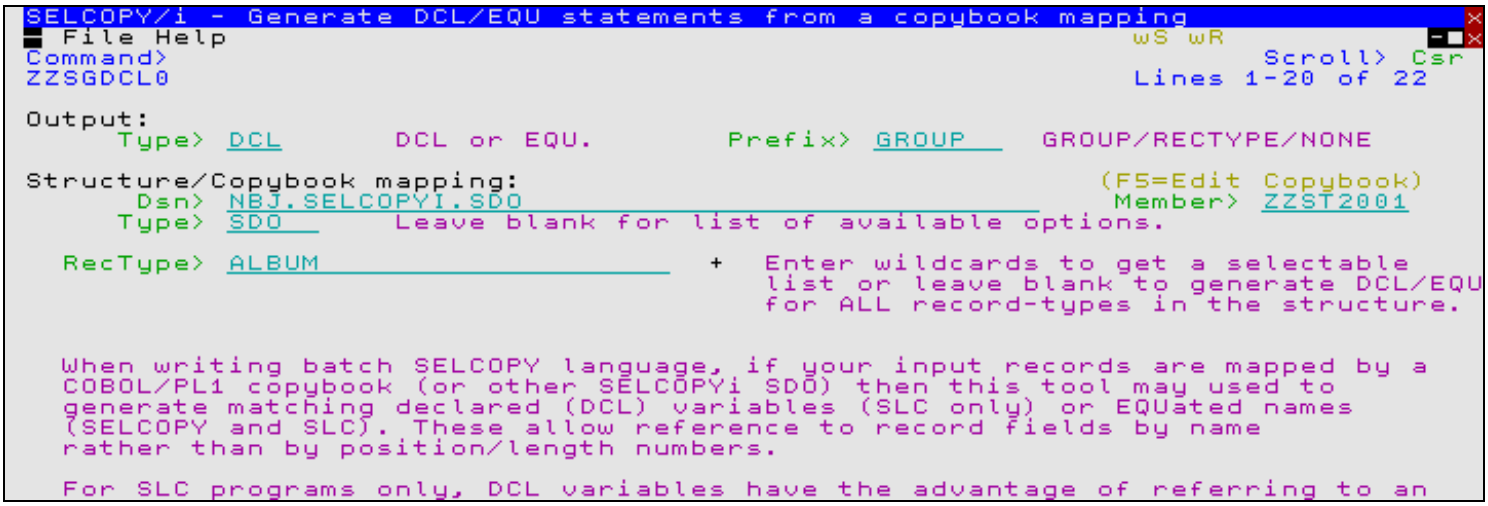


Figure 176. SELCOPYi - Generate SELCOPY DCL/EQU Statements.

The Generate DCL/EQU Statements panel (ZZSGDCL0) is an interactive panel window, opened on selection of option 7. in the Create Structure (SDO) Menu.

Panel Input Fields

`Output:`
Fields that ...

`Structure/Copybook mapping:`
Fields that identify the SELCOPYi structure (SDO) data set or PDS/PDSE member name. If the specified data set or PDS/PDSE member is an existing SDO which was generated **using this panel**, then panel options and field values will be re-populated using equivalent fields from the SDO.

`Dsn>`
Fully qualified name of a sequential data set or PDS/PDSE library. Quotes are permitted but unnecessary.
A selectable list of data sets will be presented if wildcard character "*" occurs anywhere within the specified DSN.

`Member>`
If DSN is a PDS/PDSE library, specifies the SDO member name.
A selectable list of members will be presented if wildcard character "*" occurs anywhere within the specified member name or the member name is left blank.

`Type>`

Create File Filter (=10)

Create File Filter Panel

SDE structured data BROWSE and update in-place EDIT supports use of a filter so that only records that satisfy criteria defined by the filter clause are eligible for display in the SDE window view.

A filter clause may be passed to the SDE EDIT or BROWSE operation via in-line command syntax or via a filter file. A filter file contains the filter syntax that would be specified in-line but has the advantage that it may be kept for use in other EDIT and BROWSE operations.

The Create File Filter panel assists with generation of a filter clause which is ultimately saved to a filter file. The filter file may then be specified on the FILTER: field of the **SDE - Structured Data Browse/Edit** panel.

The Create File Filter Dialog panel (ZZSGFLT0) is an interactive panel window and may be started via the following:

- Select option 10. in the SELCOPYi Primary option menu.
- Select 'Create File Filter' from the Utilities menu.
- Select menu item **FILTER** from the SDE - Structured Data Browse/Edit panel.
- Execute the command **FILTERGEN** with no parameters from the command line of any window.

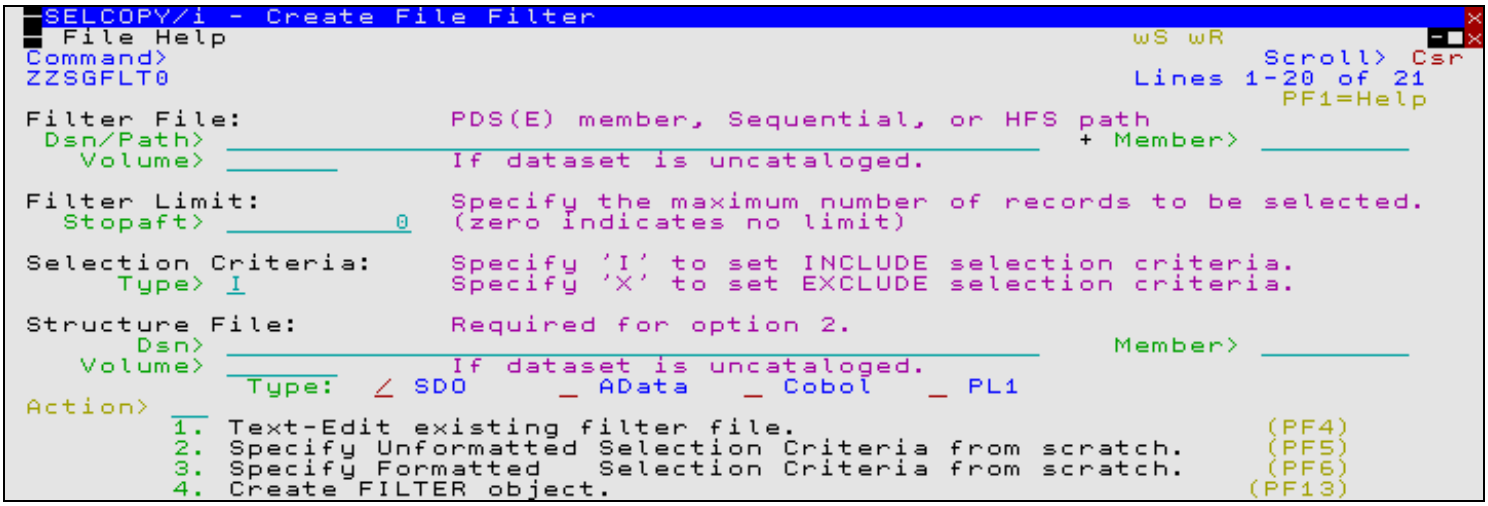


Figure 177. SELCOPYi - Create File Filter (=10).

Panel Input Fields

By default, field entries are populated with arguments and options that were entered the last time the panel was used.

Filter File:
Fields that together identify the PDS/PDSE library member, sequential data set or HFS file path to be edited, using the CBLe text editor, and in which a generated filter clause will be copied (replacing any existing data).

This file is not saved to disk before it is displayed, however, the user will be prompted to save it on exit (PF3).

Dsn/Path>
An absolute or relative HFS Path name or the fully qualified name of a sequential data set or PDS/PDSE library. Quotes are permitted but unnecessary.

A selectable list of data set names or HFS files will be displayed as appropriate if either wild card character "%" (percent) or "*" (asterisk), both representing zero or more characters, is specified. If a volume id exists in the Volume field, then a list of selectable data sets will be restricted to those contained in that volume's VTOC.

Member>
If DSN is a PDS/PDSE library, specifies the FILTER file member name.

A selectable list of members will be presented if wildcard character "*" or "%" occurs anywhere within the specified member name or the member name is left blank.

Volume>
Specifies a volume serial id mask for an uncataloged FILTER file. (Not applicable to HFS files.)

Filter Limit:
The filter limit defines when to stop filtering input records.

Stopaft>
Specifies the maximum number of records to be selected by this filter. If this threshold is reached, then:

- For an INCLUDE filter, all remaining untested records are excluded.
- For an EXCLUDE filter, all remaining untested records are included.

A value of 0 (zero) removes this filter limit.

Selection Criteria:
Determines whether filtering is to include or exclude records that match the filter clause record selection criteria.

Type>
Specifies either **I** (INCLUDE) or **X** (EXCLUDE) to indicate the action performed on records that selected by the filter clause.

Structure File:
Fields that identify the sequential data set name or PDS/PDSE member name of a file containing record structure definitions to be used to select formatted field names in the filter clause WHERE expression. (See Action 3. "*Specify Formatted Selection Criteria*".)

Dsn>
Fully qualified name of a sequential data set or PDS/PDSE library. Quotes are permitted but unnecessary.

A selectable list of data sets will be displayed if either wild card character "%" (percent) or "*" (asterisk), both representing zero or more characters, is specified.

Member>
If DSN is an existing PDS/PDSE library, this field specifies the member name.

A selectable list of members will be presented if wildcard character "*" or "%" occurs anywhere within the specified member name or the member name is left blank.

Volume>
Specifies a volume serial id mask for an uncataloged data set.

Type:
Specifies the format (SDO, **COBOL** or **PL1** Copybook, COBOL or PL1 **ADATA** output) of the record structure definition file.

Action>
Option field specifying the action to be taken when the <Enter> key is pressed.

1. **Text-Edit existing filter file.**
   Opens a CBLe text edit view to display and allow edit of a filter clause specified in an existing file identified by Filter File.

   No new filter is generated as a result of selecting action 1. so the filter expresions that already exist in the file are preserved.

2. **Specify Unformatted Selection Criteria.**
   Opens the Filter (unformatted) - Selection Criteria panel to build filter clause record selection criteria based on unformatted record data.

3. **Specify Formatted Selection Criteria.**
   Opens the Filter (formatted) - Selection Criteria panel to build filter clause record selection criteria based on record data which has been formatted using a Structure File.

   This panel will first display a table view of all record structures defined within the structure file. Select the record structures that are to be included or excluded.

   For each of the selected record structures, further record selection criteria may then be specified based on fields defined within the record structures.

4. **Create FILTER object.**
   Opens a CBLe text edit view to display the generated filter clause in the file identified by Filter File.

   When the edit window is closed (<PF3>), the user will be prompted to save the file and, if necessary, open the Allocate NonVSAM dialog window.

   In order to prevent accidental overwrite of data, if the file already exists, then the save will be rejected with message ZZSE046E. To force the save regardless of the warning, issue command **SSave,** (or **FFile** to save and exit at once).

Unformatted Selection Criteria

Unformatted Selection Criteria Panel Table View

The FILTER (unformatted) - Selection Criteria panel (ZZSGFLTR) is an interactive panel window, opened on selecting Action 2. in the Create File Filter panel.

The panel contains a table where each row represents a sub-expression of the single INCLUDE or EXCLUDE sub-clause (WHERE expression). Each generated sub-expression involves a SUBSTR() built-in function which operates on the unformatted record data referenced by "record".

The sub-expressions are separated by logical operator AND or OR, where AND is higher in the order of operator precedence than OR. i.e. A AND B OR C is equivalent to (A AND B) OR C.

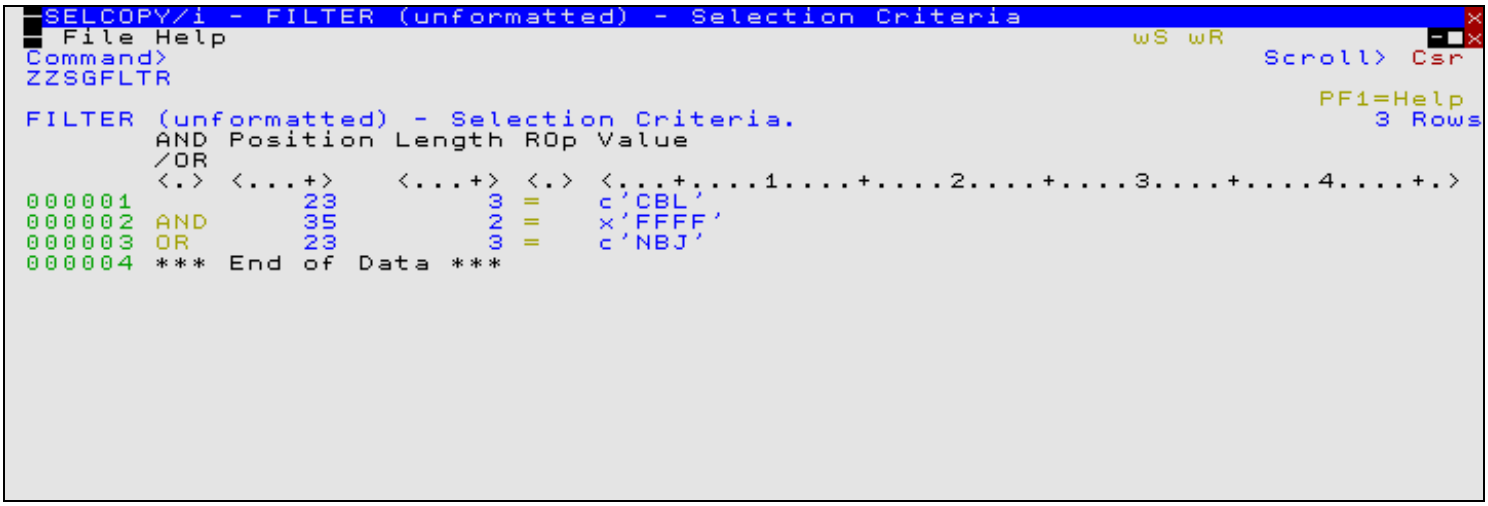


Figure 178. SELCOPYi - Unformatted Selection Criteria.

Standard SELCOPYi table editing techniques should be used to add a table row entry for each filter sub-expression.

Each table row consists of the logical operator (AND/OR) position and length of the data within the record to be tested, the relational operator (ROp) and the value to be tested. Enter null or invalid enties in the **AND**/**OR** and **ROp** fields to display and select permitted entries for these fields.

Since logical operators AND and OR are dyadic, the **AND**/**OR** field of the first table row (the first term of the expression) is always blank and cannot be updated.

Unformatted Selection Criteria Panel Single View

Having inserted a table row, the user can display the single record view of the row using the ZOOM command (assigned to <PF2> by default.)

Pressing <PF3> to exit the zoomed view of the panel, will update the filter sub-expression table row and return to the multi-record view of the table. Pressing <PF3> again to exit the Selection Criteria panel, will save the table sub-expressions and return to the Create File Filter panel.

Fields in this table corresponds to the filter WHERE *expression*.

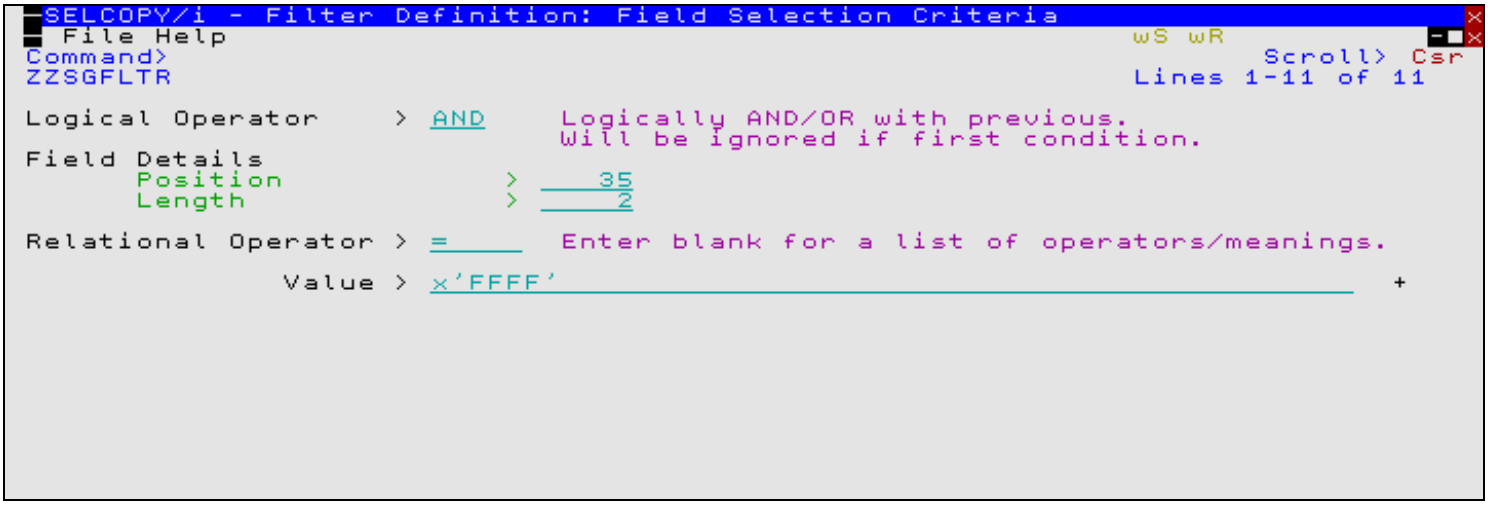


Figure 179. SELCOPYi - Unformatted Selection Criteria Zoomed View.

Panel Input Fields

By default, field entries are populated with arguments and options entered in the table view for that entry.

Logical Operator>
Logical operator that identifies the relationship of this field test sub-expression with the previously specified sub-expression. If this is the first, then this field is ignored. Permissible operators are AND or OR.

Field Details:
Fields that together identify the location within the input records of the data field to be tested.

Position>
Integer numeric identifying the position of the first byte of the test field within the record data.

Length>
Integer numeric identifying the length of the test field within the record data.
A length of 0 (zero) indicates a length from the start position to the end of the record.

Relational Operator>
Identifies the type of test to be performed on the field data. Enter blank in this field to display a selectable list of supported relational operator symbols and a brief description of each.

Value>
Specifies the quoted literal string term against which the field data will be tested.

Formatted Selection Criteria

The Formatted Selection Criteria panels relate specifically to generating record selection criteria based on assigned record structure and, optionally, formatted field data.

Filter (formatted) - INCLUDE/EXCLUDE record-types

The FILTER (formatted) - INCLUDE/EXCLUDE record-types panel (ZZSGFLTI) is an interactive panel window, opened on selecting Action 3. in the Create File Filter panel.

The type of filter (INCLUDE or EXCLUDE) is determined by the selection criteria type specified in the **Create File Filter** panel.

The panel contains a table where each selected and non-excluded row represents an INCLUDE or EXCLUDE sub-clause based on a record structure defined within the structure file. By default, the table contains one row for each record structure defined within the structure file.

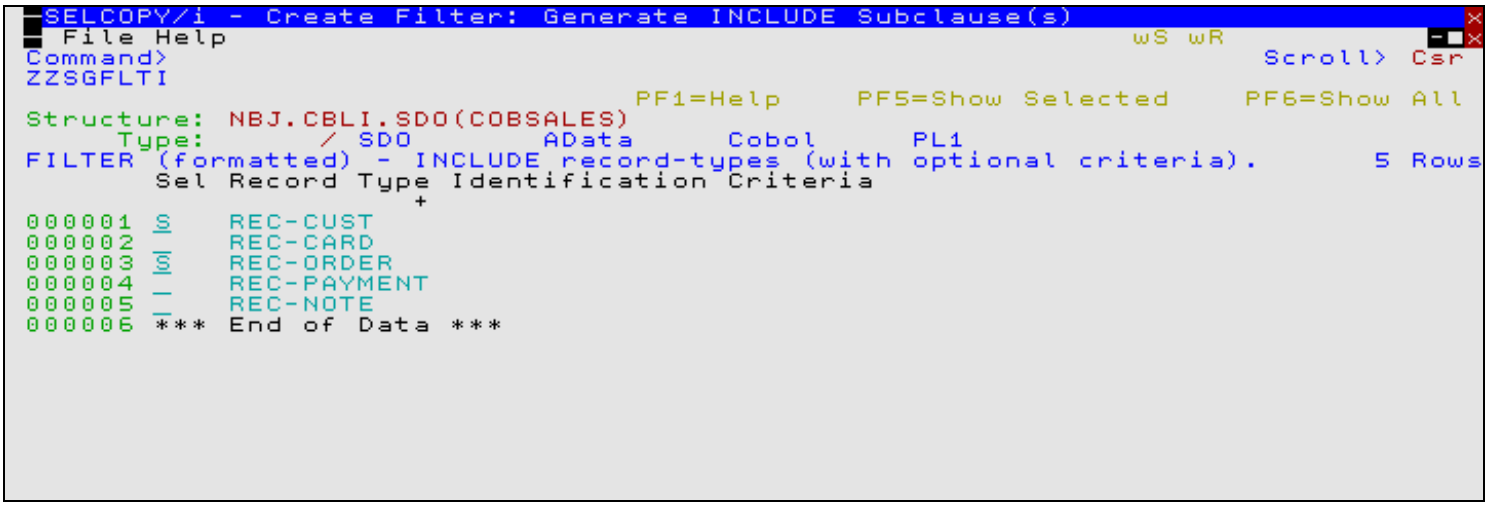


Figure 180. SELCOPYi - Formatted - INCLUDE/EXCLUDE record-types.

Standard SELCOPYi table editing techniques should be used to duplicate, move or exclude table row entries for each INCLUDE/EXCLUDE sub-clause.

Select a record structure table entry for INCLUDE/EXCLUDE by entering "S" in the "Sel" column or placing the cursor on the table entry and pressing the <Enter> key or, if configured, double-clicking the left mouse button. To de-select a record structure table entry, simply remove the "S" from the "Sel" column.

<PF5> and <PF6> may be used to alternate between display of only selected rows and all rows in the table respectively. This has no effect on the generated filter, but is useful as a visual aid.

"Record Type" identifies the record structure name and "Identification Criteria" displays any USE WHEN record type criteria used to identify when the record structure is assigned to a record. These are output fields may not be updated. "Selection Criteria" indicates the number of potential field name based sub-expressions defined for the record structure.

On selecting a record structure, the Filter (formatted) - Selection Criteria is opened automatically, giving the user the option to further specify selection criteria based on named fields within the formatted record.

When the filter is generated, each INCLUDE/EXCLUDE sub-clause is separated by a logical OR operation. Therefore, if required, a record structure table row entry may be duplicated in order to provide alternative INCLUDE/EXCLUDE sub-clause record selection criteria based on different values within fields belonging to the same record structure.

Pressing <PF3> to exit the panel, will save the table INCLUDE/EXCLUDE sub-clauses and return to the Create File Filter panel.

Filter (formatted) - Selection Criteria

The FILTER (formatted) - Selection Criteria panel (ZZSGFLTW) is an interactive panel window, opened when a record structure is selected from the FILTER (formatted) - INCLUDE/EXCLUDE record-types panel.

The panel contains a table where each row represents a sub-expression of the single INCLUDE or EXCLUDE sub-clause (WHERE expression) generated for the selected record structure. Each generated sub-expression is based on values referenced by one or more field names defined in the record structure.

By default, the table contains one row for every level of nested field defined within the record structure. These entries may be duplicated, copied or re-ordered before generating the sub-expression. Note that table entries that do not have a test value will **not** be included as part of the generated sub-expression.

The sub-expressions are separated by logical operator AND or OR, where AND is higher in the order of operator precedence than OR. i.e. A AND B OR C is equivalent to (A AND B) OR C. To filter on fields A AND (B OR C), then two rows should exist for the same record structure in the **FILTER (formatted) - INCLUDE/EXCLUDE record-types** table so that one tests fields A AND B and the other tests fields A AND C.

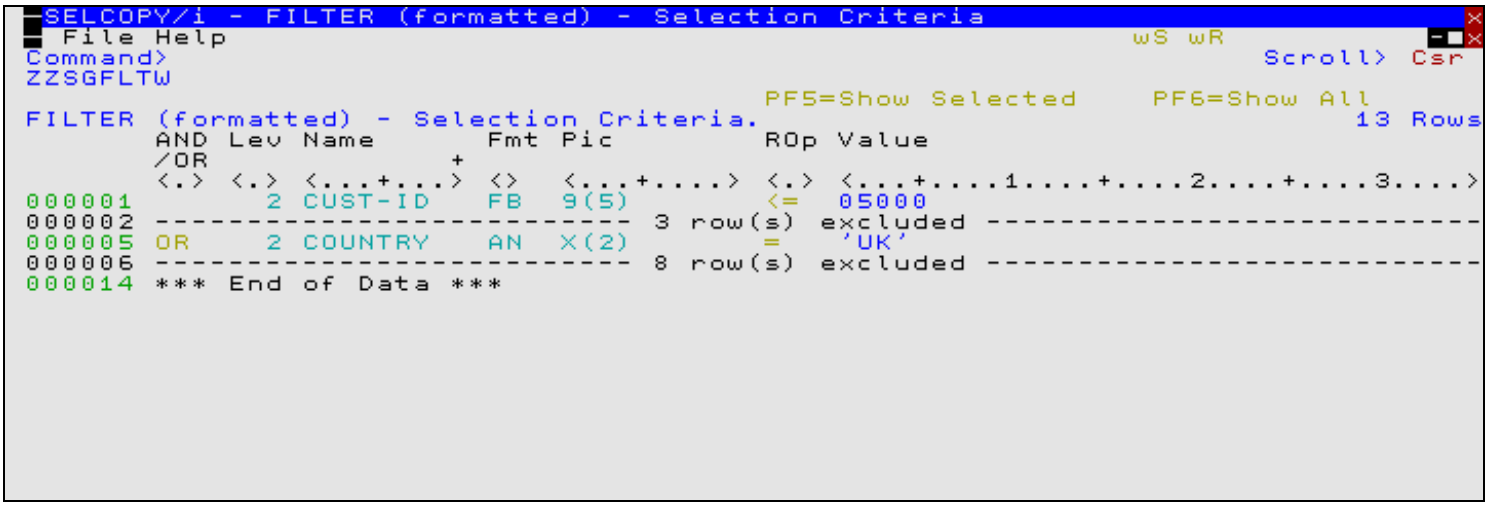


Figure 181. SELCOPYi - Formatted Selection Criteria.

Standard SELCOPYi table editing techniques should be used to update, move, copy and exclude table row entries for each filter sub-expression.

Each table row contains input fields for logical operator (AND/OR), optional opening parentheses "(", relational operator (ROp), the value against which the field will be tested and optional closing parentheses ")". Null or invalid enties may be entered in the **AND**/**OR** and **ROp** fields to display and select permitted entries for these fields.

All other fields are output (non-updatable) fields providing useful information about the field to be tested (i.e. the field's level of nesting, its name, data type (format) and, if defined, its picture definition.)

Since logical operators AND and OR are dyadic, the **AND**/**OR** field of the first table row (the first term of the expression) is always blank and cannot be updated.

<PF5> and <PF6> may be used to alternate between display of only selected rows and all rows in the table respectively. This has no effect on the generated filter, but is useful as a visual aid.

The user can display the "form" view of any individual row using the ZOOM command (assigned to <PF2> by default.) As well as providing additional helpful comment information, the single record view allows the user to use the panel EXPAND feature to enter a Value field entry that is longer than the provided input area.

Pressing <PF3> to exit the zoomed view of the panel, will update the filter sub-expression table row and return to the multi-record view of the table. Pressing <PF3> again to exit the Selection Criteria panel, will save the table sub-expressions and return to the FILTER (formatted) - INCLUDE/EXCLUDE record-types panel.

Fields in this table correspond to the filter WHERE *expression*.

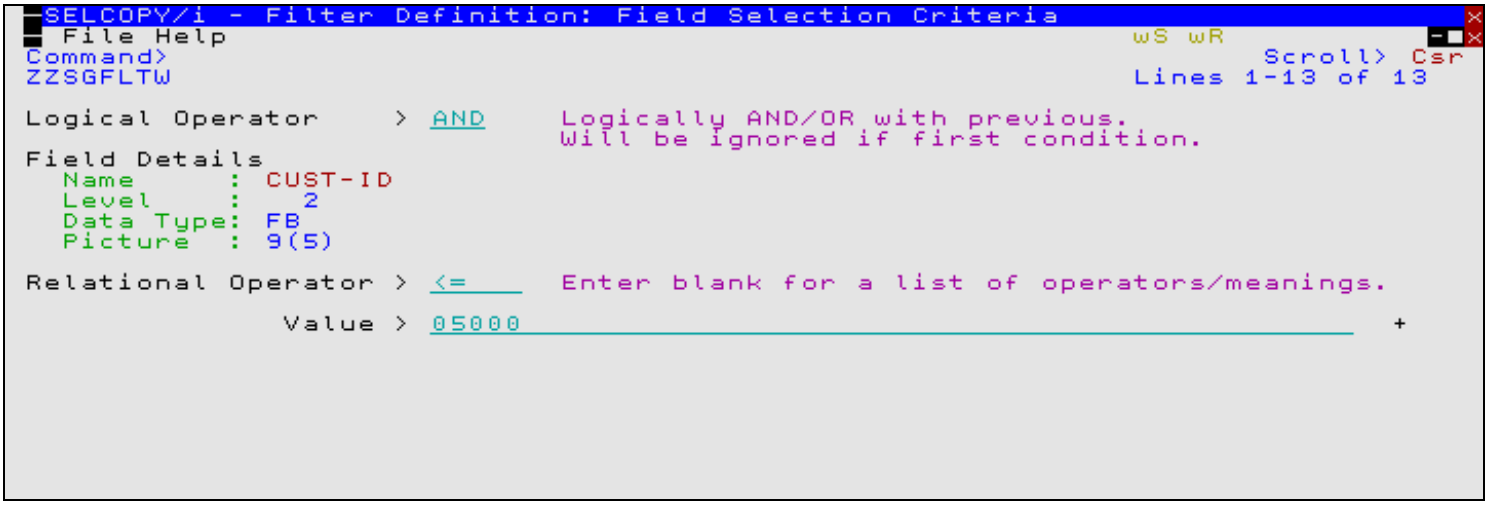


Figure 182. SELCOPYi - Formatted Selection Criteria Zoomed View.

Panel Input Fields

By default, field entries are populated with arguments and options entered in the table view for that entry.

`Logical Operator> AND|OR`
Logical operator that identifies the relationship of this field test sub-expression with the previously specified sub-expression. If this is the first, then this field is ignored. Permissible operators are **AND** or **OR**.

`Opening Bracket(s)>`
One or more leading "(" that the user may provide in order to specify parenthesised expressions.

Space is provided for only one "(" in the table view of the selection criteria panel. To specify more than one "(", first use the **"ZOOM"** key **(Shift-F4)** to display a form panel corresponding to the focus table row.

The form panel provides larger entry fields for many of the table columns.

`Field Details:`
Informational output fields that describe the formatted field.

`Level:`
Level of nesting below the first level (GROUP or STRUCTURE) field.

`Name:`
Field name.

`Data Type:`
The field's defined data type. See *"SDE Data Types"* for details of supported data types and their abbreviated names.

`Picture:`
If the source field is defined with a COBOL picture string, its representation is displayed in this field.

`Relational Operator>`
Identifies the type of test to be performed on the field data. Enter blank in this field to display a selectable list of supported relational operator symbols and a brief description of each.

`Value>`
Specifies the literal string term against which the field data will be tested.

Values may be specified using standard Expression Terms.

To specify values longer than the available entry field length, first use the **"ZOOM"** key **(Shift-F4)** to display a form panel corresponding to the focus table row.

The form panel provides a larger entry field for the Value. If this is still inadequate, you may press the "EXPAND" key (Shift-2) with your cursor in the form entry field in order to open a Text-Edit style window providing for values of up to 256 bytes.

`Closing Bracket(s)>`
One or more leading ")" that the user may provide in order to specify parenthesised expressions.

Space is provided for only one ")" in the table view of the selection criteria panel. To specify more than one ")", first use the **"ZOOM"** key **(Shift-F4)** to display a form panel corresponding to the focus table row.

The form panel provides larger entry fields for many of the table columns.

Print/Report Features Menu (=11)

The Print/Report Features Menu panel (ZZSGPRTM) is an interactive panel window opened on selection of option 11. from the SELCOPYi Primary option menu.

Options

1 Print	Print a dataset
2 Report	Create a formatted report with optional sort/totals etc
3 DB2 Report	Create a Report from a DB2 Table or from SQL
4 SMF Report	Create a Report from an SMF Dataset

Print Data File (=11.1)

Print Data File - Input

The Print Data File panel view (ZZSGPRT0) is an interactive panel window, opened on selection of option 11 from the SELCOPY/i Primary option menu, or by typing the primary command PRINT with no parameters from a non-SDE (Data Edit) browse or edit document window.

For Data Edit windows, the SDE primary command, PRINT, will open the SDE PRINT File panel to perform similar print functionality for text in the focus Data Edit window.

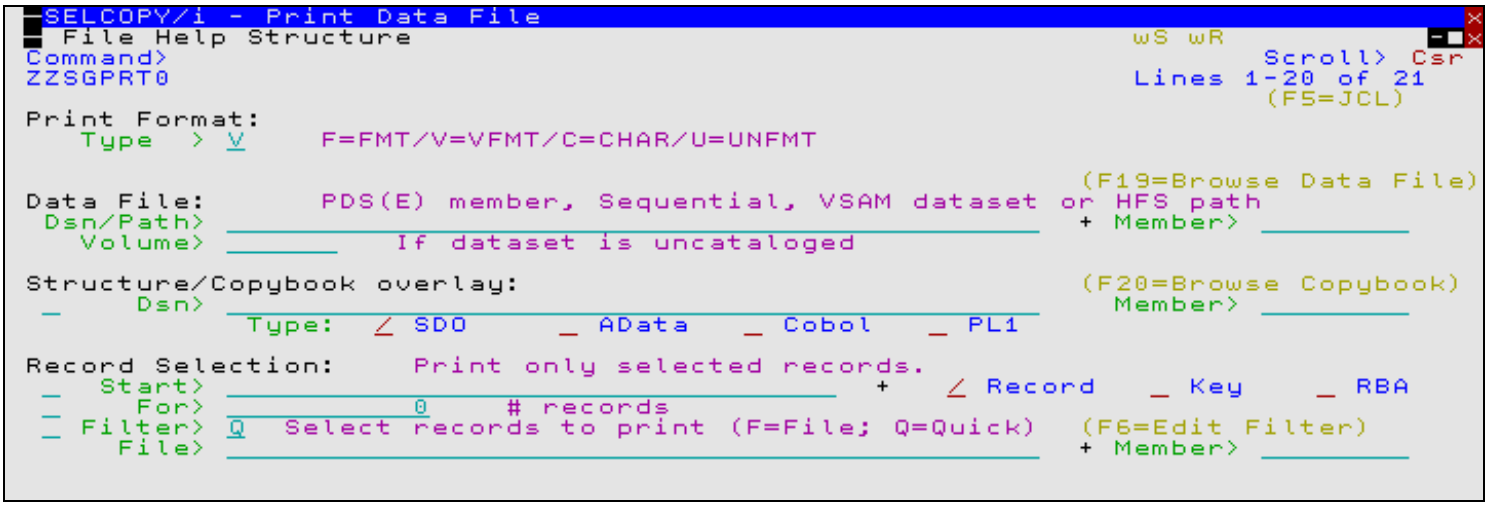


Figure 183. SELCOPY/i - Print Data File.

The **Print Data File** utility provides users with the ability to print any sequential, VSAM, library member or HFS file with data having been formatted using any display formatting supported by an SDE edit/browse view.

Print output may be written to any of the following:

1. An existing sequential or VSAM data set or a PDS/PDSE library member of any RECFM, BLKSIZE and LRECL. The print page width will default to be the LRECL (or VSAM maximum record length) of the output data set. This type of output data set may be specified via its DSN (and member name) or a pre-allocated DDName.
2. An HFS file. The print page width default is 133.
3. A SYSOUT data set. The print page width default is 133.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will will proceed to the next Print Data File panel view based on the selected print format.

Type **JCL** (F18) to generate the PRINT File batch job based on the existing values set in the supporting panel views, otherwise the print process will run in the foreground.

Print Data File - Input Panel Fields

Print Format:
Identifies the type of print that will occur.

Records will be formatted or unformatted in either single record or multi record view and relevant display options applied prior to printing the file.

Supported print formats are:

VFMT	Vertical Formatted records (multi record view).
FMT	Formatted records in single record view.
CHAR	Unformatted records in multi record view.
UNFMT	Unformatted records in single record view.

PDS/PDSE member, Sequential, VSAM or HFS path:
Input fields which together identify a single existing, sequential or VSAM data set, GDG file generation, HFS file or PDS/PDSE library member to be printed.

Name>
Identifies the fully qualified data set name or an absolute or relative HFS file path.
A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.
A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Volume>
Specifies the name of the input data set volume. This is required only if output is to an uncataloged data set.

Structure/Copybook overlay:
If activated, this option defines fields which together specify a cataloged structure file (COBOL or PL1 Copybook, ADATA file or a SELCOPY/i SDO) used to format the printed records. The structure may be a sequential data set or a PDS/PDSE library member.

If no structure is specified, each data set record will be of the default record type "Unmapped", i.e. a single character field of length equal to that of the record.

These fields must be selected and contain valid entries if record data is to be formatted before it is printed (print format VFMT or FMT).

Dsn>
Identifies the fully qualified data set name of a sequential data set or PDS/PDSE library containing a COBOL or PL1 Copybook, ADATA file or SDO structure.
A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.
A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Type:
Indicate the type of structure (COBOL, PL1, ADATA or SDO).

Record Selection:
Fields which together identify criteria by which a subset of records from the file are selected for browse/edit. If any record selection field is activated for data edit, then Edit In-Place is performed regardless of the selected edit technique.

Start>
If activated, the **Start>** field identifies the first record in the file at which printing will start. Records occurring sequentially before the start record will be excluded. If this field is not activated, records are selected beginning at record 1.

This input field may contain a record number, an RBA number (for ESDS input only), or a key string (for KSDS input only).

A record/RBA number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**. A key string may be specified as a literal **abc** or **'abc'**, which will be upper cased before keyed look-up, character string **C'abc'** (character case preserved) or a hexadecimal string **X'818283'**.

Record | Key | RBA
Identifies the type of start value specified in the **Start>** field. Enter "/" in the appropriate, mutually exclusive parameter field.

For>

If activated, the **For>** field specifies the maximum number of records within the file to be printed. If this field is not activated, records are selected beginning at start record and ending at the last record in the file.

Filter>

If activated, the **Filter>** field specifies options to either generate a new record filter or use an existing record filter file. A record filter will perform further subsetting on input file records selected for processing by the Start> and/or For> input fields.

Filter options are as follow:

Q	On pressing <PF6>, the Quick Filter dialog panel will be opened in order to generate a temporary filter on the unformatted record data.
F	Use a permanent filter identified by the sequential data set or member name in the File> field. On pressing <PF6>, the Create File Filter dialog panel will be opened in order to display the contents of an existing filter file or create and save a new filter file. This option requires specification of a filter fileid.

File>

Identifies the fully qualified data set name of a sequential data set or PDS/PDSE library containing a record filter. Quotes are unnecessary but permitted.
A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>

If the File> field contains the DSN of a PDS/PDSE library, then this field must reference a member name within that library which contains a record filter.
A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Print Data File - VFMT Options

The **Print Data File - VFMT Options** panel view is displayed after pressing the <Enter> key or, if configured, double-clicking the left mouse button when Print Format "VFMT" has been selected in the Print Data File panel view.

This panel allows additional SDE view display formatting and record-type based record and field selection on the vertically formatted file records.

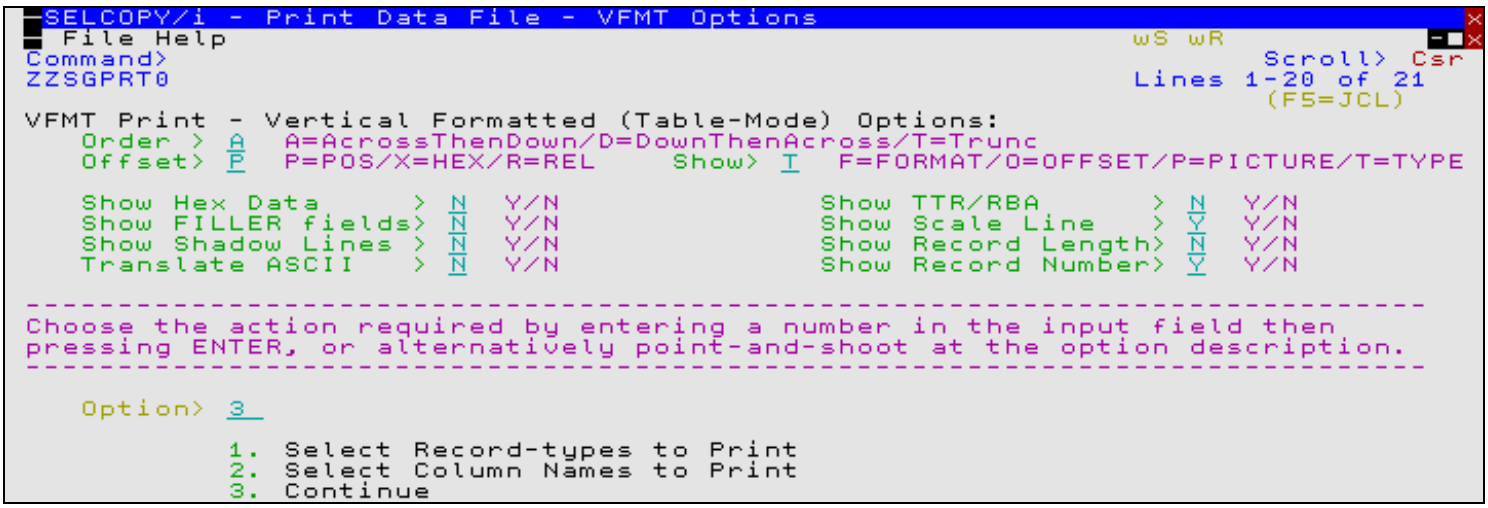


Figure 184. SELCOPY/i - Print Data File (VFMT Options).

Print Data File - VFMT Options Primary Commands

VIEW - Select Record-Types to Print

By default records of any record-type are included in the printed output. Type the VIEW command (F22) to restrict the printed records to only records assigned the specified record-types.

A separate panel will be displayed containing a list of record types defined by the **Structure/Copybook overlay** SDO or copybook. For informational purposes only and where one exists, each record-type is accompanied by its **USE WHEN** condition, used by SDE to identify assignment of that particular record-type.

From this panel, select those record types to be included in the printed output. (If DOWNTHENACROSS has been selected, only one record-type may be selected.)

SELECT (SEL) - Select Column Names to Print

By default all fields (from all selected record-types) are included in the print output Type the SELECT command (F23) to restrict the printed output to a subset of field names identified within those record-types selected for print.

A separate panel will be displayed containing a list of record-types defined by the **Structure/Copybook overlay** SDO or copybook. For informational purposes only and where one exists, each record-type is accompanied by its **USE WHEN** condition, used by SDE to identify assignment of that particular record-type.

From this panel, select those record-types for which fields are to be excluded from the printed output. Another panel will be opened for each selected record type allowing the user to exclude field columns from the print.

Print Data File - VFMT Options Panel Fields

Order>

Where printed lines are longer than the output page width, then this field value determines the action taken on the overflowing line data.

Unless T (Truncate) is selected, overflow line data is printed on as many continuation pages as is required to accomodate the longest print line in the current set of print lines. Note that a set of print lines is the number of SDE view lines that may be displayed for the specified page depth.

A - Across Then Down

Indicates that continuation pages for the current set of lines are to be printed before scrolling down to the next set of print lines. i.e. Print all pages scrolling across to the right then scroll down to the first page of the next set of lines and repeat the process.

This option corresponds to parameter ACROSSTHENDOWN of the SDE primary command PRINT.

Because only as many pages are printed to accomodate the longest line in the current set of print lines, the number of pages printed scrolling across may be different for each set of print lines.

D - Down Then Across

Indicates that the first page of all sets of print lines are to be printed before printing the next (continuation) page of all sets of print lines and repeating this process until all contuation pages have been printed. i.e. Print pages scrolling down, scroll back to the first set of print lines, scroll across to the right once and repeat the process.

This option corresponds to parameter DOWNTHENACROSS of the SDE primary command PRINT and is suitable only when formatted records of a single record-type are selected for display. (See "Select Record-types to Print" if the SDO used contains multiple record-type definitions.)

If this option is selected and records of different record-types are displayed concurrently, then the print will fail with the following message:

```
ZZSD472E The DOWNTHENACROSS format of the PRINT command requires
         that only one record type is visible.
```

Because only one record-type is used, the number of pages printed scrolling across will be the same for each set of print lines.

T - Truncate

Indicates that no continuation pages are to be printed so truncating the print lines. Only the first page of all sets of print lines will be printed.

This option corresponds to parameter TRUNC of the SDE primary command PRINT.

Show> F=FORMAT/O=OFFSET/P=PICTURE/T=TYPE/X=None

Determines the contents of the field format/location header line.

F	Displays ***length*/*format*** where *length* identifies the field's length within the unformatted record, and *format* identifies the field's data-type in a descriptive form. e.g. **'30/CHAR'** indicates an Alpha-Numeric character field of length 30 bytes.
O	Displays ***nnnnn*** representing the field's location within the unformatted record. The format of the offset display is determined by the Offset> value.
P	For COBOL or PL1 source structures or SDO structures generated from COBOL or PL1 copy books, this option will display a field's defined PICTURE string.
T	Displays ***data-type position*:*length*** where *position* and *length* identify the field's position and length within the unformatted record. e.g. **'AN 111:30'** indicates an Alpha-Numeric character field of length 30 bytes starting at position (decimal) 111 of the unformatted record.
X	This option will suppress the field format/location header line.

This option corresponds to the SDE primary command SHOW.

Offset>

Determines the format of the offset field description and scale header, displayed when **Show> O** and **Show Scale Line> Y** is selected.

P	Displays the location of the start of each field as a decimal position and a decimal scale line.
X	Displays the location of the start of each field as a hexadecimal offset and a hexadecimal scale line.
R	Displays the location of the start of each field as a decimal offset and a decimal scale line.

This option corresponds to the SDE option SET OFFSET.

Show Hex Data>
Determines whether hexadecimal display of the file record data is on or off.

This option corresponds to the SDE primary command HEX.

Show FILLER fields>
Determines whether unnamed fields (e.g. COBOL FILLER fields) are included in the printed output.

This option corresponds to the SDE option SET UNNAMED.

Show Shadow Lines>
Determines whether shadow lines, used to mark groups of suppressed records (i.e. records assigned a record-type that has not been selected for view), are included in the printed output.

This option corresponds to the SDE option SET SHADOW.

Translate ASCII>
Determines whether or not character display of the file record data is interpreted as ASCII or EBCDIC.

This option corresponds to the SDE option SET ASCII.

Show TTR/RBA>
Determines whether or not the record identification column is included in the printed output.

For non-VSAM data sets, the record identification is by Relative Volume, Track and Physical Record number (TTR) followed by an offset within the physical record.

For VSAM data sets, the record identification is by Relative Byte Address (RBA). The column is displayed as a decimal value with header "RBA".

This option corresponds to the SDE option SET RECINFO ID.

Show Ref Numbers>
Determines whether or not the field reference numbers (**#nn**) header line is included in the printed output.

This option corresponds to the SDE option SET REFERENCE.

Show Scale Line>
Determines whether or not the scale header line, providing a counting guide for the width of displayed fields, is included in the printed output.

This option corresponds to the SDE option SET SCALE.

Show Record Length>
Determines whether or not the record (or record segment) length column is included in the printed output.

This option corresponds to the SDE primary command RECLENGTH.

Show Record Number>
Determines whether or not the line prefix area displaying the record number is included in the printed output.

This option corresponds to the SDE option SET PREFIX LEFT 8 PHYSICAL.

Print Data File - FMT Options

The **Print Data File - FMT Options** panel view is displayed after pressing the <Enter> key or, if configured, double-clicking the left mouse button when Print Format "FMT" has been selected in the Print Data File panel view.

This panel allows additional SDE view display formatting and record-type based record and field selection on the single record view, formatted file records.

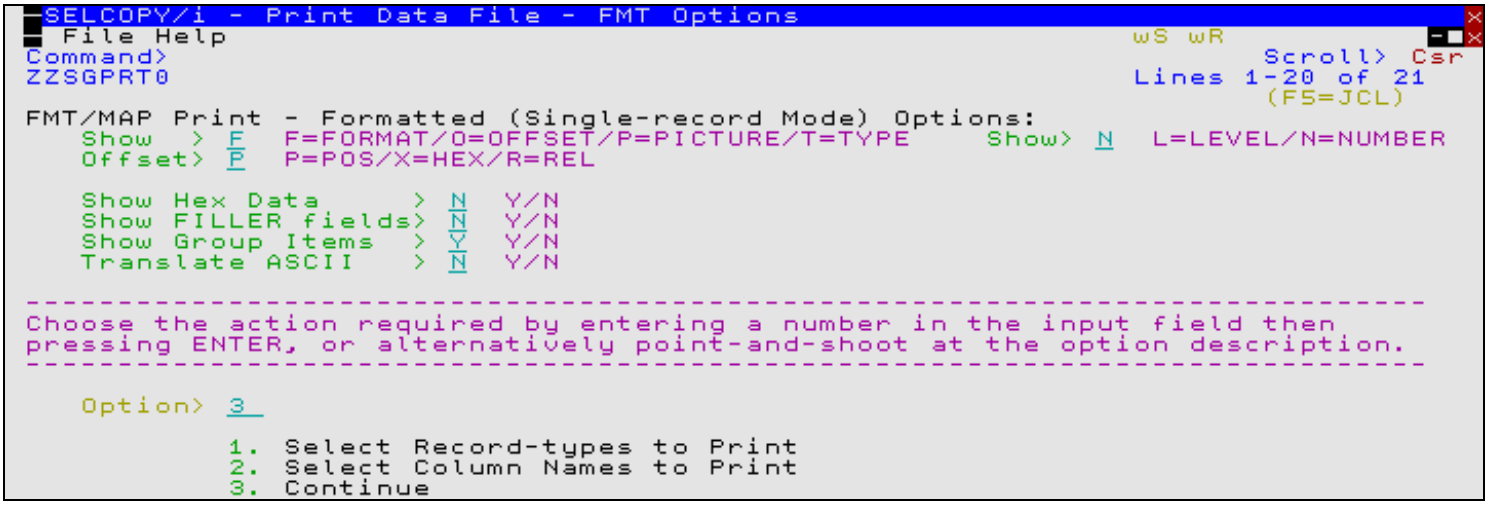


Figure 185. SELCOPY/i - Print Data File (FMT Options).

Print Data File - FMT Options Primary Commands

VIEW - Select Record-Types to Print

By default records of any record-type are included in the printed output. Type the VIEW command (F22) to restrict the printed records to only records assigned the specified record-types.

A separate panel will be displayed containing a list of record types defined by the **Structure/Copybook overlay** SDO or copybook. For informational purposes only and where one exists, each record-type is accompanied by its **USE WHEN** condition, used by SDE to identify assignment of that particular record-type.

From this panel, select those record types to be included in the printed output. (If DOWNTHENACROSS has been selected, only one record-type may be selected.)

SELECT (SEL) - Select Column Names to Print

By default all fields (from all selected record-types) are included in the print output Type the SELECT command (F23) to restrict the printed output to a subset of field names identified within those record-types selected for print.

A separate panel will be displayed containing a list of record-types defined by the **Structure/Copybook overlay** SDO or copybook. For informational purposes only and where one exists, each record-type is accompanied by its **USE WHEN** condition, used by SDE to identify assignment of that particular record-type.

From this panel, select those record-types for which fields are to be excluded from the printed output. Another panel will be opened for each selected record type allowing the user to exclude field columns from the print.

Print Data File - FMT Options Panel Fields

Show> L=LEVEL/N=NUMBER/B=Both/X=Neither

Determines the contents of the field name display.

L	Suppresses display of the field reference numbers assigned by SDE and replaces them with the field's hierarchical level number as a prefix to the field name. Additionally, the level number and field name is indented for each increment in level number.
N	Suppresses display of the field's hierarchical level number and displays the field reference numbers assigned by SDE instead (e.g. #2)
B	Shows both the field's hierarchical level number and the field reference numbers assigned by SDE.
X	Suppresses both the field's hierarchical level number and the field reference numbers assigned by SDE.

This option corresponds to the SDE primary command SHOW LEVEL/NUMBER.

Show> F=FORMAT/O=OFFSET/P=PICTURE/T=TYPE/X=None

Determines the contents of the field format/location display column.

F	Displays ***length*/*format*** where *length* identifies the field's length within the unformatted record, and *format* identifies the field's data-type in a descriptive form. e.g. **'30/CHAR'** indicates an Alpha-Numeric character field of length 30 bytes.
O	Displays ***nnnnn*** representing the field's location within the unformatted record. The format of the offset display is determined by the Offset> value. Furthermore, start-end values of character field continuation lines are replaced by an offset value matching the offset format set for other field entries.
P	

	For COBOL or PL1 source structures or SDO structures generated from COBOL or PL1 copy books, this option will display a field's defined PICTURE string.
T	Displays ***data-type position:length*** where *position* and *length* identify the field's position and length within the unformatted record. e.g. **'AN 111:30'** indicates an Alpha-Numeric character field of length 30 bytes starting at position (decimal) 111 of the unformatted record.
X	This option will suppress the field format/location display column.

This option corresponds to the SDE primary command SHOW FORMAT/OFFSET/PICTURE/TYPE.

Offset>
Determines the format of the offset field description and field contents scale header, displayed when **Show> O** and **Show Scale Line> Y** is selected.

P	Displays the location of the start of each field as a decimal position and a decimal scale line.
X	Displays the location of the start of each field as a hexadecimal offset and a hexadecimal scale line.
R	Displays the location of the start of each field as a decimal offset and a decimal scale line.

This option corresponds to the SDE option SET OFFSET.

Show Hex Data>
Determines whether hexadecimal display of the file record data is on or off.

This option corresponds to the SDE primary command HEX.

Show FILLER fields>
Determines whether unnamed fields (e.g. COBOL FILLER fields) are included in the printed output.

This option corresponds to the SDE option SET UNNAMED.

Show Group Items>
Determines whether or not each occurrence of a group item is included in the printed output. Group items correspond to structure, union and root array field names.

This option corresponds to the SDE option SET GROUP.

Translate ASCII>
Determines whether or not character display of the file record data is interpreted as ASCII or EBCDIC.

This option corresponds to the SDE option SET ASCII.

Print Data File - Select Record-Types

The **Print File - Select Record-Types** panel (ZZSGPRTV) is displayed following selection of option 1. from either the Print Data File - VFMT Options or Print Data File - FMT Options panel view.

This panel contains a list of all record-types, defined by the SDO or copy book specified in Structure/Copybook overlay, which are presented to the user as an editable IPO table. Any **USE WHEN** condition, used to determine whether record data fits the record type definition, is also displayed.

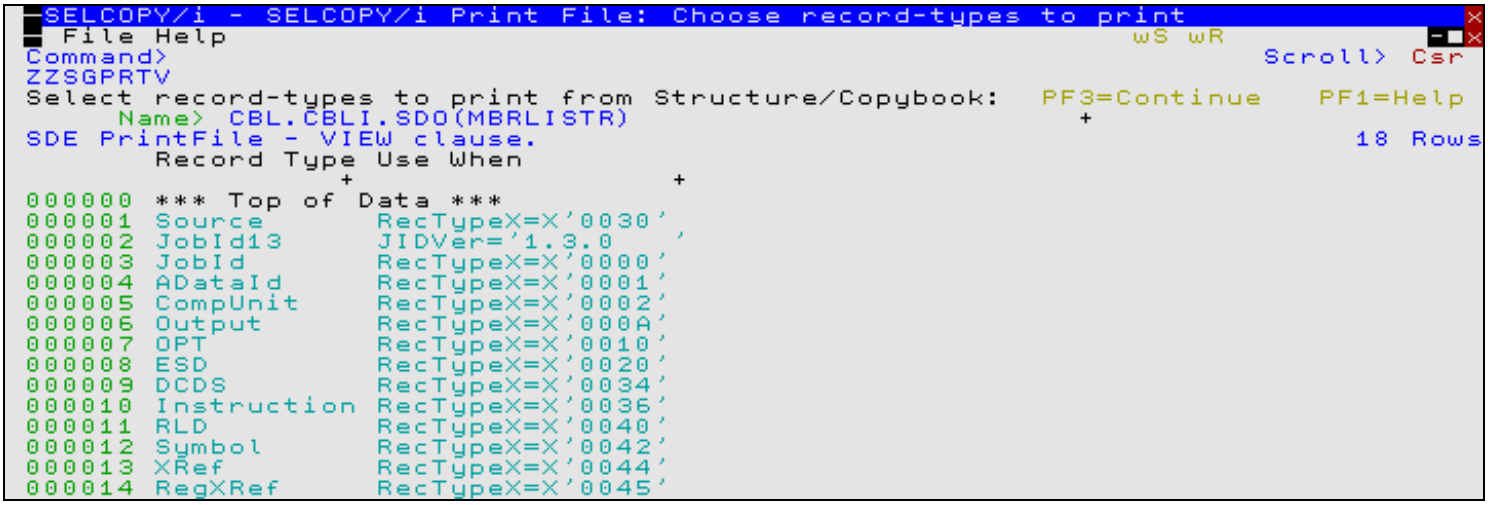


Figure 186. SELCOPY/i - Print Data File - Select Record-Types.

Records assigned a record-type that is included in this list will be selected for print. Use standard SELCOPY/i table editing techniques to exclude or delete record types from this list and so exclude records assigned these record-types from being printed.

For example, the following command may be executed to exclude all rows except those where the record type name begins with literal "**ABC-**":

```
WHERE   ViewRT >> 'ABC-'
```

Pressing <PF3> to exit the panel, will also save the table of selected record-types and return to the previous (VFMT or FMT options) panel view as appropriate.

Default is to include records assigned any of the available record-types.

Print Data File - Select Field Names

The **Print Data File - Select Field Names** panel (ZZSGPRTS) is displayed following selection of option 2. from either the Print Data File - VFMT Options or Print Data File - FMT Options panel view.

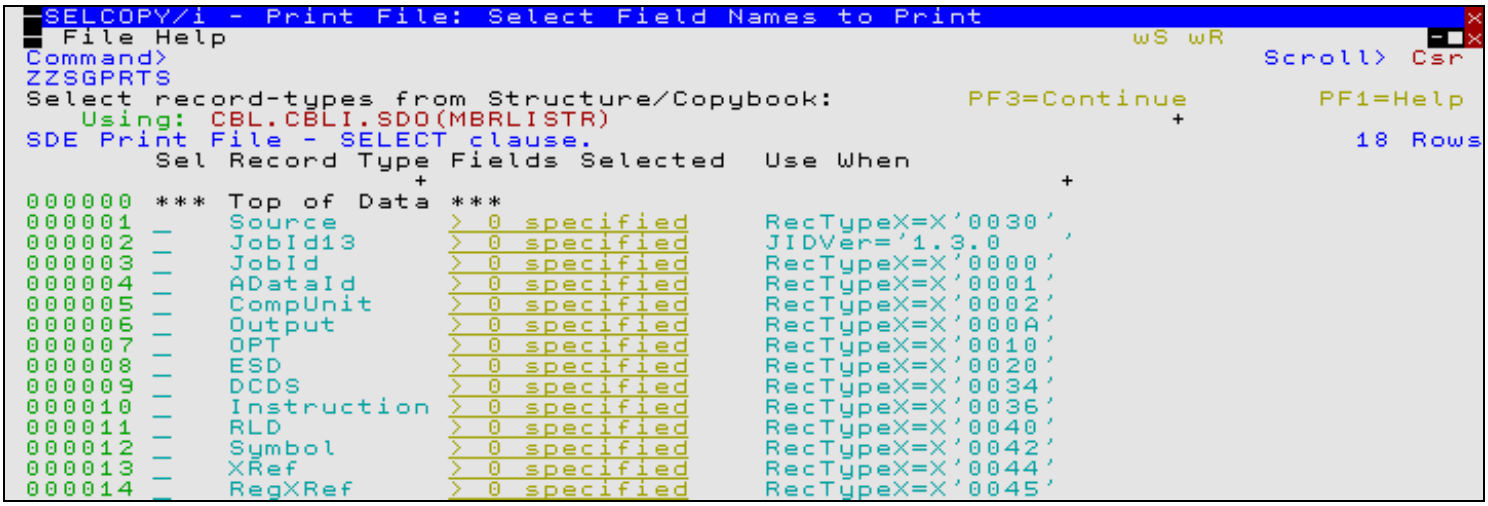


Figure 187. SELCOPY/i - Print Data File - Select Field Names.

Select each record-type for which specific field columns are to be selected by entering 'S' against the record type in the **Sel** column or by positioning the cursor on the required record type then either pressing the <Enter> key or, if configured, double-click the left mouse button. To deselect the record type key field definition, remove the 'S' against its entry in the 'Sel' column.

For each selected record type, the Print Data File - Field Names List panel (ZZSGPRTF) is opened displaying a list of fields that comprise the record-type. The list of field names should be edited so that only the required field names are diplayed. The order in which these fields occur in this list will be the order in which they appear in the printed output.

On returning from the selectable field list, the **Fields Selected** column will be updated to indicate the number of fields selected for print.

Note that, selecting fields from a record-type that has been excluded from the print in panel Print Data File - Select Record-Types (ZZSGPRTV), will generate the appropriate SELECT *field-name* syntax but will ultimately have no effect. i.e. The record-type must be included for print in order for its selected fields to be printed.

Pressing <PF3> to exit the panel, will also save the table of selected record-types for which selected fields will be printed, and return to the previous (VFMT or FMT options) panel view as appropriate.

Default is to print all field columns belonging to all selected record-types.

Print Data File - Field Names List

The **Print Data File - Field Names List** panel (ZZSGPRTF) is displayed for each record-type selected from the Print Data File - Select Field Names panel.

A list of field names, defined by the selected record-type, is presented to the user as an editable table. Standard SELCOPY/i table editing techniques should be used to exclude and re-order the fields so that only required fields are displayed and in the order in which they are to be printed.

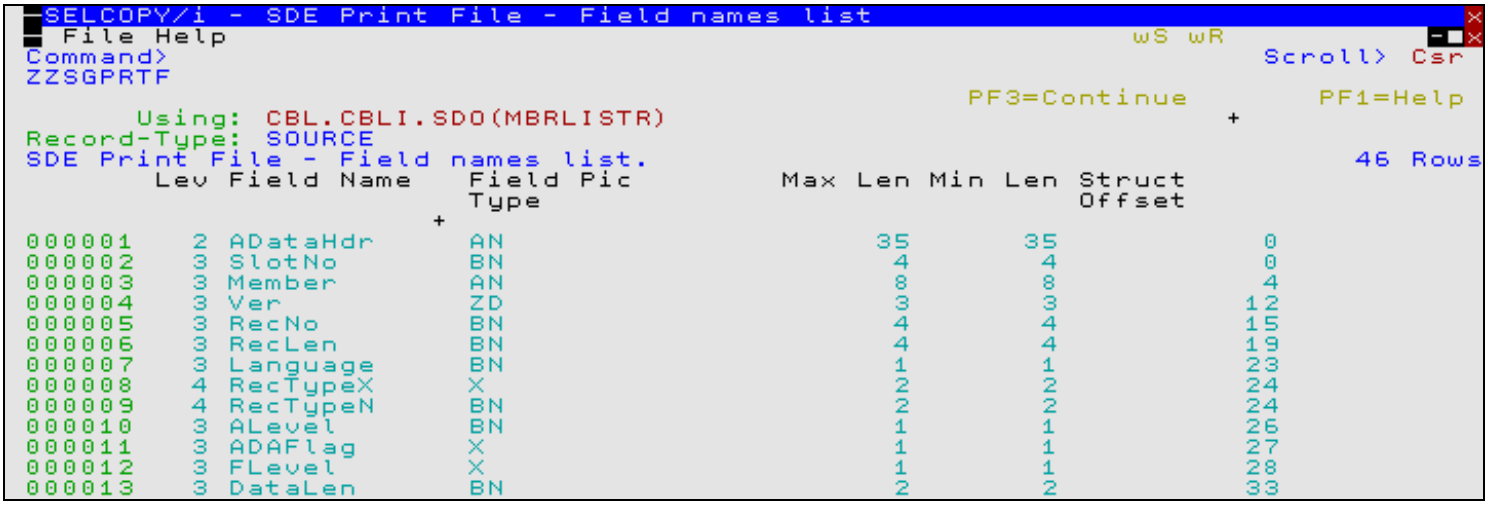


Figure 188. SELCOPY/i - Print Data File - Field Names List.

Only included field name entries are printed, therefore a field name may be excluded (as opposed to deleted) in order to exclude it from the print output. This has the benefit that the field may easily be included again later if necessary. For example, the following commands may be executed to filter (include) specific table rows:

`WHERE (length(strip(SelectFld),'T') > 5) and (#3 = 'BN')`
: Exclude all rows except those where the length of the Field Name entry is greater than 5 and the Field Picture Type is "BN". Note that the "Field Pic Type" column is field reference number 3.

`MORE SelectLev < 3`
: Include previously excluded entries where the field level is 1 or 2. (Entries that are already included will remain included.)

The order in which the field names occur is the order in which the fields will be appear in the printed output.

Pressing <PF3> to exit the panel, will also save the table of selected field names and return to the **Print Data File - Select Field Names** panel.

Default is to print all field columns belonging to all selected record-types.

Print Data File - CHAR Options

The **Print Data File - CHAR Options** panel view is displayed after pressing the <Enter> key or, if configured, double-clicking the left mouse button when Print Format "CHAR" has been selected in the Print Data File panel view.

This panel allows additional SDE view display formatting on the unformatted, multi record view of file records.

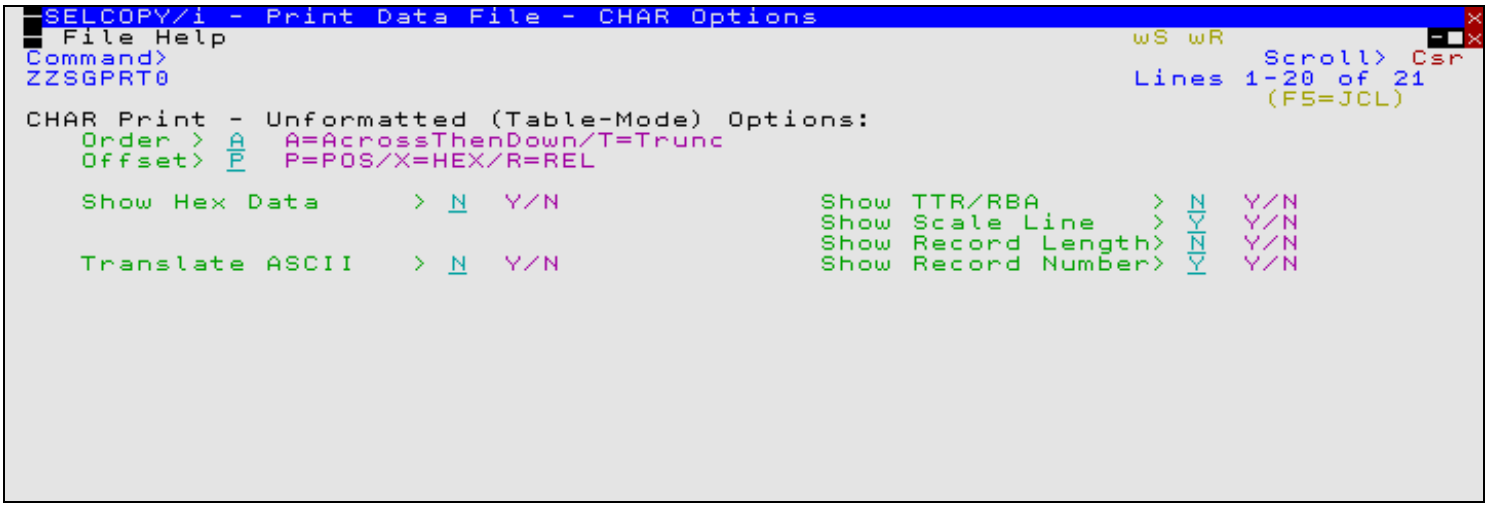


Figure 189. SELCOPY/i - Print Data File (CHAR Options).

Print Data File - CHAR Options Panel Fields

Order>
Where printed lines are longer than the output page width, then this field value determines the action taken on the overflowing line data.

Unless T (Truncate) is selected, overflow line data is printed on as many continuation pages as is required to accomodate the longest print line in the current set of print lines. Note that a set of print lines is the number of SDE view lines that may be displayed for the specified page depth.

A - Across Then Down
Indicates that continuation pages for the current set of lines are to be printed before scrolling down to the next set of print lines. i.e. Print all pages scrolling across to the right then scroll down to the first page of the next set of lines and repeat the process.

This option corresponds to parameter ACROSSTHENDOWN of the SDE primary command PRINT.

Because only as many pages are printed to accomodate the longest line in the current set of print lines, the number of pages printed scrolling across may be different for each set of print lines.

T - Truncate
Indicates that no continuation pages are to be printed so truncating the print lines. Only the first page of all sets of print lines will be printed.

This option corresponds to parameter TRUNC of the SDE primary command PRINT.

Offset>
Determines the format of the record contents scale header, displayed when **Show Scale Line> Y** is selected.

P or **R**	Displays a decimal scale line.
X	Displays a hexadecimal scale line.

This option corresponds to the SDE option SET OFFSET.

Show Hex Data>
Determines whether hexadecimal display of the file record data is on or off.

This option corresponds to the SDE primary command HEX.

Translate ASCII>
Determines whether or not character display of the file record data is interpreted as ASCII or EBCDIC.

This option corresponds to the SDE option SET ASCII.

Show TTR/RBA>
Determines whether or not the record identification column is included in the printed output.

For non-VSAM data sets, the record identification is by Relative Volume, Track and Physical Record number (TTR) followed by an offset within the physical record.

For VSAM data sets, the record identification is by Relative Byte Address (RBA). The column is displayed as a decimal value with header "RBA".

This option corresponds to the SDE option SET RECINFO ID.

Show Scale Line>
Determines whether or not the scale header line, providing a counting guide for the record data, is included in the printed output.

This option corresponds to the SDE option SET SCALE.

Show Record Length>
Determines whether or not the record length column is included in the printed output.

This option corresponds to the SDE primary command RECLENGTH.

Show Record Number>
Determines whether or not the line prefix area displaying the record number is included in the printed output.

This option corresponds to the SDE option SET PREFIX LEFT 6 PHYSICAL.

Print Data File - UNFMT Options

The **Print Data File - UNFMT Options** panel view is displayed after pressing the <Enter> key or, if configured, double-clicking the left mouse button when Print Format "UNFMT" has been selected in the Print Data File panel view.

This panel allows additional SDE view display formatting on single record view, unformatted file records.

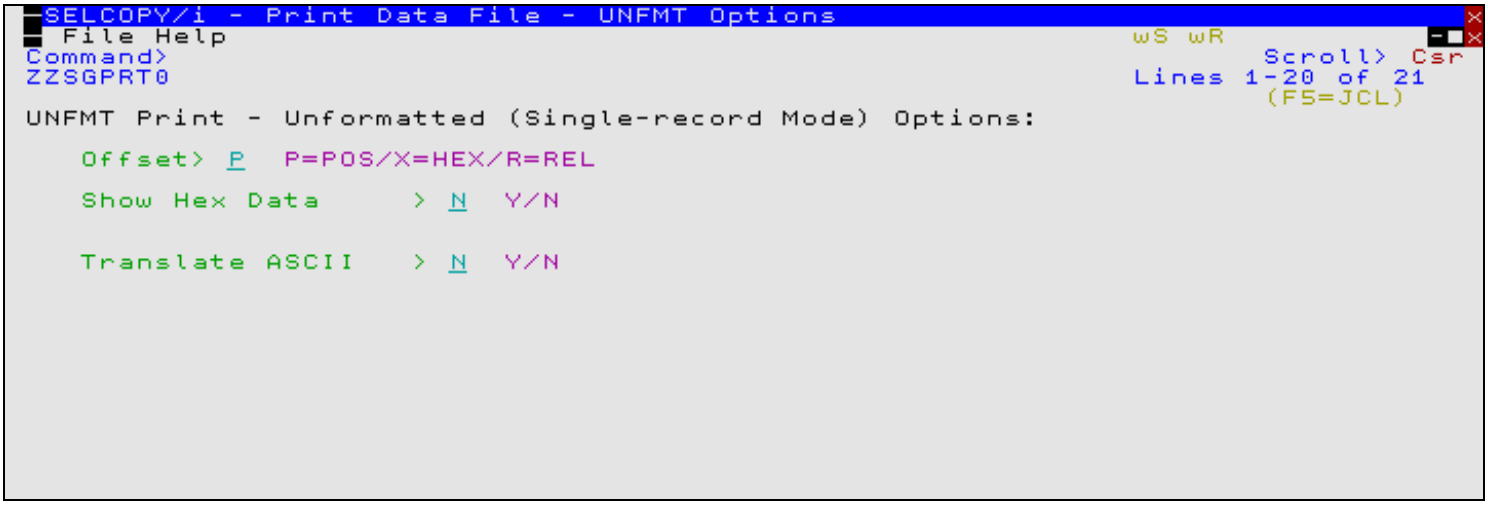


Figure 190. SELCOPY/i - Print Data File (UNFMT Options).

Print Data File - UNFMT Options Panel Fields

Offset>
Determines the format of the record contents scale header, displayed when **Show Scale Line> Y** is selected.

P or **R**	Displays a decimal scale line.
X	Displays a hexadecimal scale line.

This option corresponds to the SDE option SET OFFSET.

Show Hex Data>
Determines whether hexadecimal display of the file record data is on or off.

This option corresponds to the SDE primary command HEX.

Translate ASCII>
Determines whether or not character display of the file record data is interpreted as ASCII or EBCDIC.

This option corresponds to the SDE option SET ASCII.

Print Data File - Destination

The **Print Data File - Destination** panel view is displayed after pressing the <Enter> key or, if configured, double-clicking the left mouse button from the "CHAR" or "UNFMT" options panel view, or on selection of Option 3. in the "VFMT" or "FMT" options panel view.

This panel identifies the destination of the printed output.

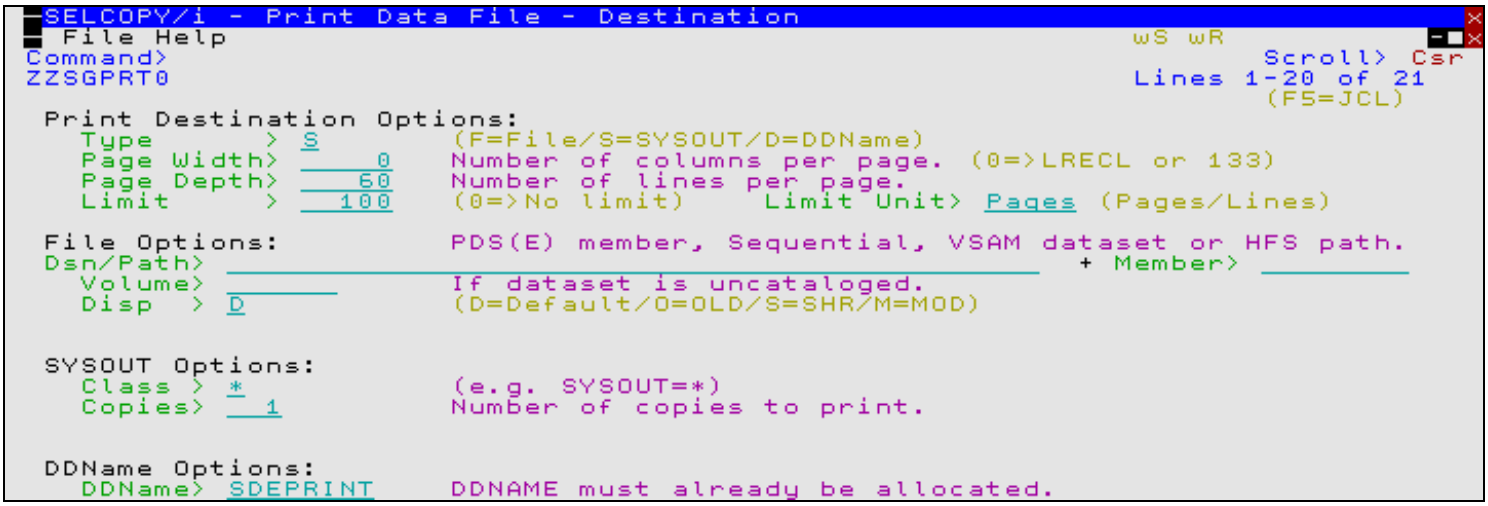


Figure 191. SELCOPY/i - Print Data File Destination.

Print Data File - Destination Panel Fields

Type>
Identifies the print output destination type.

F	Print output is to a FILE specified as a DSN, PDS/PDSE DSN and member name or HFS fileid.
S	Print to a system (SYSOUT) data set.
D	Print to a file or SYSOUT data set allocated to an existing DDName.

Page Width>
Set the print output page width (number of columns).

A page width value of 0 (zero) will use the default page width of 133 for SYSOUT and HFS file output, and the maximum record length (LRECL) for an output data set or library member.

Note that, for FMT and UNFMT printed output, page width is restricted to a maximum of 255 print columns.

Page Depth>
Set the print output page depth (number of lines).

The page depth value includes the 5 Print header lines so that the number of lines of data printed will be 5 less than the page depth value.

Limit>
Limits the amount of output data printed.

The value in this field correspond to a number of units as specified by **Limit Unit>**. A limit value of 0 (zero) indcates that there is no limit imposed on the printed output.

Limit Unit>
Identifies the units (LINES or PAGES) represented by the **Limit>** value.

A PAGE limit corresponds to a number of lines equal to the number of pages multiplied by the page depth value.

Note that a LINE limit includes any blank lines printed on each page to satisfy the page depth specification.

File Options:
Applicable only when print destination type is "File", File options specify parameters used by the PRINT operation to dynamically allocate a sequential or VSAM data set, PDS/PDSE library member or HFS file.

If a specified DSN does not belong to an existing library, sequential or VSAM data set, the Allocate Non-VSAM panel is displayed so that a new output print file may be allocated before proceeding to generate the JCL.

Dsn/Path>
Member>
Volume>
Identifies the fileid (DSN) of the sequential or VSAM data set, PDS/PDSE library member or HFS file to be printed. Dataset names must be fully qualified, quotes being unnecessary but permitted.
A selectable list of files will be presented if wildcards are entered, or dataset is an existing PDS/PDSE library and member is left blank.

Disp>
Identifies the disposition used when allocating a sequential, VSAM or library DSN for output. Note that output to an HFS file does not involve dynamic allocation and so, unless DISP MOD is specified, this option is ignored.

D	Use the PRINT utility default disposition for the type of data set being allocated (i.e. DISP=OLD for sequential and VSAM data sets, DISP=SHR for PDS/PDSE libraries.)
O	DISP=OLD. Exclusive, unshared ENQ for overwrite of any existing file data.
S	DISP=SHR. Shared ENQ for overwrite of any existing file data.
M	DISP=MOD. Exclusive, unshared ENQ for appending output to existing file data. For HFS file output, "M" is used to indicate write append to the file even though no dynamic allocation occurs. DISP=MOD is ignored if output is to a PDS/PDSE library member.

SYSOUT Options:
Applicable only when print destination type is "SYSOUT", SYSOUT options specify parameters used by the PRINT operation to dynamically allocate a system (SYSOUT) data set.

Class>
Identifies the SYSOUT class.

Copies>
Identifies the number of file copies to be printed (1-255). This parameter corresponds to the SYSOUT allocation parameter COPIES.

DDName Options:

Applicable only when print destination type is "DDName", DDName options specify the DDName of a pre-allocated data set only.

DDName>
Identifies an existing DDName.

Formatted Report Utility (=11.2)

The Formatted Report Utility panel (ZZSGRPT0) is an interactive panel window, used to generate a printable report from a dataset containing records mapped by a structure (e.g. COBOL/PL1 copybook).

The layout of your desired report should be specified using control statements saved in the Report Definition dataset.

The report produced will consist mainly of data extracted from a list of data fields from one (or more) Record-Type as defined by the mapping structure (copybook). The report can reference fields from both Primary (Base) and Secondary segments.

A user definable heading will be printed at the top of each page, followed by user definable columns heading for each selected field.

By default grand totals will automatically be printed for any selected field containing integer data, and sub-totals will also be printed if a sort/control-break has been requested.

See Formatted Report Definition Control Statements for full details of supported report definition control statements.

Primary command REPORT provides a Command Line Interface (CLI) to the options on this panel.

The panel may be opened via the following:

- Select option 2. 'Report' from the Print/Report option menu (=11).

Sample Report Definition

```
<----1----+----2----+----3----+----4----+----5----+----6
TITLE:    *** Sample Music Collection ***

COLUMNS:
   TRACK.TRACK-NUM      'Trk'       3 R
   TRACK.ALBUM          ''         35
   TRACK.NAME           ''         20
   TRACK.FILE-SIZE      ''         15 R
*  TRACK.DATE-ADDED     'Added'    10

FILTER:
   TRACK.ARTIST << 'Springsteen'

SORT:
   TRACK.ALBUM
   TRACK.TRACK-NUM

BREAK:
   TRACK.ALBUM

TOTALS:
   TRACK.FILE-SIZE
* * * End of File * * *
```

Figure 191. Report Control Statements.

Sample Report Output

```
<---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
12019/04/17 14:48           *** Sample Music Collection ***         PAGE    1

 Trk ALBUM                               NAME                 FILE-SIZE
 --- ----------------------------------- -------------------- ---------------

  13 Bruce Springsteen & The E Street Ba The Rising                 9914746
                                                               ---------------
                                                                    9914746

   1 Bruce Springsteen In Concert - MTV  Red Headed Woman           6020163
   2 Bruce Springsteen In Concert - MTV  Better Days                9298082
   3 Bruce Springsteen In Concert - MTV  Atlantic City             11559071
   4 Bruce Springsteen In Concert - MTV  Darkness On the Edge       9686903
   5 Bruce Springsteen In Concert - MTV  Man's Job                 11769432
   6 Bruce Springsteen In Concert - MTV  Human Touch               15228420
   7 Bruce Springsteen In Concert - MTV  Lucky Town                10401995
------------------------- 149 line(s) not displayed -------------------------
```

```
12019/04/17 14:48          *** Sample Music Collection ***          PAGE    4

 Trk ALBUM                                NAME                  FILE-SIZE
 --- ------------------------------------ --------------------- ---------------
  15 We Shall Overcome - The Seeger Sess How Can I Keep from           4893983
  16 We Shall Overcome - The Seeger Sess How Can a Poor Man S          7049339
  17 We Shall Overcome - The Seeger Sess Bring 'Em Home                7552700
  18 We Shall Overcome - The Seeger Sess American Land                 9770156
                                                                     ---------------
                                                                       158461507


                                                                     ===============
                                                                      1059233137
                                                                     ===============
* * * End of File * * *
```

Figure 191. Report Output.

Panel Input Fields

By default, field entries are populated with arguments and options that were entered the last time it was used.

Report Definition:
Input fields which together identify a single, sequential or VSAM data set, GDG file generation, HFS file or PDS/PDSE library member that contains (or will contain) the Report Definition control statements.

Execute primary command **EDIT** (or **E**) to edit the report definition file.

DSN/Path>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

Dataset names beginning with "." (dot) will be treated as though they actually begin with a high level qualifier equal to the user's own userid.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of an existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Data File:
Input fields which together identify a single existing, sequential or VSAM data set, GDG file generation, HFS file or PDS/PDSE library member containing data records that are to be reported upon.

Execute primary command **INPUT** (or **I**) to browse the data file using the defined layout.

DSN/Path>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

Dataset names beginning with "." (dot) will be treated as though they actually begin with a high level qualifier equal to the user's own userid.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of an existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

This field may also be used to enter the **relative generation number** of a **GDG.** e.g. 0 or -10.

Record Selection:
Fields which together identify criteria by which a subset of records from the file are selected for the report.

Input Limit>
The maximum number of records that may be read from the Data File for potential selection.
Output Limit>
The maximum number of records that may be selected for reporting.

An input limit or output limit of "0" (zero) is not sensible and so is totally ignored.
Find String>
Specifies one or more strings to be located within each Data File record.

Note that this option is ignored if your report definition control statements include a FILTER section, which should be coded to handle all your selection requirements.

Multiple strings may be specified, each **separated by a comma**. A record containing any one of the strings will be selected. e.g.

```
Find String> SYS1.MACLIB, SYS1.MIGLIB, SYS1.MODGEN, SYS1.MSGEN
```

Each string may be specified:

- Using an unquoted character literal. e.g. ABC will match any record containing 'ABC' (case-insensitive).
- Using a quoted character literal. e.g. 'ABC ' will match any record containing 'ABC ' (case-insensitive).
- Using a quoted character literal prefixed with "C". e.g. C'aBc ' will match any record containing 'aBc ' (case-sensitive).
- Using a quoted hex literal prefixed with "X". e.g. X'81C2C340' will match any record containing 'aBc ' (case-sensitive).
- Using the single-character wildcard '%' (percent). e.g. 'ABC%DEF' will match any record containing 'ABC' followed by any other single character followed by 'DEF'. e.g.

  ```
  Find String> SYS1.M%%LIB
  ```

- Using the multiple-character wildcard '*' (asterisk). e.g. 'ABC*DEF' will match any record containing 'ABC' followed by any number of other characters followed by 'DEF'. e.g.

  ```
  Find String> SYS1.M*LIB
  ```

The panel entry field for this option is 45 bytes in length. To specify values longer than this, first type **"EXPAND"** on the command line, then place your cursor in the **Find String>** entry field before pressing ENTER. A "Text-Edit" window will be displayed allowing you to enter long values over multiple lines.

Options:

Run Type> B | C | F

B indicates that JCL should be produced for submission to batch.

C indicates that command line interface should be produced. The REPORT primary command is displayed in a Text-Edit window in a format suitable for execution using the ACTION key (Shift-F4) ready to be copied into your HOME file (=4).

F indicates that immediate foreground execution is required as soon as the **ENTER** key is pressed.

In this case the report output will be collected in-storage and displayed in a Text-Edit window. The output will not be saved to disk but the user may enter the CREATE or REPLACE commands to do so.

If the expected report output is larger than your available foreground region then you should choose the B option to generate a batch job which will write the report to the DD name **SDEPRINT**.

Output Type> P=Print | C=CSV | J=JSON | X=XML | B=Browse

P requests standard **Print** output with page headings, optional sort, control breaks and totals/sub-totals.

C requests **Comma Separated Variable (CSV)** output suitable for loading into various external formats such as **DB2**. The first row will contain the column headings as defined by your report definition file.
Defined control breaks and totals/sub-totals are all ignored for **CSV** output.

J requests **Java Script Object Notation (JSON)** output suitable for loading into various external formats such as a web page.
Defined control breaks and totals/sub-totals are all ignored for **JSON** output.

X requests **eXtensible Markup Language (XML)** output suitable for loading into various external formats including directly into **Microsoft Excel.**
Defined control breaks and totals/sub-totals are all ignored for **XML** output.

B requests an online **Browse** session is started with the format of the displayed data controlled by the report definition.
Defined sort, control breaks and totals/sub-totals are all ignored for **BROWSE** output.

Page Depth> nnn

Specifies the number of lines printed per page.

If left blank (or specified as zero) then the user's own prevailing Data-Edit PAGEDEPTH value will be used.

Type **"SD Q PAGEDEPTH"** to query your current Data-Edit pagedepth value.

Type **"SD SET PAGEDEPTH nnn"** to set your current Data-Edit pagedepth value.

DB2 Utilities

SELCOPY/i DB2 facilities are incorporated within the SELCOPY Product Suite base product and do not require any additional licensing over and above the SELCOPY product key.

SELCOPY/i DB2 provides a suite of tools to assist working with DB2 data and objects. Command syntax and panels provide functions that include:

- Edit and Browse of DB2 table data in a SELCOPY/i Structured Data Edit (SDE) view.
- List, Create, Drop and Alter of DB2 objects. (Tables, Indexes, etc.)
- Interactive Execution of DB2 commands and SQL statements.
- Generation of JCL for SQL statement execution and stand-alone DB2 utilities.

SELCOPY/i users may connect to any local DB2 subsystem for which the SELCOPY/i DB2 plan has been bound and the user granted EXECUTE authority. For successful operation, users must also be granted SELECT access to the subsystem's DB2 catalog tables and, if configured, READ access to the SELCOPYI.DB2 SAF resource. See the "*SELCOPY Product Suite Customisation Guide*" for details on enabling SELCOPY/i DB2.

Multiple connections to one or more local DB2 subsystems may exist in the same SELCOPY/i session. Each connection has its own audit setting (on or off) and audit log data set. The DB2 susbsystem name is displayed in the title bar of any SELCOPY/i DB2 window (panel or SDE edit view) to which that window relates.

All DB2 related features may be accessed via the suite of SELCOPY/i DB2 interactive panel windows (window class WINWIPO0) that are invoked by selecting the "DB2" drop-down menu item of the "File" main menu, or by executing the DB2 CLI command. Note that SELCOPY/i panels are window objects within SELCOPY/i and should not be confused with ISPF panels.

DB2 Primary Option Menu

The DB2 Primary Option Menu panel (ZZS2PRIM) is an interactive panel window providing the entry point to all SELCOPYi DB2 panels, encompassing the DB2 functionality available in SELCOPYi.

This DB2 panel is the first in a hierarchical chain of DB2 panels (menus, functions, lists) that are opened thereafter. Select an item from the menu of DB2 related tasks to open the relevant DB2 task panels. Note that, although part of the same DB2 panel hierarchy, these panels are not owned by DB2 primary options menu panel. Window Focus may be returned to the primary options panel to select another branch of DB2 task panels in the same hierarchy without having to exit existing DB2 panels.

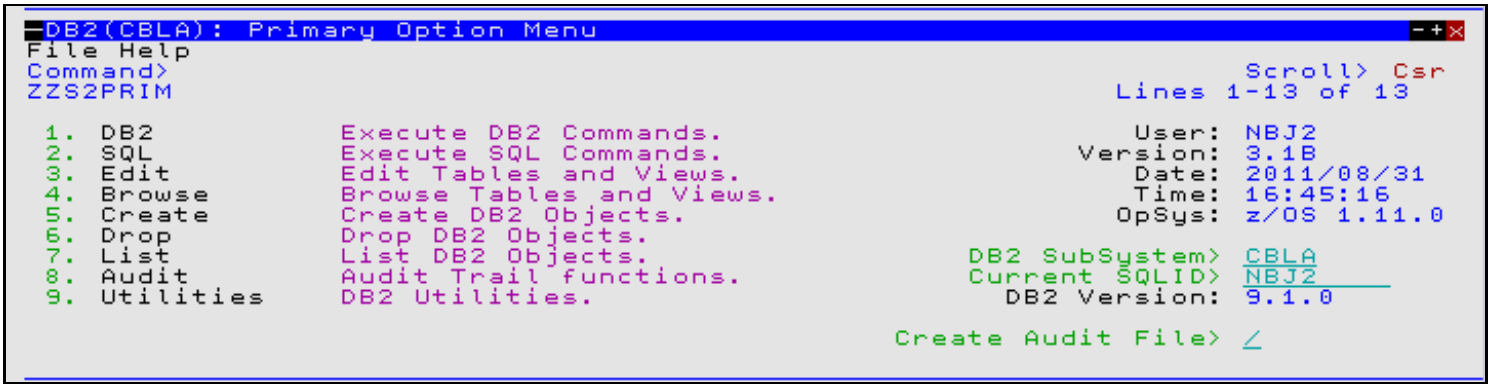


Figure 192. DB2 Primary Option Menu Panel.

On selecting an item from this menu panel, or if a PFKey/<Enter> is actioned, an attempt is made to connect to the DB2 subsystem specified in the DB2 SubSystem field. If the value in this field is subsequently changed, then a new DB2 hierarchy of DB2 panels is started and a connection is made to the new DB2 subsystem. If no other panel exists in the DB2 hierarchy belonging to the previously specified DB2 subsystem, then connection to that subsystem is dropped before the new connection is made.

The name of the connected DB2 subsystem is displayed in parentheses in the window title bar of the DB2 primary options menu and therafter in the title bars of DB2 panels and SDE edit views opened in the DB2 panel hierarchy.

Although DB2 table SDE edit views may be opened via the DB2 panels, these are not included as part of the DB2 panel hierarchy. This is because a separate DB2 connection is performed and a separate audit file is maintained for each edited results table allowing updates made to a DB2 table to be isolated from other DB2 tasks (including updates made to other DB2 tables.)

To work with multiple DB2 subsystems concurrently, the command DB2 SSN=*name* may be executed to open multiple DB2 primary option menu panels each connected to different DB2 subsystems. Each invocation of a new DB2 primary option menu panel constitutes another hierarchy of DB2 related panels.

Note that if a DB2 primary option menu panel is already open for the SSN=*name*, then no new connection is made and the DB2 panel hierarchy that exists for that SSN is used instead.

Menu Bar Items

File
The File drop-down menu contains the single item, Exit, to close the panel.

SwapList
If SELCOPYi is operating within an ISPF split screen, opens the ISPF task list of active ISPF logical sessions.

Window
Opens the Window List window containing a selectable list of all open windows in the SELCOPYi session.

Help
Open the general help for the Primary Option menu panel.

QuickRef
Open the SELCOPYi Quick Reference.

Options

0 Settings	Set DB2 options
1 DB2	Execute DB2 Commands
2 SQL	Execute SQL Statements
3 Edit	Edit Tables and Views
4 Browse	Browse Tables and Views
5 Create	Create DB2 Objects
6 Drop	Drop DB2 Objects
7 List	List DB2 Objects
8 Audit	Audit Trail Functions
9 Compare	Compare DB2 Tables
10 Rename	Rename DB2 Tables and Indexes
11 Structure	Create/Edit a DB2 Table SDO
12 Utilities	DB2 utility job generation
13 Report	Create a Report from a DB2 Table or from SQL
T Training	Setup DB2 Training Material

Panel Input/Output Fields

User:
An output field displaying the user's userid.

Version:
An output field displaying the version of SELCOPYi.

Date:
An output field displaying the current date.

Time:
An output field displaying the current time.

OpSys:
An output field displaying the operating system release.

DB2 SubSystem>
An input field identifying the DB2 subsystem to which a connection will be made. SELCOPYi DB2 functions and panels will operate on objects defined in this subsystem. A connection will not be attempted until <Enter> is hit or the panel window is repainted (e.g. as a result of actioning a PFKey.)
The ZZS2PRIM internal field name for DB2 Subsystem is **SSN**.

Current SQLID>
An input field that sets the SQL authorisation ID for this particular hierarchy of DB2 panels' access to DB2. This value is the initial value of the DB2 special register CURRENT SQLID that is provided on the DB2 connection. See the IBM publication *"DB2 SQL Reference"* for further information on CURRENT SQLID and its usage with dynamically prepared SQL statements.

Changing the contents of this input field following connection will change the CURRENT SQLID for subsequent functions executed via panels in this DB2 panel hierarchy. To do this, the user requires appropriate DB2 authorisation to use the new value. (See the IBM publication *"DB2 Administration Guide"* for further information.)

The default value for Current SQLID is the user's TSO or SELCOPYi VTAM logon id.
The ZZS2PRIM internal field name for Current SQLID is **SQLID**.

DB2 Version:
An output field displaying the version of DB2 for the connected DB2 subsystem. Note that DB2 version 9 is the earliest release supported by SELCOPYi DB2.

Create Audit File>
An option check box that indicates that SELCOPYi DB2 auditing will occur for actions performed in this DB2 panel hierarchy. Note that auditing of DB2 table edit views is managed separately and is not affected by this check box setting.

If Create Audit File is selected, an audit log file will be allocated immediately before attempting to connect to the DB2 subsystem and closed when the connection is dropped.

See Audit Trail Functions for details of SELCOPYi DB2 auditing.

Execute DB2 Commands

The Execute DB2 Commands panel (ZZS2XDB2) is an interactive panel window, opened on selection of option 1. in the DB2 Primary options menu or on execution of the DCMD command.

This panel provides facility to execute DB2 and related commands to the connected DB2 subsystem and view the command output. (See the IBM publication *"DB2 Command Reference"* for further information.) The individual user must have the required level of authority in order to successfully execute a DB2 command.

Output from the DB2 command execution is displayed in a scrollable list window within the panel. The list consists of a single column with header "Output".

The Output data reports the DB2 command executed; the return and reason code received on execution of the DB2 command; the instrumentation facility interface (IFI) return and reason code; and the number of bytes returned/not returned. If number of bytes **not** returned is greater than zero, then this value indicates the amount of additional buffer space required to display the complete output from the command.

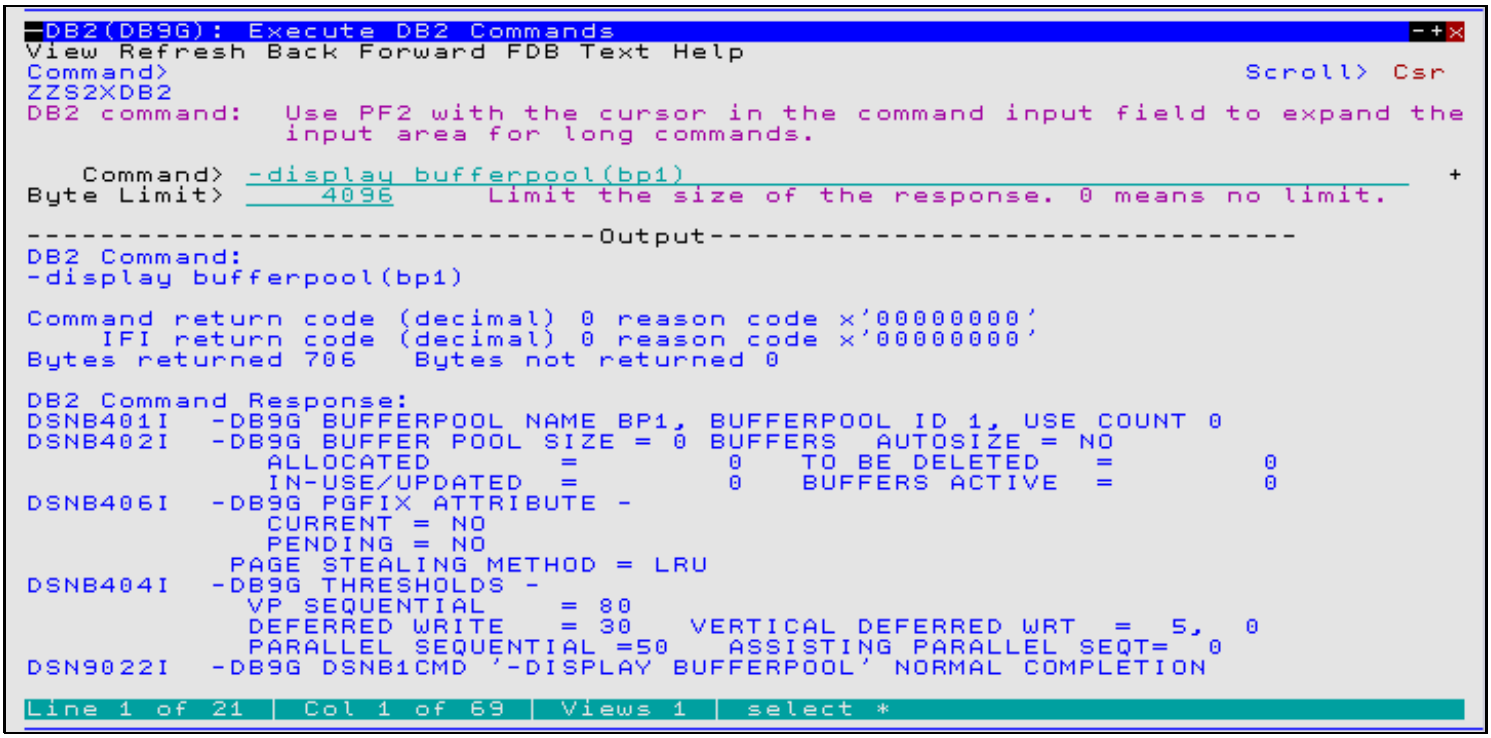


Figure 193. Execute DB2 Commands Panel.

Menu Bar Items

See List Window menu for description of menu bar items.

Field Entries

Command>
An input field in which the DB2 command is entered.

The ZZS2XDB2 internal field name for Command is **DB2CMD**.

Byte Limit>
An input field defining the maximum size of the DB2 command output data buffer.

If set to 0 (zero), then there is no limit to the output buffer size.

Where the length of data returned by the command exceeds the output buffer size, then error message ZZSX016W is returned indicating the number of bytes of output data returned, and number of bytes not returned by the command.

The ZZS2XDB2 internal field name for Byte Limit is **LIMIT**.

Execute SQL Statements

The Execute SQL Statements Options panel (ZZS2XSQM) is an interactive panel window, opened on selection of option 2. in the DB2 Primary options menu.

This panel allows the user to select the SELCOPY/i DB2 utility and method to be used to execute one or more DB2 SQL statements, by entering the relevant option number or by positioning the cursor on the required option and pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Note that successful execution of SQL statements is dependent upon the user's level of authority or granted privileges. Please refer to the relevant edition of the *"z/OS SQL Reference"*.

Menu Bar Items

File

The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Help

Open the general help for the Execute SQL option menu panel.

Options

1 ExecSQL	Execute SQL using file input and output
2 dSQL	Execute SQL using panel-field input and panel-list output
3 SQL	As for dSQL but execute using SELCOPY/batch plan

ExecSQL

The Execute SQL statements panel (ZZS2SQL0) is an interactive panel window, opened on selection of option 1. in the Execute SQL options menu or on execution of the EXECSQL command.

This panel provides facility to execute SQL statements from a data set or library member in a specified DB2 subsystem and direct the SQL output to a data set as either plain text or formatted output. (See the IBM publication *"SQL Reference"* for further information on SQL statement syntax.) The user must have the required level of authority in order to successfully execute a SQL statement.

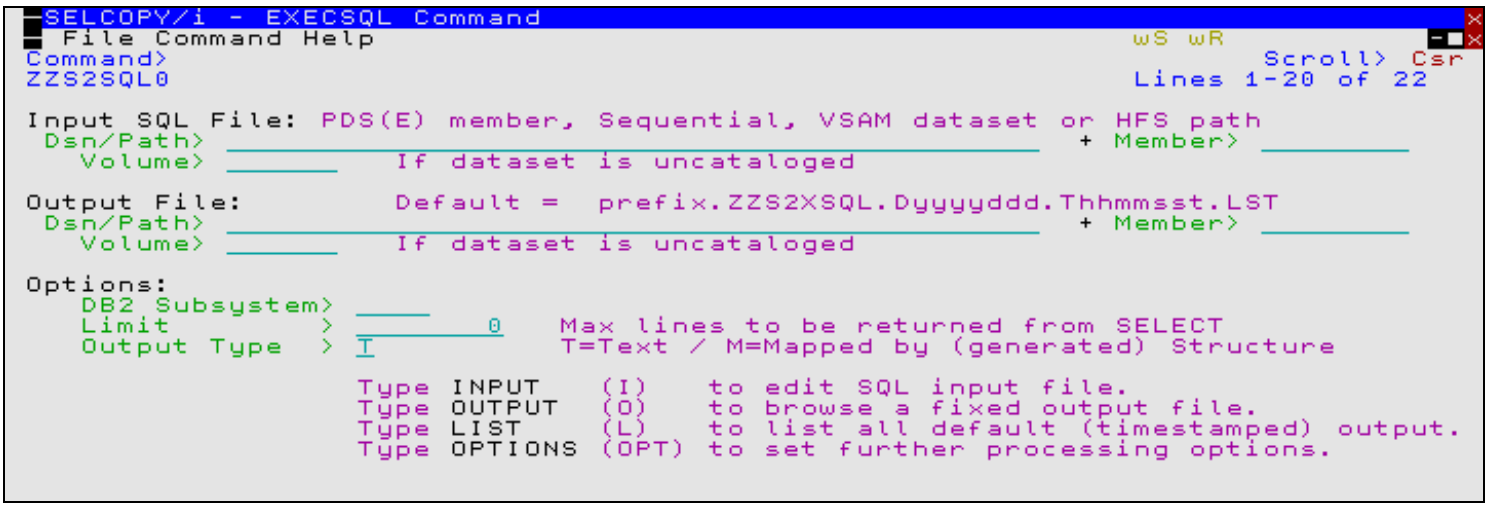


Figure 194. ExecSQL Command panel.

Menu Bar Items

File

The File drop-down menu contains the single item "Exit" which simply closes the panel window. Note that, unlike CANCEL, CLOSE will save field values entered in the panel so that they may be redisplayed the next time the panel is opened.

Command

Generate the EXECSQL primary command syntax and display it in a temporary CMX file text edit view. This command may be executed using ACTION point-and-shoot execution <F16> or copied into the user's HOME file and saved for future execution.

The user has the opportunity to edit the command prior to its execution and/or copying it to the home (CMX) command centre for future reference and re-execution.

Help
Display help for this panel view.

Panel Input Fields

Input SQL File:
Input fields which together identify a single existing, sequential or VSAM data set, GDG file generation, HFS file or PDS/PDSE library member from which SQL statements are to be executed.

DSN/Path>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of an existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Volume>
Specifies the name of the input data set volume. This is required only if input is from an uncataloged data set.

Output File:
Input fields which together identify the name of the a new or existing, sequential data set, HFS file or PDS/PDSE library member to which EXECSQL output will be written.

DSN/Path>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Default DSN of a generated output file is *userpfx*.ZZS2XSQL.D*yyyyddd*.T*hhmmsst*.LST.

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Volume>
Specifies the name of the output data set volume. This is required only if output is to an uncataloged data set.

Options:
Options that determine the execution parameters.

DB2 Subsystem>
Identifies the DB2 subsystem to which a connection will be made. For successful execution, the supplied DB2 plan for SELCOPY/i dynamic SQL, CBLPLAN1, must have been bound to this subsystem and execute privilages granted.

Limit>
Limits the number of rows fetched by an SQL query (SELECT) statement.

This limit is imposed by SELCOPY/i on all SELECT statements as rows are fetched. If a *fetch-first-clause* is specified on a SELECT, then a number of rows will be fetched which is the lesser of the limit value and the *fetch-first-clause* value.

Note that, if all selected rows are fetched, then SQLCODE 100 is returned indicating that a FETCH statement was executed when the cursor was positioned after the last row of the result table. If the limit threshold is reached before this occurs, then the execution of SELECT will end with SQLCODE 0.

Output Type>
Specifies the format of the output report. "T" indicates plain text, "M" indicates formatted records that are mapped by record-type structures in a generated SDO. Output is automatically displayed in an SDE Browse view when execution completes.

Output Options Panel Fields

Opened on entering primary command, OPTIONS (min abbrev OPT), the Output Options panel view allows specification of output record and file attributes.

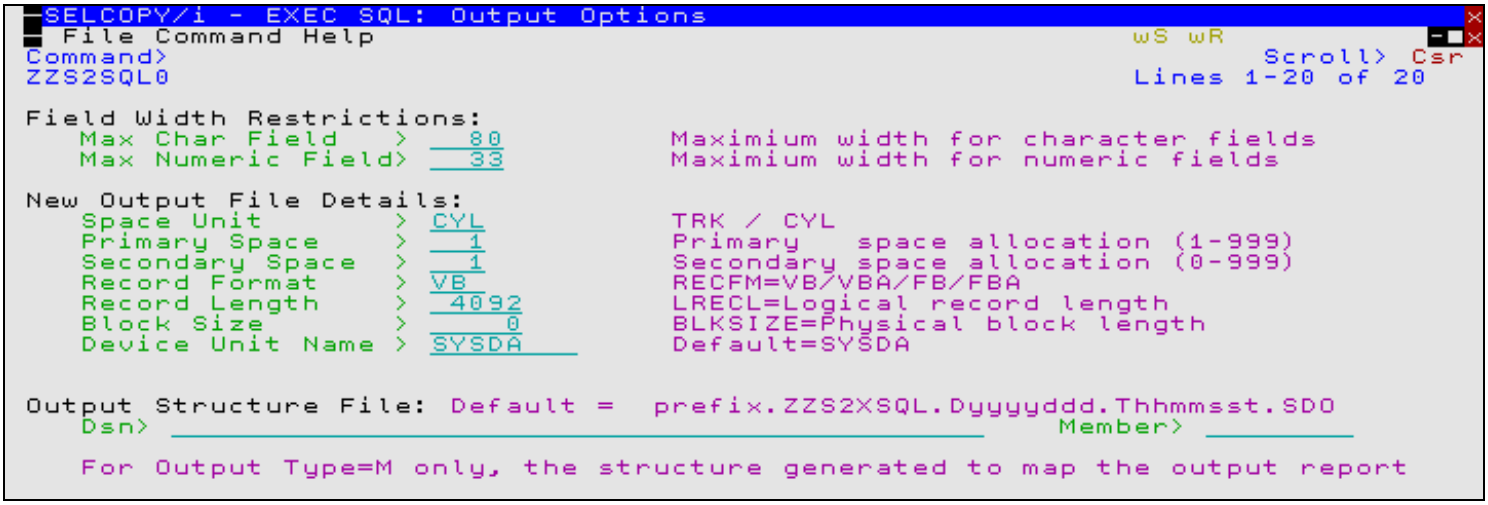


Figure 195. ExecSQL Output Options panel.

Field Width Restrictions:

Input fields that control the maximum width of columns displayed by an SQL SELECT query. Restricting the width of these column fields restrict the overall length of records written to the output file.

Max Char Field>

The maximum width of data displayed by selected columns of character type (CHAR, VARCHAR, GRAPHIC, etc.)

Max Numeric Field>

The maximum width of data displayed by selected columns of numeric type (INTEGER, DECIMAL, FLOAT, etc.)

New Output File Details:

Input fields which define attributes used to allocate a new output data set.

Space Unit>

Identifies the allocation unit CYL (cylinders) or TRK (tracks).

Primary Space>

The primary allocation number of allocation units. Default value is 1.

Secondary Space>

The secondary allocation number of allocation units. A value of 0 (zero) indicates no secondary allocations may occur. Default value is 1.

Record Format>

Data set record format. Supported options are VB, VBA, FB and FBA.

Block Size>

Data set physical record (block) size. Default is 0 (zero) implying SMS determined BLKSIZE.

Device Unit Name>

Device unit from which space will be allocated to the data set. Default is an esoteric unit of SYSDA.

Output Structure File:

Applicable only to formatted (mapped) output when an output file name is specified. These input fields constitute the name of the SDO structure file. The SDO file may be a cataloged sequential data set or library member.

DSN>

Identifies the fully qualified data set name of a new or existing sequential file or PDS/PDSE library.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Default DSN of a generated SDO is *userpfx*.ZZS2XSQL.D*yyyyddd*.T*hhmmsst*.SDO.

Member>

If the DSN> field contains the DSN of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Primary Commands

The following primary commands are supported.

INPUT	Edit the input file.
OUTPUT	Edit the output file.
LIST	List all output files generated with the default output data set name.
OPTIONS	Display the Output Options panel view.

dSQL

The Execute SQL statements panel (ZZS2XSQL) is an interactive panel window, opened on selection of option 2. in the Execute SQL options menu or on execution of the DSQL command.

This panel provides facility to execute SQL statements to the connected DB2 subsystem and view the statement output. (See the IBM publication *"SQL Reference"* for further information.) The individual user must have the required level of authority in order to successfully execute a SQL statement.

Note that, whereas the dSQL panel uses the SELCOPY/i DB2 plan (default CBLPLAN1) to execute the prepared SQL statement, the DB2 Dynamic SQL window may be used to perform the same operation using the SELCOPY batch program DB2 plan (default CBLPLAN0). This alternative method of executing DB2 SQL statements has the additional benefit of being able to process one or more SQL statements provided via an input contol file.

Output from the SQL statement execution is displayed in a scrollable list window within the panel.

The format of the list output depends on the type of SQL statement executed. Successful execution of SQL query (SELECT) statements will display the selected results table columns and rows. The list column headers are the results table column names.

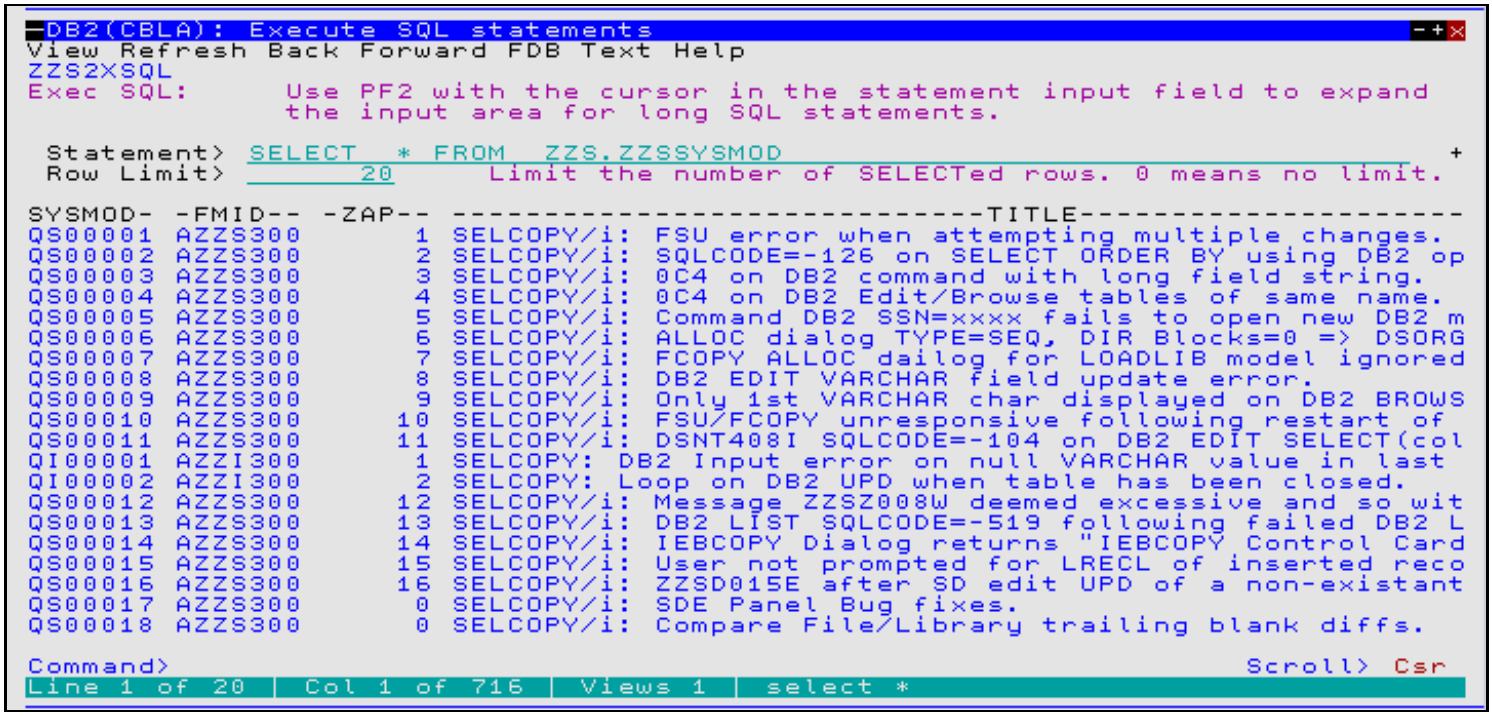


Figure 196. dSQL Panel - SQL Query.

All other SQL statements display a report of SQL messages detailing successful or unsuccessful execution. The message text output are rows of a list with the single column header, "Result".

See the IBM publications *"DB2 Messages"* and *"DB2 Codes"* for DSN prefixed messages and SQL error codes.

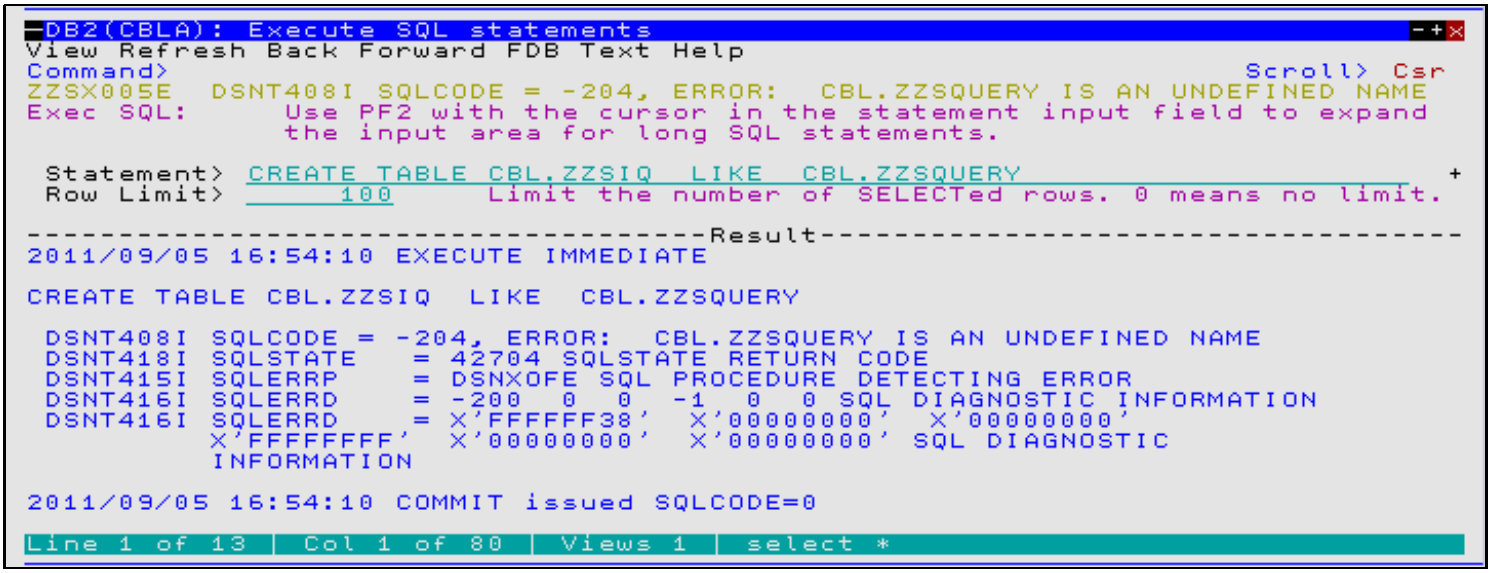


Figure 197. dSQL Panel.

Menu Bar Items

See List Window menu for description of menu bar items.

Field Entries

`Statement>`
: An input field in which the SQL statement is entered.

  The ZZS2XSQL internal field name for Statement **SQLCMD**.

`Row Limit>`
: Limit the number of rows to selected by an SQL query (SELECT) statement. Once the limit threshold has been reached, no further attempt is made to retrieve selected rows of data.

  If set to 0 (zero), then there is no limit to the number of rows retrieved from the results table.

  The ZZS2XSQL internal field name for Row Limit is **LIMIT**.

SQL

The DB2 Dynamic SQL window is a list window that uses the **SELCOPY** batch program DB2 interface to execute dynamic SQL statements and then display the output messages and/or result table rows in the list client area. This window may be opened opened via the following:

- Select option 3. SQL from the Execute SQL Statements options menu.
- Execute primary command SQL.

The DB2 Dynamic SQL capability is available only to MVS sites where SELCOPY is installed. The SELCOPY DB2 interface, **SELCOPQL** load module is used to pass SQL statements to the DB2 data base and so must also be available in the SELCOPY load library.

On startup, the Dynamic SQL window connects the user to the DB2 subsystem using the DB2 subsystem name and plan specified in the panel fields. By default, these are values assigned in the **CBLNAME** options module.

The DB2 Dynamic SQL window should only be used to test SQL statements that will form part of a SELCOPY batch program to determine the results. For general execution of SQL statements the ExecSQL or dSQL panels should be used.

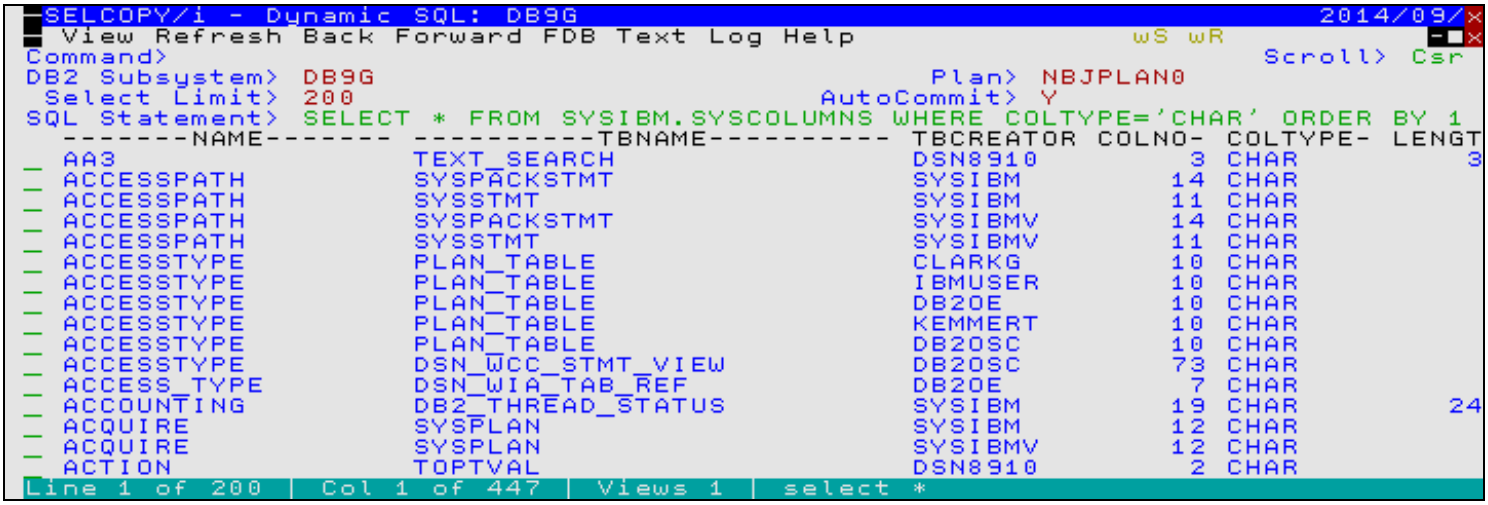


Figure 198. SELCOPY SQL window.

The contents of the display area may be edited in a new file using the text editor by selecting **Edit** from the window menu.

Select **Log** from the window menu to open the SQL Log output which displays the diagnosis information and SQL return codes passed from the DB2 sub-system.

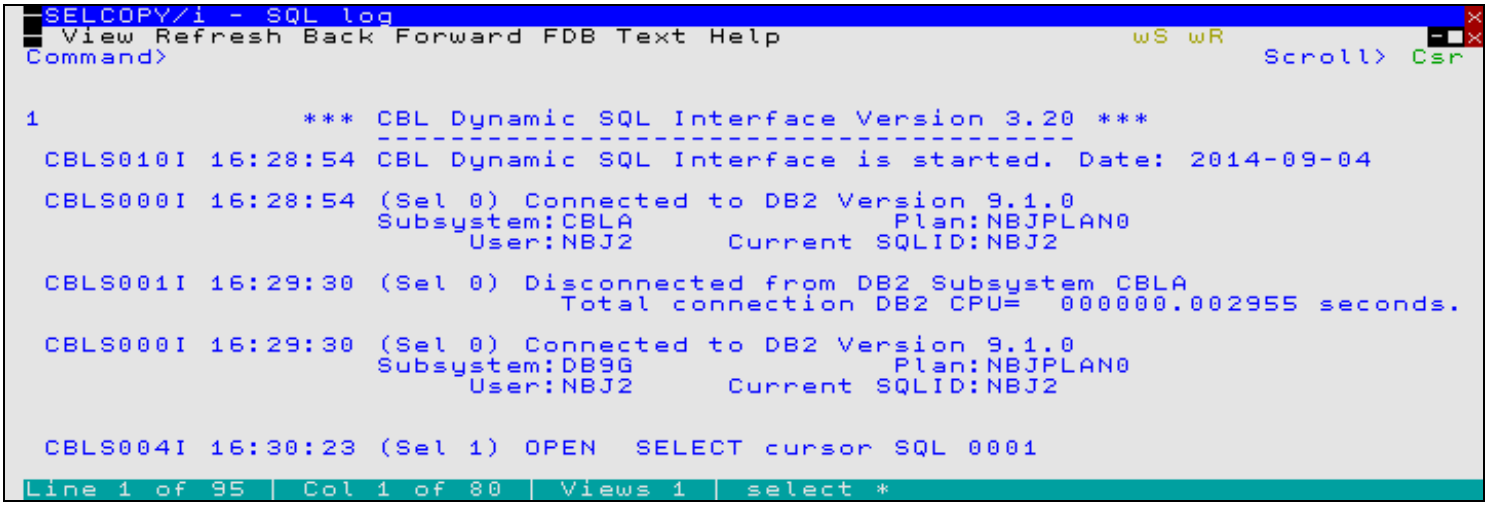


Figure 199. DB2 Dynamic SQL LOG window.

Select **FDB** from the window menu to open the Field Descriptor Block which provides detailed information on each field displayed by an SQL SELECT statement.

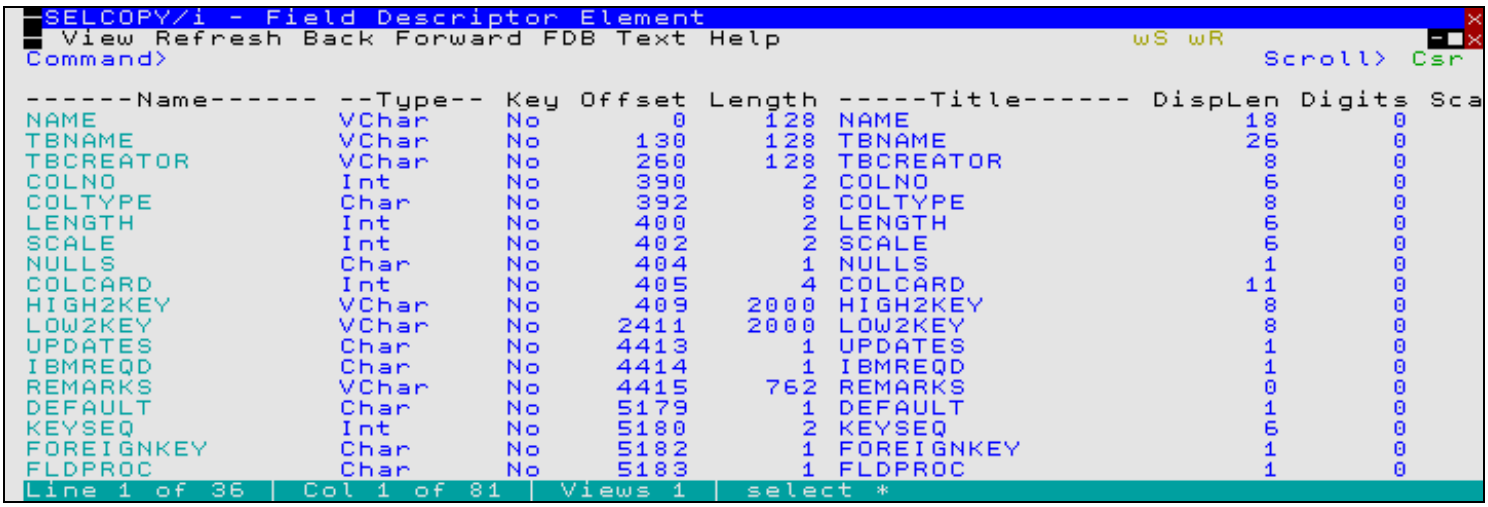


Figure 200. DB2 SQL FDB Window.

Field Entries

DB2 Subsystem>
Specify the DB2 sub-system name to be the target of the CONNECT.
Changing the Subsystem name will cause CBLi to disconnect the user from the current subsystem and reconnect to the new subsystem.

Default is that defined by the CBLiINI option, DB2.SSN, otherwise the sub-system name specified in the DB2SubSys field of the CBLNAME load module is used.

Plan>
Specify the SELCOPY (SELCOPQL) DB2 plan name which has been bound to the DB2 sub-system.
This is the name assigned to the SELCOPY plan during the BIND to the DB2 subsystem.

Default is that defined by the CBLiINI option, DB2.Plan, otherwise the plan name specified in the DB2Plan field of the CBLNAME load module is used.

Select Limit>
Limit the number of rows to be displayed in the Dynamic SQL window following a SELECT transaction. Once the limit threshold has been reached, a pop-up message window is displayed and no further attempt is made to retrieve selected rows of data.
The *n_rows* value is placed in the "Select Limit>" field of the Dynamic SQL window.
The default limit is that defined by the CBLiINI option, DB2.SelectLimit, otherwise no limit is implied.

AutoCommit>
Determine whether a COMMIT is to be automatically issued following every transaction (AutoCommit). If COMMIT=NO, then the user should issue COMMIT manually to commit any changes made to the data. A commit is executed automatically when the Dynamic SQL window is closed, regardless of the AutoCommit field setting.
The commit value is reflected in the "AutoCommit>" field of the Dynamic SQL window.
The default is YES.

SQL Statement>
Specify valid SQL statement syntax to be executed either directly or via an input control file.

If an input control file is specified, then the input fileid must be prefixed by "<" (less than). e.g.

```
< NBJ.SQL.CTL(SYSTABS)
```

The resulting data or messages are displayed in the window display area.

Edit Tables and Views

DB2 table data may be edited using the SELCOPYi Data Editor. A DB2 result table is generated based on an SQL query and rows are fetched and displayed in a structured format as for dataset records displayed using a COBOL/PL1/HLASM copybook structure.

Unless a SELCOPYi DB2 structure (SDO) is specified, SELCOPYi will generate a temporary structure using the DB2 SQLDA chain for the results table columns.

All standard Data Editor features (including FIND, CHANGE, WHERE and LOCATE) are available for DB2 table edit. General features of the Data Editor and those specific to DB2 table edit views are documented in detail in the SELCOPYi Structured Data Editor (SDE) manual.

Differences between the standard Data Editor view for datasets and that displayed for DB2 tables are as follows:

1. Abbreviated forms of the standard DB2 datatypes are displayed in the field type header line instead of the SDE field datatype, position and length. Display of this header line is controlled using SET TYPE ON/OFF.
2. An additional SQLCode record information field for DB2 which may be displayed and hidden using SET RECINFO SQLCODE/NOSQLCODE or selecting option 2 in the DB2 Edit Utilities manu (<PF4>).

Edit DB2 Table Panel

The Edit DB2 Table panel (ZZS2EDIT) allows the user to configure DB2 table edit and data management options prior to loading rows from the DB2 results table and presenting the column data in an Data Editor view.

It is an interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select "Edit", option 3, from the SELCOPYi DB2 primary option menu. (DB2 3)
- Type the "E" line-command against an entry in a DB2 Tables List.

Input fields in each of the DB2 Edit panel views reflects the DB2 input parameters supported by its equivalent Data Editor line command, EDIT.

Having configured the input fields, press <Enter> to execute the edit. A new connection is made to the DB2 subsystem specified on the DB2 Primary option menu panel and, optionally, a new audit log data set is allocated. This isolates actions specific to that DB2 object edit from all other DB2 actions and object editing.

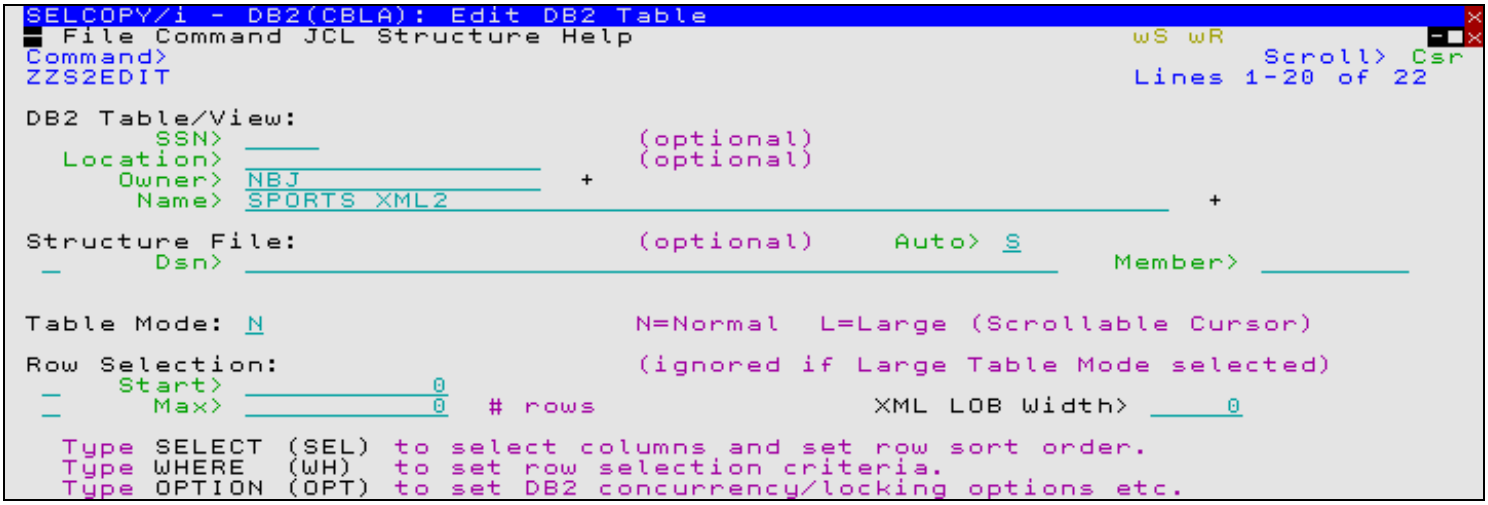


Figure 201. DB2: Edit DB2 Table panel.

Menu Bar Items

File
The File rename-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Command
The "Command" menu bar item will display in a text edit view the SDE EDIT command generated for the selected input field values. The generated command may then be executed by placing the cursor and pressing the "ACTION" key, or saved in the user's HOME file.

JCL
The "JCL" menu bar item will display in a text edit view a batch job containing the SDE EDIT command generated for the selected input field values. Further edit commands may be added (each separated by semi-colon) in order to run useful edit features in batch. If changes are to be made, the sequence should end with the "SAVE" command to avoid the loss of

valuable updates.

Structure
Open the Create DB2 Table Edit Structure panel to edit or create a permanent SELCOPYi DB2 structure that may be used in the Edit DB2 table panel.

Help
Open the general help for the Edit DB2 table panel.

Panel Fields - Edit DB2 Table

DB2 Base Table:

SSN>
The DB2 subsytem on which the table to be edited is located.

If left blank then the current subsystem (identified in the window title bar) will be used.

Location>
The server location of the table to be edited . If left blank, the local server for the connected DB2 sub-system is used.

Owner>
The owner (schema) of the table to be edited.
A table selection list will be provided if wildcards (*) are supplied.

Name>
The name of the table to be edited.
A table selection list will be provided if wildcards (*) are supplied.

Structure File:
If activated, defines fields which together specify a structure file used to map the table data and specify various options for the edit session. If no structure is specified then SELCOPYi will generate one using the DB2 SQLDA chain for the specified results table columns.

Dsn>
Identifies the fully qualified data set name of an existing sequential data set or PDS/PDSE library. The dataset must containing an existing DB2 SDO structure.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must a specify a member name.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Auto>
This option controls the level at which the association of structure to DB2 table is applied

A selectable list of options will be presented if blank is entered.

D (Default)	Use current AUTOSTRUCT setting
Y (Yes)	Auto-struct applied and saved
N (No)	Auto-struct neither applied nor saved
A (Apply)	Auto-struct applied only
S (Save)	Auto-struct saved only

Table Mode:
The table edit mode is either Normal (N) or Large (L).

In Normal table edit mode, the entire result table will be loaded into the user's local Data Editor storage.

In Large table edit mode, a DB2 SENSITIVE STATIC scrollable cursor is used to fetch result table rows. This option is incompatible with the row selection options "Start" and "Max". See Using Scrollable Cursors for information on how the use of this option affects the edit session.

Use of DB2 scrollable cursors is disabled by default. They may be enabled by an authorised user by setting site-wide option option **DB2.SCROLL=YES** in the SELCOPYi System INI file.

Row Selection:

Start>
The start row within the result set.

Maxrt>
The maximum number of rows to be selected for edit.

XML LOB Width>

Specifies the number of bytes (*n_bytes*) of text, at the start of an XML or large object (LOB) column, to be displayed for all XML and LOB columns in the edited table view.

Use command XMLEDIT to open a new Text Editor view in order to edit and update the XML document text located within a specific row and XML column of the current DB2 data edit view.

Edit DB2 Table Options

The Edit DB2 Table Options panel view is displayed only if command **OPTION** is executed.

This panel view determines DB2 specific options used when accessing table data and also options used specifically by SELCOPYi DB2 table edit.

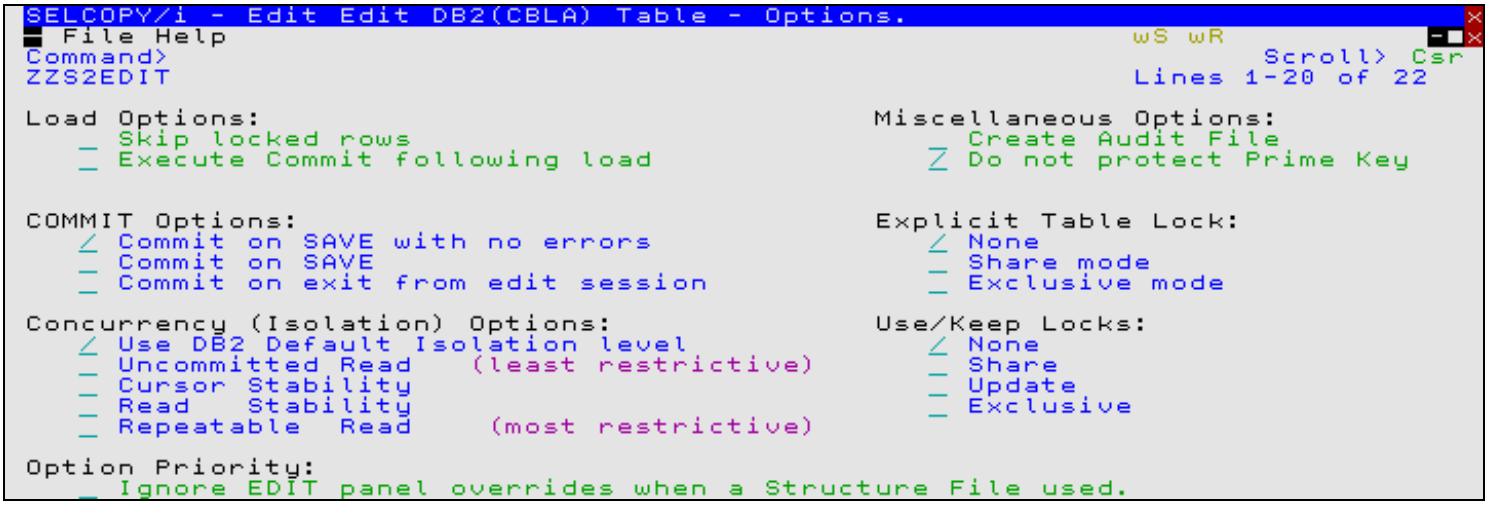


Figure 202. DB2: Edit DB2 Table Options panel.

Panel Fields - Edit DB2 Table Options

Load Options:

Skip Locked Rows
Ignored unless an isolation level of Cursor Stability (CS) or Read Stability (RS) is in effect, this option specifies that any selected rows that are already locked by another process should be skipped and not be included in the edit display. See "DB2 SQL Reference" for details on the SKIP LOCKED DATA clause.
Default is to allow display of locked rows whenever possible.

Execute Commit following Load
Perform a COMMIT following the initial load of rows to be edited, thus releasing all DB2 table locks performed during load of the data. This includes any explicit table locks applied via the LOCKTABLE parameter.
Default is not to perform a COMMIT following load of the table rows.

Miscellaneous Options:

Create Audit File
Open a new SELCOPYi DB2 audit data set to record changes to the edited table made during this edit session. See Audit Trail Functions for details.
Default is not to perform edit auditing.

Do not protect Prime Key
Specifies that data occupying columns that comprise the table's primary key is eligible for update.
Default is that this data is read-only.

COMMIT Options:

Commit on SAVE with no error
COMMIT only if SAVE is executed without errors.

Commit on SAVE
COMMIT on SAVE regardless of errors.

Commit on exit from edit session
COMMIT only on exit of the edit session.

Explicit Table Lock:

None
No explict table locking prior to load. (Recommended)

Share mode
Prevents anything other than read-only operations being performed on the table whilst it is being edited.

Exclusive mode
Prevents another process from performing any operations on the table whilst it is being edited, unless the process is running with an isolation level of Uncommitted Read (UR) in which case read-only (dirty read) operations may be performed.

Concurrency (Isolation) Options:

Use DB2 Default Isolation level
No "WITH" clause is added to the SQL statement used to fetch data.

Uncommitted Read (least restrictive)
"WITH UR" clause is added to the SQL statement used to fetch data.

Cursor Stability
"WITH CS" clause is added to the SQL statement used to fetch data.

Read Stability
"WITH RS" clause is added to the SQL statement used to fetch data.

Repeatable Read (most restrictive)
"WITH RR" clause is added to the SQL statement used to fetch data.

Use/Keep Locks:

None
No "KEEP" clause is added to the SQL statement used to fetch data.

Share
"KEEP SHR" clause is added to the SQL statement used to fetch data. Applicable only if either "Read Stability" (RS) or "Repeatable Read" (RR) isolation levels are selected.

Update
"KEEP UPDATE" clause is added to the SQL statement used to fetch data. Applicable only if either "Read Stability" (RS) or "Repeatable Read" (RR) isolation levels are selected.

Exclusive
"KEEP EXCLUSIVE" clause is added to the SQL statement used to fetch data. Applicable only if either "Read Stability" (RS) or "Repeatable Read" (RR) isolation levels are selected.

Option Priority:

Ignore EDIT panel overrides when a Structure File used.
Priority is given to options saved in the structure file.

Edit DB2 Table SQL Clauses

The Edit DB2 Table SQL Clauses panel view is displayed only if command **SQL** is executed.

This panel view displays and supports editing of the SQL query clauses generated by the DB2 Row Selection (WHERE) and DB2 Column Selection and Ordering (SELECT/SORT) panels.

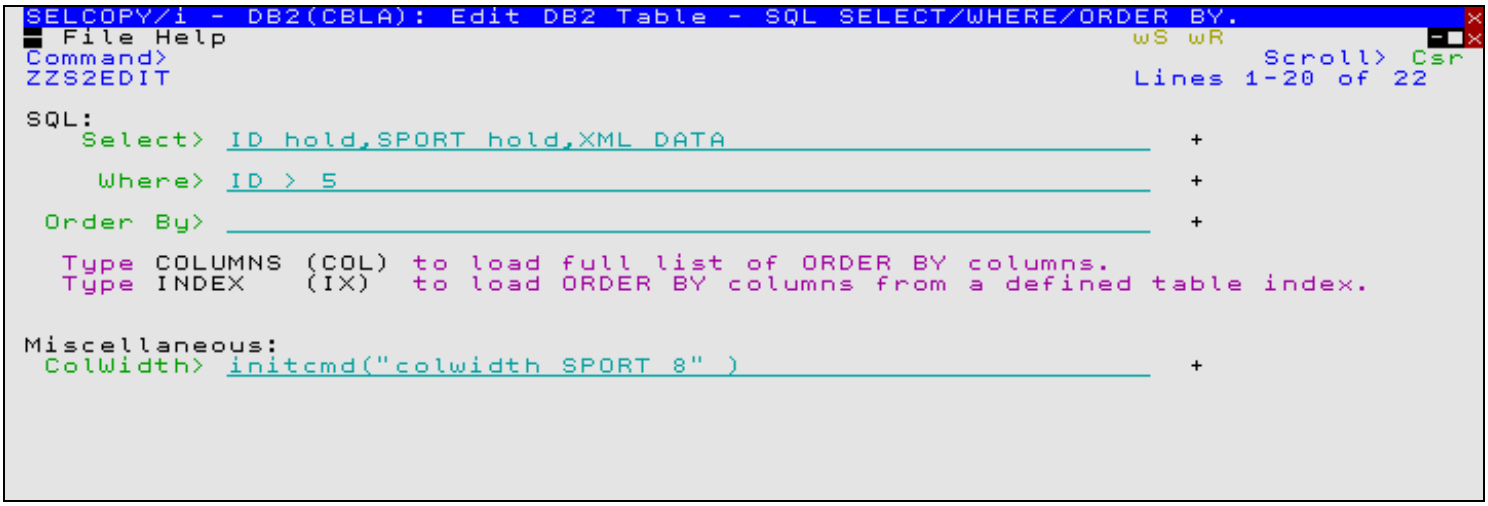


Figure 203. DB2: Edit DB2 Table SQL Clauses panel.

Panel Fields - Edit DB2 Table SQL CLauses

SQL:

Select>
The comma separated list of DB2 table column names that will be initially selected (e.g. visible in edit/browse). Although this field may be modified manually, it will be automatically updated (with all manually modification lost) whenever the SELECT primary command is used to select columns using the panel interface.

Where>
The "WHERE" clause passed as part of the SQL SELECT used to load data rows. Although this field may be modified manually, it will be automatically updated (with all manually modification lost) whenever the WHERE primary command is used to set row selection criteria using the panel interface.

Order By>
The "ORDER BY" clause passed as part of the SQL SELECT used to load data rows. Although this field may be modified manually, it will be automatically updated (with all manually modification lost) whenever the SELECT primary command is used to set row sort order using the panel interface.

Miscellaneous:

ColWidth>
A series of "OPTION(**COLWIDTH command**)" clauses passed to edit/browse to define the desired restricted visible width of long data columns. Although this field may be modified manually, it will be automatically updated (with all manually modification lost) whenever the SELECT primary command is used to set column width values using the panel interface.

Primary Commands

The following primary commands are supported by the DB2 Edit Tables and Views panels.

CMX

```
>>--+- CMX ----------------------+-----------------------------------><
    +- EDITCMX -------------------+
```

Applicable only to the **Edit DB2 Table** panel, CMX generates the SELCOPYi EDIT command and copies it to an in-storage output file in a format suitable for subsequent execution using the ACTION (default >F16>) facility.

CMX is assigned to <F5> by default.

COLUMNS

```
>>---- COLumns ------------------------------------------------------><
```

Applicable only to the **Edit DB2 Table SQL Clauses** panel, COLUMNS sets the "Order By" field to a comma separated list of all columns names in the specified table.

Each column name (including its trailing comma) is deliberately blank padded up to 50 bytes. This is designed so that, when "EXPAND" (F14) is executed with the cursor in the "Order By" field so that the field contents are displayed in a separate Text Editor view, each column appears on a new line. This make it easy to re-order the fields and delete unwanted sort field names.

INDEX

```
>>--+- INDEX --------------------+-----------------------------------><
    +- IX ----------------------+
```

Applicable only to the **Edit DB2 Table SQL Clauses** panel, INDEX displays a selectable list of DB2 indexes that apply to the specified base table.

On selection of an individual index, its column names are used to populate the "Order By" field.

JCL

```
>>--+- JCL ----------------------+-----------------------------------><
    +- EDITJCL -------------------+
```

Applicable only to the **Edit DB2 Table** panel, JCL generates the SELCOPYi EDIT command and copies it to an in-storage output file with JCL statements that execute the SDEAMAIN program. This job may be submitted to batch using the SELCOPYi text editor SUBMIT primary command.

JCL is assigned to <F6> by default.

OPTIONS

```
>>---- OPTions ------------------------------------------------------><
```

Applicable only to the **Edit DB2 Table** panel, OPTION opens the Edit DB2 Table Options panel view to tailor DB2 concurrency and locking options, etc.

OPTIONS is assigned to <F19> by default.

SELECT

```
>>--+- SELect --------------------+-----------------------------------><
    +- SORT ---------------------+
```

Applicable only to the **Edit DB2 Table** panel, SELECT opens the DB2 Column Selection and Ordering panel to select table columns and establish the row (order by) sequence.

SELECT is assigned to <F17> by default.

SQL

```
>>---- SQL -----------------------------------------------------------><
```

Applicable only to the **Edit DB2 Table** panel, SQL opens the to view and optionally modify the DB2 SELECT, WHERE and ORDER BY clauses generated by the DB2 Row Selection (WHERE) and DB2 Column Selection and Ordering (SELECT/SORT) panels.

SQL is assigned to <F20> by default.

WHERE

```
>>---- WHere ----------------------------------------------------------><
```

Applicable only to the **Edit DB2 Table** panel, WHERE opens the DB2 Row Selection panel to define the table row selection criteria.

WHERE is assigned to <F18> by default.

Browse Tables and Views

Like DB2 table edit, table data may be browsed using the SELCOPYi Data Editor. A DB2 result table is generated based on an SQL query and rows are fetched and displayed in a structured format as for dataset records displayed using a COBOL/PL1/HLASM copybook structure.

Unless a SELCOPYi DB2 structure (SDO) is specified, SELCOPYi will generate a temporary structure using the DB2 SQLDA chain for the results table columns.

All standard Data Editor browse features (including FIND, WHERE and LOCATE) are available for DB2 table browse. General features of the Data Editor and those specific to DB2 table edit views are documented in detail in the SELCOPYi Structured Data Editor (SDE) manual.

Differences between the standard Data Editor view for datasets and that displayed for DB2 tables are as follows:

1. Abbreviated forms of the standard DB2 datatypes are displayed in the field type header line instead of the SDE field datatype, position and length. Display of this header line is controlled using SET TYPE ON/OFF.
2. An additional SQLCode record information field for DB2 which may be displayed and hidden using SET RECINFO SQLCODE/NOSQLCODE or selecting option 2 in the DB2 Edit Utilities manu (<PF4>).

Browse DB2 Table Panel

The Browse DB2 Table panel (ZZS2BROW) allows the user to configure SQL query cluases in order to generate a DB2 result table for browse in an Data Editor view.

It is an interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select "Browse", option 4, from the SELCOPY/i DB2 primary option menu. (DB2 4)
- Type the "B" line-command against an entry in a DB2 Tables List.

Input fields in each of the DB2 Edit panel views reflects the DB2 input parameters supported by its equivalent SDE line command, BROWSE.

Having configured the input fields, press <Enter> to execute the browse.

Menu Bar Items

File
: The File rename-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Command
: The "Command" menu bar item will display in a text edit view the SDE BROWSE command generated for the selected input field values. The generated command may then be executed by placing the cursor and pressing the "ACTION" key, or saved in the user's HOME file.

JCL
: The "JCL" menu bar item will display in a text edit view a batch job containing the SDE BROWSE command generated for the selected input field values. Further edit commands may be added (each separated by semi-colon) in order to run useful browse features in batch. e.g. to print portions of a table.

Structure
: Open the Create DB2 Table Edit Structure panel to edit or create a permanent SELCOPYi DB2 structure that may be used in the Browse DB2 table panel.

Help
: Open the general help for the Browse DB2 table panel.

Panel Fields - Browse DB2 Table

DB2 Base Table:

SSN>
: The DB2 subsytem on which the table to be browsed is located.
If left blank then the current subsystem (identified in the window title bar) will be used.

Location>
: The server location of the table to be browsed. If left blank, the local server for the connected DB2 sub-system is used.

Owner>
: The owner (schema) of the table to be browsed.
A table selection list will be provided if wildcards (*) are supplied.

Name>
The name of the table to be browsed.
A table selection list will be provided if wildcards (*) are supplied.

Structure File:
If activated, defines fields which together specify a structure file used to map the table data and specify various options for the browse session. If no structure is specified then SELCOPYi will generate one using the DB2 SQLDA chain for the specified results table columns.

Dsn>
Identifies the fully qualified data set name of an existing sequential data set or PDS/PDSE library. The dataset must containing an existing DB2 SDO structure.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must a specify a member name.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Auto>
This option controls the level at which the association of structure to DB2 table is applied

A selectable list of options will be presented if blank is entered.

D (Default)	Use current AUTOSTRUCT setting
Y (Yes)	Auto-struct applied and saved
N (No)	Auto-struct neither applied nor saved
A (Apply)	Auto-struct applied only
S (Save)	Auto-struct saved only

Table Mode:
The table browse mode is either Normal (N) or Large (L).

In Normal table browse mode, the entire result table will be loaded into the user's local Data Editor storage.

In Large table browse mode, a DB2 SENSITIVE STATIC scrollable cursor is used to fetch result table rows. This option is incompatible with the row selection options "Start" and "Max". See Using Scrollable Cursors for information on how the use of this option affects the browse session.

Use of DB2 scrollable cursors is disabled by default. They may be enabled by an authorised user by setting site-wide option option **DB2.SCROLL=YES** in the SELCOPYi System INI file.

Row Selection:

Start>
The start row within the result set.

Maxrt>
The maximum number of rows to be selected for edit.

XML LOB Width>
Specifies the number of bytes (*n_bytes*) of text, at the start of an XML or large object (LOB) column, to be displayed for all XML and LOB columns in the edited table view.

Use command XMLVIEW to open a new **text editor** view in order to display the XML document text located within a specific row and XML column of the current DB2 data editor view.

Browse DB2 Table SQL Clauses

The Browse DB2 Table SQL Clauses panel view is displayed only if command **SQL** is executed.

This panel view displays and supports editing of the SQL query clauses generated by the DB2 Row Selection (WHERE) and DB2 Column Selection and Ordering (SELECT/SORT) panels.

Panel Fields - Browse DB2 Table SQL CLauses

SQL:

Select>
The comma separated list of DB2 table column names that will be initially selected (e.g. visible in edit/browse). Although this field may be modified manually, it will be automatically updated (with all manually modification lost) whenever the SELECT primary command is used to select columns using the panel interface.

Where>
The "WHERE" clause passed as part of the SQL SELECT used to load data rows. Although this field may be modified manually, it will be automatically updated (with all manually modification lost) whenever the WHERE primary command is used to set row selection criteria using the panel interface.

Order By>
The "ORDER BY" clause passed as part of the SQL SELECT used to load data rows. Although this field may be modified manually, it will be automatically updated (with all manually modification lost) whenever the SELECT primary command is used to set row sort order using the panel interface.

Miscellaneous:

ColWidth>
A series of "OPTION(**COLWIDTH command**)" clauses passed to edit/browse to define the desired restricted visible width of long data columns. Although this field may be modified manually, it will be automatically updated (with all manually modification lost) whenever the SELECT primary command is used to set column width values using the panel interface.

Primary Commands

The following primary commands are supported by the DB2 Browse Tables and Views panels.

CMX

```
>>--+- CMX ----------------------+-----------------------------------><
    +- EDITCMX -------------------+
```

Applicable only to the **Browse DB2 Table** panel, CMX generates the SQL statement and copies it to an in-storage output file in a format suitable for subsequent execution using the ACTION (default >F16>) facility.

CMX is assigned to <F5> by default.

COLUMNS

```
>>---- COLumns -------------------------------------------------------><
```

Applicable only to the **Browse DB2 Table SQL Clauses** panel, COLUMNS sets the "Order By" field to a comma separated list of all columns names in the specified table.

Each column name (including its trailing comma) is deliberately blank padded up to 50 bytes. This is designed so that, when "EXPAND" (F14) is executed with the cursor in the "Order By" field so that the field contents are displayed in a separate Text Editor view, each column appears on a new line. This make it easy to re-order the fields and delete unwanted sort field names.

INDEX

```
>>--+- INDEX --------------------+-----------------------------------><
    +- IX ------------------------+
```

Applicable only to the **Browse DB2 Table SQL Clauses** panel, INDEX displays a selectable list of DB2 indexes that apply to the specified base table.

On selection of an individual index, its column names are used to populate the "Order By" field.

JCL

```
>>--+- JCL ----------------------+-----------------------------------><
    +- EDITJCL -------------------+
```

Applicable only to the **Browse DB2 Table** panel, JCL generates the SQL statement and copies it to an in-storage output file with JCL statements that execute the SDEAMAIN program. This job may be submitted to batch using the SELCOPYi text editor SUBMIT primary command.

JCL is assigned to <F6> by default.

SELECT

```
>>--+- SELect -------------------+-----------------------------------><
    +- SORT ---------------------+
```

Applicable only to the **Browse DB2 Table** panel, SELECT opens the DB2 Column Selection and Ordering panel to select table columns and establish the row (order by) sequence.

SELECT is assigned to <F17> by default.

SQL

```
>>---- SQL ------------------------------------------------------------><
```

Applicable only to the **Browse DB2 Table** panel, SQL opens the to view and optionally modify the DB2 SELECT, WHERE and ORDER BY clauses generated by the DB2 Row Selection (WHERE) and DB2 Column Selection and Ordering (SELECT/SORT) panels.

SQL is assigned to <F20> by default.

WHERE

```
>>---- WHere ----------------------------------------------------------><
```

Applicable only to the **Browse DB2 Table** panel, WHERE opens the DB2 Row Selection panel to define the table row selection criteria.

WHERE is assigned to <F18> by default.

Create DB2 Objects

Create DB2 Objects Panel

The Create DB2 Object panel (ZZS2C000) is an interactive panel window, opened on selection of option 5. in the DB2 Primary options menu.

This panel allows the user to select the type of object to be created by entering the relevant option number or by positioning the cursor on the required option and pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Note that successful creation of some DB2 objects is dependent upon the version of DB2 used by the connected DB2 system to which this panel applies. An SQL error message will occur if an SQL CREATE parameter field is used which is unsupported by the DB2 version. Similarly, successful creation of individual DB2 objects is also dependent upon the user's level of authority or granted privileges. Please refer to the relevant edition of the *"z/OS SQL Reference"*.

Menu Bar Items

`File`
: The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Help`
: Open the general help for the Create DB2 Objects option menu panel.

Options

1 Storage group	9 View
2 Work File Database	10 Alias
3 User Database	11 Synonym
4 Tablespace	12 Type
5 Work File Tablespace	13 Trigger
6 Base Table	14 Sequence
7 Materialized Query Table	15 Role
8 Index	16 Clone Table

Create Storage Group

The Create Storage Group sequence of panel views (ZZS2CSG0) generate an SQL CREATE STOGROUP statement to create a new storage group in the current DB2 server.

The DB2 Create Stroarge Group panel views are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select Storage Group option 1. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.1)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made in a the panel views. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL CREATE STOGROUP syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

Storage Group Values

Enter the name of the storage group to be created in the current DB2 sub-system. The current DB2 subsystem is displayed in the panel window title bar.

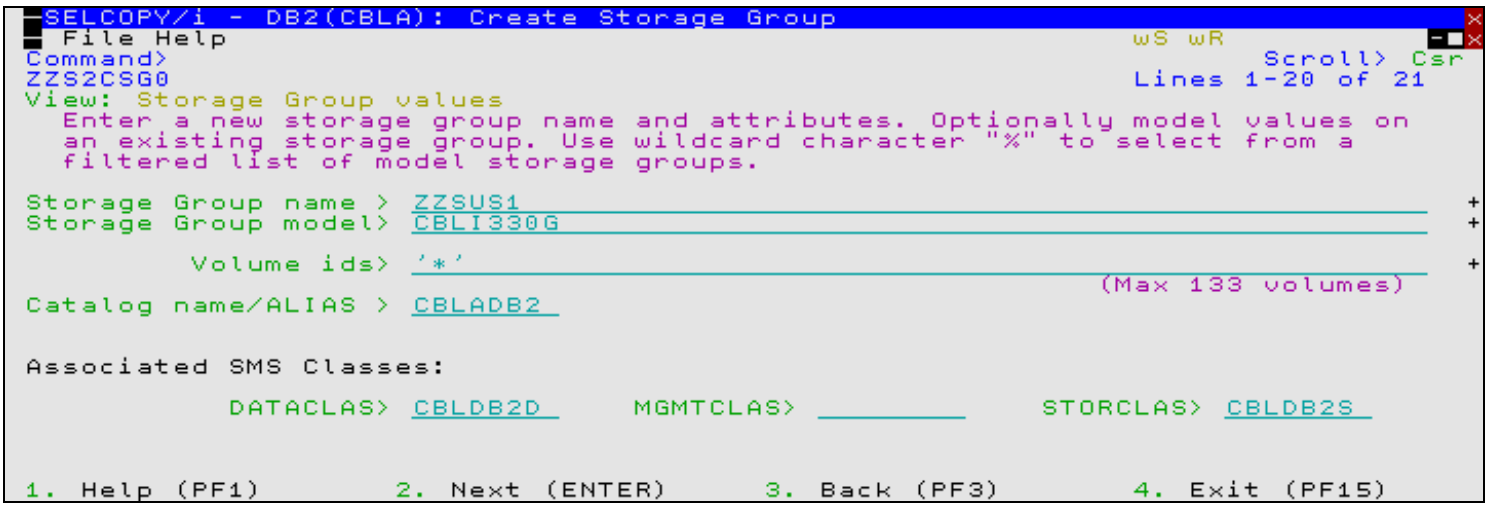


Figure 204. DB2: Create Storage Group.

Menu Bar Items

The following menu bar items are displayed in the Create Storage Group panel views.

File
The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

Storage Group Values - Panel Fields

Storage Group Name>
The name of a new storage group to be created at the current server. Maximum length of a storage group name is 128 characters.

Storage Group Model>
Optionally specifies the name of an existing storage group from which values for the new storage group may be modelled.

The model storage group may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the storage group name. If this is the case, a Select Storage Group panel is displayed containing a list of eligible storage groups that satisfy the model storage group name filter.

Use prefix command "S", or press the <Enter> key on the required entry to select it. Having selected a model storage group name, the values assigned to the panel fields will be updated accordingly.

Volume ids>
A volume ID or a comma separated list of volume IDs to be assigned to the storage group.

If a data set associated with the storage group is to be SMS managed, '*' (asterisk) may be specified as one of the volume IDs in order to allow SMS to select volumes as appropriate. Use of SMS is highly recommended rather than using DB2 to allocate data to specific volumes which would require non-SMS usage or an SMS Storage Class with guaranteed space. Assigning an SMS Storage Class with guaranteed space is not recommended as it reduces the benefits of SMS allocation.

A volume ID may be specified once only and the maximum number of volumes IDs in a storage group is 133. If one or more of DATACLAS, MGMTCLAS or STORCLAS is specified, then an entry in this input field is optional and so volume selection controlled by SMS.
This parameter field corresponds to SQL CREATE STOGROUP parameter VOLUMES

Catalog name/ALIAS>
Specifies the ICF catalog in which DB2 data sets will be cataloged.

The ICF catalog name or catalog alias has a maximum length of 8 characters and its specification is mandatory.
This parameter field corresponds to SQL CREATE STOGROUP parameter VCAT.

Associated SMS Classes:
Applicable to SMS controlled volume selection only.

DATACLAS>
The name of the SMS data class to be associated with the DB2 storage group. (Maximum length 8 characters).
This parameter field corresponds to SQL CREATE STOGROUP parameter DATACLAS.

MGMTCLAS>
The name of the SMS management class to be associated with the DB2 storage group. (Maximum length 8 characters).
This parameter field corresponds to SQL CREATE STOGROUP parameter MGMTCLAS.

STORCLAS>
The name of the SMS storage class to be associated with the DB2 storage group. (Maximum length 8 characters).
This parameter field corresponds to SQL CREATE STOGROUP parameter STORCLAS.

Generate SQL

The Generate SQL panel view is the last panel view in the sequence and is common to all SELCOPY/i DB2 Create Objects panel view sequences. For details, please refer to Generate SQL under the description of the DB2 Create Table sequence of panels.

This view allows the user to select how the generated SQL CREATE STOGROUP statement is to be implemented.

Create Work File Database

Applicable only in a data sharing environment, the Create Work File Database sequence of panel views (ZZS2CDBW) generate an SQL CREATE DATABASE statement to create the one and only work file database in the current DB2 server.

The DB2 Create Work File Database panel views are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select Work File Database option 2. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.2)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made in a the panel views. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL CREATE DATABASE syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

Work File Database Values

Enter the name of the work file database to be created in the current DB2 sub-system. The current DB2 subsystem is displayed in the panel window title bar.

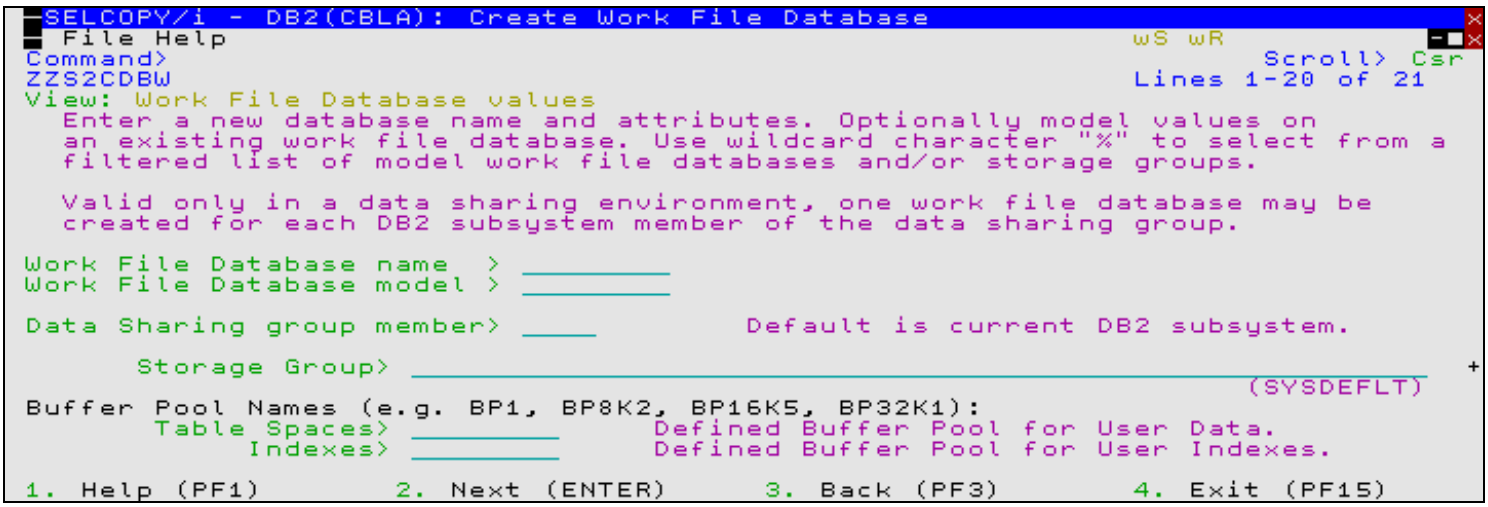


Figure 205. DB2: Create Work File Database.

Menu Bar Items

The following menu bar items are displayed in the Create Work File Database panel views.

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Work File Database Values - Panel Fields

Work File Database Name>
: The name of the new work file database.

  The database name has a maximum length of 8 characters and must not be DSN%%%%% (where '%' is any single character).

Work File Database Model>
: Optionally specifies the name of an existing database from which values for the work file database may be modelled.

  The model database may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the database name. If this is the case, a Select

Database panel is displayed containing a list of eligible databases that satisfy the model database name filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it. Having selected a model database name, the values assigned to the panel fields will be updated accordingly.

Data Sharing group member>
Specifies the DB2 data sharing member name of the DB2 subsystem for which this work file database applies.

The maximum length of a member name is 8 characters and, if not specified, defaults to the member name of the current DB2 subsystem.
This parameter field corresponds to SQL CREATE DATABASE parameter FOR.

Storage Group name>
The name of the default storage group to be used as required when allocating data sets for database table spaces and indexes.

The storage group may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the storage group name. If this is the case, a Select Storage Group panel is displayed containing a list of eligible storage groups that satisfy the model storage group name filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

The maximum length of a storage group name is 128 characters and, if not specified, defaults to SYSDEFLT.
This parameter field corresponds to SQL CREATE DATABASE parameter STOGROUP.

Buffer Pool names:
Default buffer pools to be used for table spaces and indexes created in the database.

Table Spaces>
Specifies the default buffer pool name to be used for table spaces. Possible buffer pool names are BPn (n=0 to 49) which correspond to buffer pools of size 4K.

If not specified, the buffer pool specified for user data in installation panel DSNTIP1 is used.
This parameter field corresponds to SQL CREATE DATABASE parameter BUFFERPOOL.

Indexes>
Specifies the default buffer pool name to be used for indexes.

Possible buffer pool names are BPn (n=0 to 49), BP8Kn, BP16Kn (n=0 to 9) and BP32K or BP32Kn (n=1 to 9). These correspond to buffer pools of size 4K, 8K, 16K and 32K respectively.

If not specified, the buffer pool specified for user indexes in installation panel DSNTIP1 is used.
This parameter field corresponds to SQL CREATE DATABASE parameter INDEXBP.

Generate SQL

The Generate SQL panel view is the last panel view in the sequence and is common to all SELCOPY/i DB2 Create Objects panel view sequences. For details, please refer to Generate SQL under the description of the DB2 Create Table sequence of panels.

This view allows the user to select how the generated SQL CREATE DATABASE statement is to be implemented.

Create User Database

Applicable only in a data sharing environment, the Create User Database sequence of panel views (ZZS2CDBU) generate an SQL CREATE DATABASE statement to create a new user database in the current DB2 server.

The DB2 Create User Database panel views are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select User Database option 3. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.3)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made in a the panel views. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL CREATE DATABASE syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

User Database Values

Enter the name of the user database to be created in the current DB2 sub-system. The current DB2 subsystem is displayed in the panel window title bar.

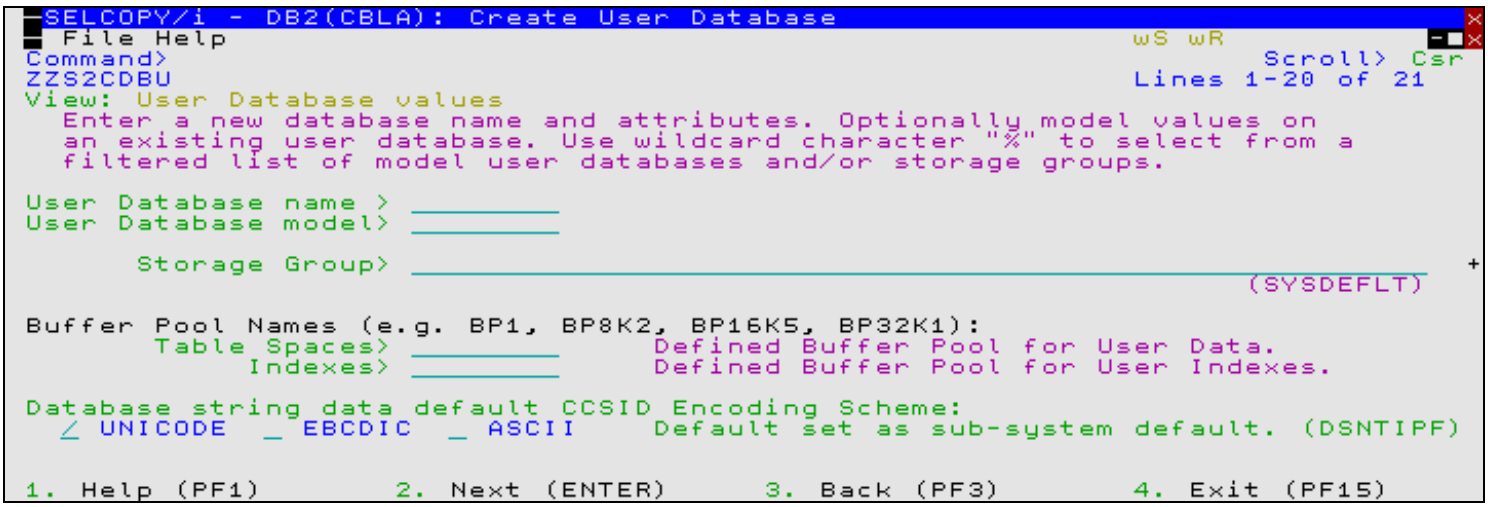


Figure 206. DB2: Create User Database.

Menu Bar Items

The following menu bar items are displayed in the Create User Database panel views.

File
The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

User Database Values - Panel Fields

User Database Name>
The name of the new user database.

The database name has a maximum length of 8 characters and must not be DSN%%%%% (where '%' is any single character) and must not begin with DSNDB.

User Database Model>
Optionally specifies the name of an existing database from which values for the user database may be modelled.

The model database may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the database name. If this is the case, a Select

Database panel is displayed containing a list of eligible databases that satisfy the model database name filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it. Having selected a model database name, the values assigned to the panel fields will be updated accordingly.

Storage Group>
The name of the default storage group to be used as required when allocating data sets for database table spaces and indexes.

The storage group may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the storage group name. If this is the case, a Select Storage Group panel is displayed containing a list of eligible storage groups that satisfy the model storage group name filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

The maximum length of a storage group name is 128 characters and, if not specified, defaults to SYSDEFLT.
This parameter field corresponds to SQL CREATE DATABASE parameter STOGROUP.

Buffer Pool names:
Default buffer pools to be used for table spaces and indexes created in the database.

Table Spaces>
Specifies the default buffer pool name to be used for table spaces.

Possible buffer pool names are BPn (n=0 to 49), BP8Kn, BP16Kn (n=0 to 9) and BP32K or BP32Kn (n=1 to 9). These correspond to buffer pools of size 4K, 8K, 16K and 32K respectively.

If not specified, the buffer pool specified for **user data** in installation panel DSNTIP1 is used.
This parameter field corresponds to SQL CREATE DATABASE parameter BUFFERPOOL.

Indexes>
Specifies the default buffer pool name to be used for indexes.

Possible buffer pool names are BPn (n=0 to 49), BP8Kn, BP16Kn (n=0 to 9) and BP32K or BP32Kn (n=1 to 9). These correspond to buffer pools of size 4K, 8K, 16K and 32K respectively.

If not specified, the buffer pool specified for **user indexes** in installation panel DSNTIP1 is used.
This parameter field corresponds to SQL CREATE DATABASE parameter INDEXBP.

Database Encoding:
Default encoding scheme for table spaces created in the database. Mutually exclusive options are as follow:

◊ **Default**
Data must be encoded using the default CCSIDs as defined by the DEF ENCODING SCHEME value specified in the installation panel DSNTIPF.

◊ **EBCDIC**
Data must be encoded using the EBCDIC CCSIDs of the server.

◊ **ASCII**
Data must be encoded using the ASCII CCSIDs of the server.

◊ **UNICODE**
Data must be encoded using the UNICODE CCSIDs of the server.

Generate SQL

The Generate SQL panel view is the last panel view in the sequence and is common to all SELCOPY/i DB2 Create Objects panel view sequences. For details, please refer to Generate SQL under the description of the DB2 Create Table sequence of panels.

This view allows the user to select how the generated SQL CREATE DATABASE statement is to be implemented.

Create Work File Table Space

The DB2 **Create Work File Table Space** sequence of panel views (ZZS2CTSW) generate an SQL CREATE TABLESPACE statement to create a new table space in the selected work file database in the current DB2 subsystem.

The DB2 Create Work File Table Space panel views and their sub-panels are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select Work File Table Space option 5. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.5)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL CREATE TABLESPACE syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

Work File Table Space Values

Enter the name of the new work file table space, table space data set attributes, data buffer pool and segment size. Enter a wild card in to select the work file database in the current DB2 server in which the table space will be created. The current DB2 subsystem is displayed in the panel window title bar.

Having entered the required values, proceed to the Generate SQL panel view which is common to all DB2 create obect panels.

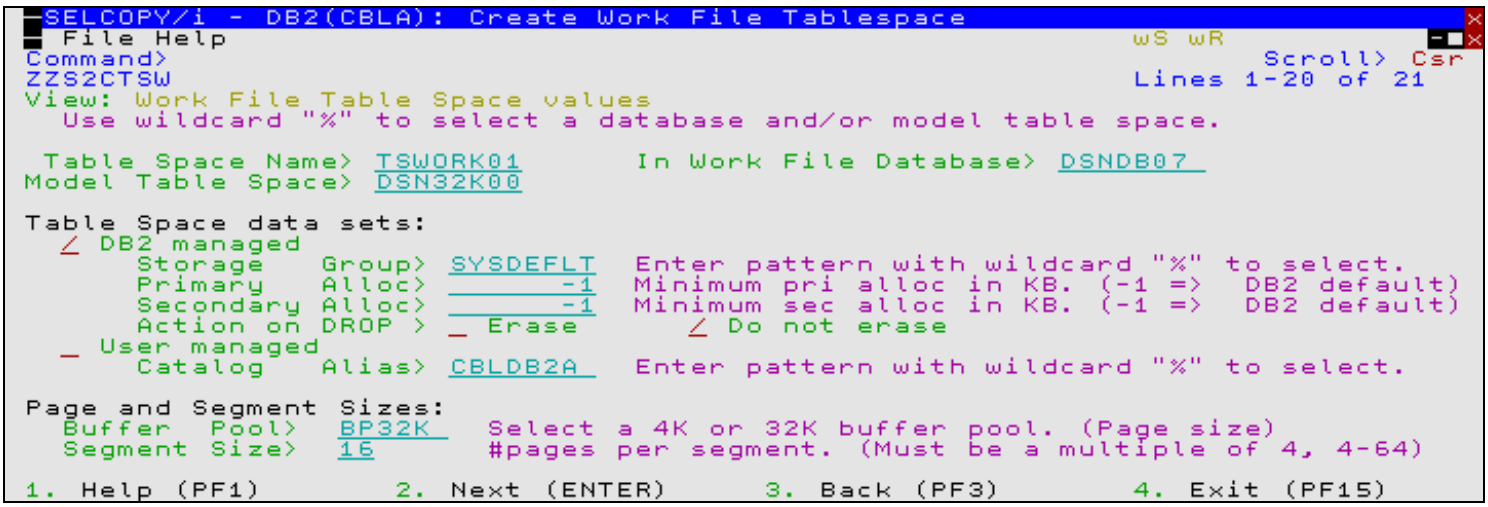


Figure 207. DB2: Create Work File Table Space - Values.

Menu Bar Items

The following menu bar items are common to all Create Table Space panel views and sub-panels.

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Work File Table Space Values - Panel Fields

Table Space Name>
: Mandatory field which specifies the name of the table space to be created.

  A table space name has a maximum length of 8 characters.

In Work File DataBase>
: Mandatory field which specifies the name of the work file database in which the table space will be created.

The database name may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the database name. If this is the case, a Select Database panel is displayed containing a list of eligible database names which satisfy the database name filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

A database name has a maximum length of 8 characters.

Model Table Space>
Optionally specifies the name of a table space from which values for the new table space may be modelled.

The model table space name may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the table space name. If this is the case, a Select Tablespace panel is displayed containing a list of eligible tablespace names which satisfy the model table space filters.
Use prefix command "S", or press the <Enter> key on the required entry to select it. Having selected a model table space name the values assigned to the panel fields will be updated accordingly.

Table Space Data sets:
Enter "/" to select the method by which table space data set will be managed and enter values appropriate to the selected method. Options are as follow:

- ◊ **DB2 managed**
  Specifies that DB2 will define and manage the data sets for the table space. Each data set will be defined on a volume of the identified storage group with primary and secondary allocations sizes defined by the Primary and Secondary Alloc fields respectively.

- ◊ **User managed**
  Specifies that table space data sets are to be managed by the user. Table space data sets are linear VSAM data sets cataloged in an ICF catalog identified by the catalog name/alias degfined by the Catalg Alias field.

Storage Group>
Mandatory field for DB2 managed data sets which specifies the name of a storage group defined in the current DB2 server.

The storage group name may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the storage group name. If this is the case, a Select Storage Group panel is displayed containing a list of eligible storage groups which satisfy the name filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

Primary Alloc>
Mandatory field for DB2 managed data sets which specifies the minimum primary allocation size in KB for the table space data set.
A value of -1 indicates to DB2 that it should use a value based on system defaults. A value other than -1 may be adjusted by DB2 to satisfy minimum requirements for the associated buffer pool page size.

Secondary Alloc>
Mandatory field for DB2 managed data sets which specifies the minimum secondary allocation size in KB for the table space data set.
A value of -1 indicates to DB2 that it should use a value based on system defaults.

Action on DROP>
Enter "/" to select whether or not the table space data set will be erased when the table space is deleted on execution of a utility or dropped using an SQL statement.

Catalog Alias>
Mandatory field for User managed data sets which specifies the name or alias of the ICF catalog in which the table space data sets are to be cataloged.

The catalog alias may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the storage group name. If this is the case, a Select Catalog Alias panel is displayed containing a list of eligible aliases that satisfy the model catalog alias filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

Buffer Pool>
Specifies the 4K or 32K buffer pool name to be used for the table space and so determines the table space page size. Enter blanks or any invalid value to select from a list of valid buffer pool names.

Segment Size>
Specifies an integer value (4-64) which defines the size of table space segments. The size corresponds to a number of pages to be assigned to each segment.

Note that an invalid value will be rounded to a multiple of 4 in the range 4 to 64 inclusive.

Generate SQL

The Generate SQL panel view is the last panel view in the sequence and is common to all SELCOPY/i DB2 Create Objects panel view sequences. For details, please refer to Generate SQL under the description of the DB2 Create Table sequence of panels.

This view allows the user to select how the generated SQL CREATE TABLESPACE statement is to be implemented.

Create User Table Space

The DB2 **Create User Table Space** sequence of panel views (ZZS2CTS0) generate an SQL CREATE TABLESPACE statement to create a new table space within the selected database of the current DB2 subsystem.

The DB2 Create User Table Space panel views and their sub-panels are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select User Table Space option 4. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.4)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made in a the panel views. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL CREATE TABLESPACE syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

Table Space Name & Type

Enter the name of the new table space, its type and, optionally, the name of the database in the current DB2 server in which the table space will be created. The current DB2 subsystem is displayed in the panel window title bar.

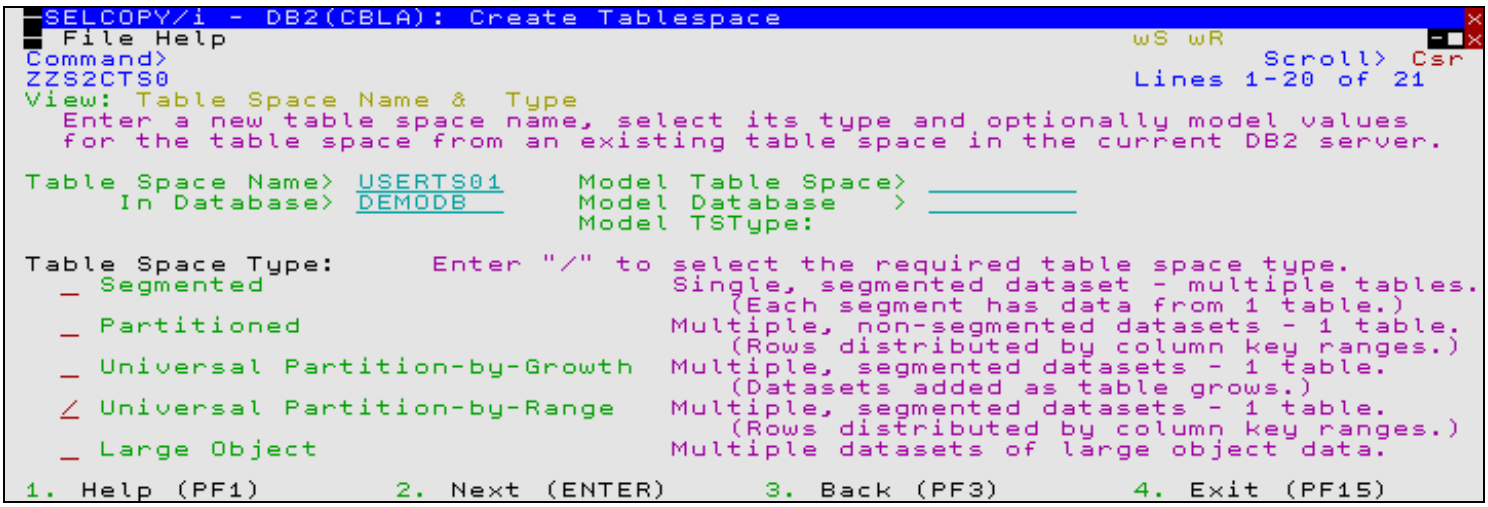


Figure 208. DB2: Create Table Space - Name & Type.

Menu Bar Items

The following menu bar items are common to all Create Table Space panel views and sub-panels.

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Table Space Name & Type - Panel Fields

Table Space Name>
: Mandatory field which specifies the name of the table space to be created.

  A table space name has a maximum length of 8 characters.

In DataBase>
: Optional field which specifies the name of the user database in which the table space will be created.

  The database name may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the database name. If this is the case, a Select Database panel is displayed containing a list of eligible database names which satisfy the database name filter.

Use prefix command "S", or press the <Enter> key on the required entry to select it. Having selected a database, if no modelling has occurred, then table space buffer pool name, DB2 storage group and CCSID encoding scheme defaults will be updated to values defined for the database.

A database name has a maximum length of 8 characters. The default database is DSNDB04.

Model Table Space>
Optionally specifies the name of a table space from which values for the new table space may be modelled.

The model table space name may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the table space name. If this is the case, a Select Tablespace panel is displayed containing a list of eligible tablespace names which satisfy the model table space filters.
Use prefix command "S", or press the <Enter> key on the required entry to select it. Having selected a model table space name the values assigned to the panel fields will be updated accordingly.

Model Database>
Optionally specifies the name of a database which is used in conjunction with the model table space field value to filter the list of model table space names presented to the user.

Model TSType:
A non-enterable field which identifies the type of table space selected from which the new table space will be modelled. Possible table space type descriptions are as follow:

- ◊ **Segmented**
  The selected table space is an exclusively segmented (non-partitioned) table space.

- ◊ **Partitioned.**
  The selected table space is an exclusively partitioned (non-segmented) table space.

- ◊ **Universal Partition-by-Growth.**
  The selected table space is a universal (partioned and segmented) partitioned-by-growth table space. Partitions added as the table space grows.

- ◊ **Universal Partition-by-Range.**
  The selected table space is a universal (partitioned and segmented) partitioned-by-range table space. All partitions defined and data clustered based on partitioning key column values.

- ◊ **LOB Table Space**
  The selected table space is a LOB table space. LOB table spaces must exist in the same database as the tablespace in which the LOB column is defined.

Table Space Type:
Enter "/" to select the type of table space to be defined.

If modelling has occurred, then this option will match that of the modelled table space but may still be updated by the user. This allows the user to create a table space of a different type to that of the model table space but still use its attributes where applicable. e.g. Create a new Universal Partition-by-Growth modelled on values from a Segmented table space for migration purposes.

Options are as described for Model TSType above.

Table Space Type Attributes

The table space attributes panel view is the next view in the sequence, displayed following the Table Space Name & Type panel view. The contents of the panel view include only those fields that are applicable to the selected table space type.

The panel view header displays one of the following types:

- Segmented Table Space
- Partitioned Table Space
- Universal Part-by-Growth
- Universal Part-by-Range
- LOB Table Space

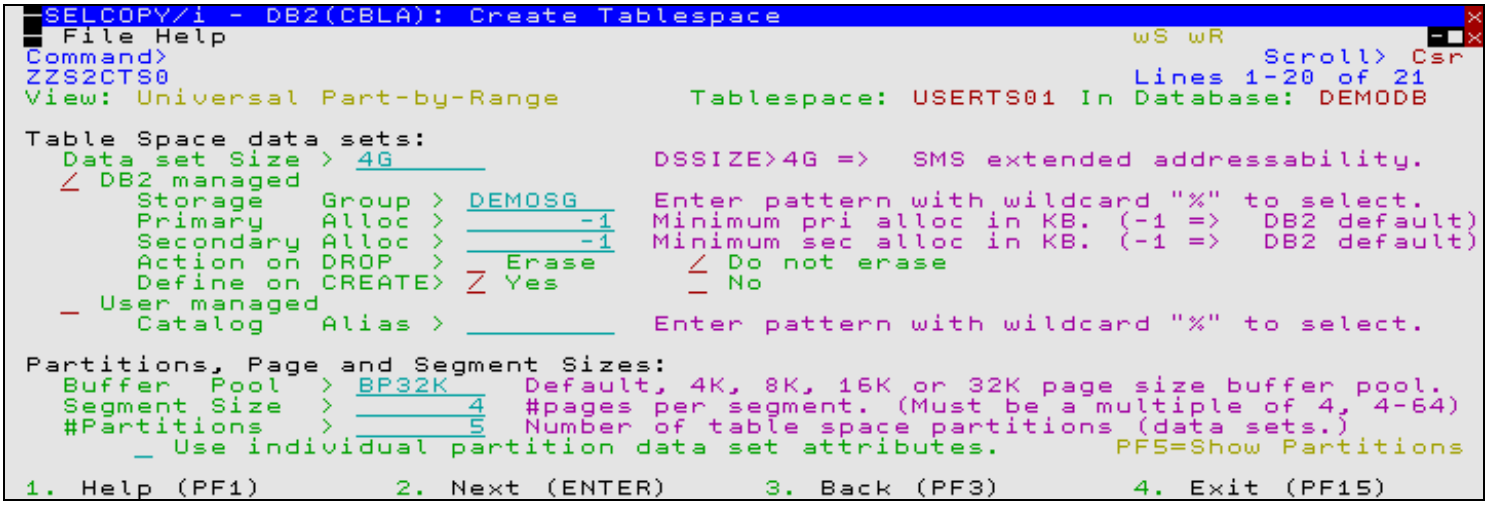


Figure 209. DB2: Create Table Space - Type attributes.

Table Space Type Attributes - Panel Fields

Note that of the panel fields described below, only those that are applicable to the selected table space type will be displayed.

TableSpace:
A non-enterable field displaying the new table space name as supplied in the **Table Name & Type** panel view.

In DataBase:
A non-enterable field displaying the database in which the new table space will be created, as supplied in the **Table Name & Location** panel view.

Data set Size>
Not applicable to exclusively Segmented table space type.

Specifies the maximum size in gigabytes of each table space partition or LOB table space data set. For sizes greater 4G, the data sets must be associated with an SMS DATACLAS that has been defined with extended format and extended addressability.

Enter blanks or any invalid value to select from a list of valid data set sizes.

Table Space Data sets:
Enter "/" to select the method by which table space data set will be managed and enter values appropriate to the selected method. Options are as follow:

- **DB2 managed**
  Specifies that DB2 will define and manage the data sets for the table space. Each data set will be defined on a volume of the identified storage group with primary and secondary allocations sizes defined by the Primary and Secondary Alloc fields respectively.

- **User managed**
  Specifies that table space data sets are to be managed by the user. Table space data sets are linear VSAM data sets cataloged in an ICF catalog identified by the catalog name/alias defined by the Catalog Alias field.
  Note that this option is not available for Universal Partition-by-Growth table space type.

Storage Group>
Mandatory field for DB2 managed data sets which specifies the name of a storage group defined in the current DB2 server.

The storage group name may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the storage group name. If this is the case, a Select Storage Group panel is displayed containing a list of eligible storage groups which satisfy the name filter. Use prefix command "S", or press the <Enter> key on the required entry to select it.

Primary Alloc>
Mandatory field for DB2 managed data sets which specifies the minimum primary allocation size in KB for the table space data set.

A value of -1 indicates to DB2 that it should use a value based on system defaults. A value other than -1 may be adjusted by DB2 to satisfy minimum requirements for the associated buffer pool page size.

Secondary Alloc>
Mandatory field for DB2 managed data sets which specifies the minimum secondary allocation size in KB for the table space data set.

A value of -1 indicates to DB2 that it should use a value based on system defaults.

Action on DROP>
Enter "/" to select whether or not the table space data set or partition data set will be erased when the table space is deleted on execution of a utility or dropped using an SQL statement.

Define on CREATE>
Enter "/" to select whether the table space data set or partition data sets will be allocated when the table space is created (Yes) or delayed until data is inserted into the table space.

Catalog Alias>
Mandatory field for User managed data sets which specifies the name or alias of the ICF catalog in which the table space data sets are to be cataloged.

The catalog alias may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the storage group name. If this is the case, a Select Catalog Alias panel is displayed containing a list of eligible aliases that satisfy the model catalog alias filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

Buffer Pool>
Specifies the 4K, 8K, 16K or 32K buffer pool name to be used for the table space and so determines the table space page size.

This field is usually initialised to be a value modelled on the model table space or the database default value. If a default value could not be determined from either of these sources, then "Default" is selected. "Default" will result in no BUFFERPOOL value being specified in the resultant CREATE TABLESPACE statement.

Enter blanks or any invalid value to select from a list of valid buffer pool names.

Segment Size>
Not applicable to exclusively Partitioned table space type.

Specifies an integer value (4-64) which defines the size of table space segments. The size corresponds to a number of pages to be assigned to each segment.

Note that an invalid value will be rounded to a multiple of 4 in the range 4 to 64 inclusive.

#Partitions>
Not applicable to exclusively Segmented or Universal Partition-by-Growth table space types.

Specifies an integer value (1-4096) which defines the number of range partitioned table space partitions.

The maximum number of partitions that can be specified depends on the selected buffer pool (page size) and data set size. Specifying a number of partitions that exceeds the maximum will return an error. See IBM publication "*DB2 SQL Reference*", "*CREATE TABLESPACE*" for partition number limits.

Max Partitions>
Applicable only to Universal Partition-by-Growth table space type.

Specifies an integer value (1-4096) which defines the maximum number of table space partitions that may be allocated as the table grows.

The maximum number of partitions that can be specified depends on the selected buffer pool (page size) and data set size. Specifying a number of partitions that exceeds the maximum will return an error. See IBM publication "*DB2 SQL Reference*", "*CREATE TABLESPACE*" for partition number limits.

Partition data set attributes
Not applicable to exclusively Segmented or Universal Partition-by-Growth table space types.

Select this option if attributes are to be specified for individual partition data sets. If this option is not selected, than all partitions will be allocated with attributes specified under **Table Space data sets:** in this panel view.

Having selected this option, the Table Space Partition Attributes sub-panel will be displayed. This will occur until at least one partition's attributes has been configured and that partition selected from the sub-panel.

Primary command SELECT (assigned to <F5> by default) will also display this sub-panel and select this option field.

Do not use clustering index (Member Cluster)
Applicable only to exclusively Partitioned table space type.

Select this option if data inserted by an insert operation is **not** to be clustered by the implicit or explicit clustering index. DB2 will, instead, locate the data in the table space based on available space (MEMBER CLUSTER).

Table Space Partition Attributes

The Table Space Partition Attributes sub-panel is displayed on executing the SELECT primary command or by selecting option **Partition data set attributes** in the Table Space Type Attributes panel view.

This sub-panel contains an embedded table of DB2 table space partitions and their data set attributes. The table is for update only so that rows may not be manipulated using the standard table editing techniques. The table of partition attributes may be reset by updating the **#Partitions** field value in the Table Space Type Attributes panel view.

Each table row identifies a partition of the table space. Zoom of an individual row displays a formatted view of the row data.

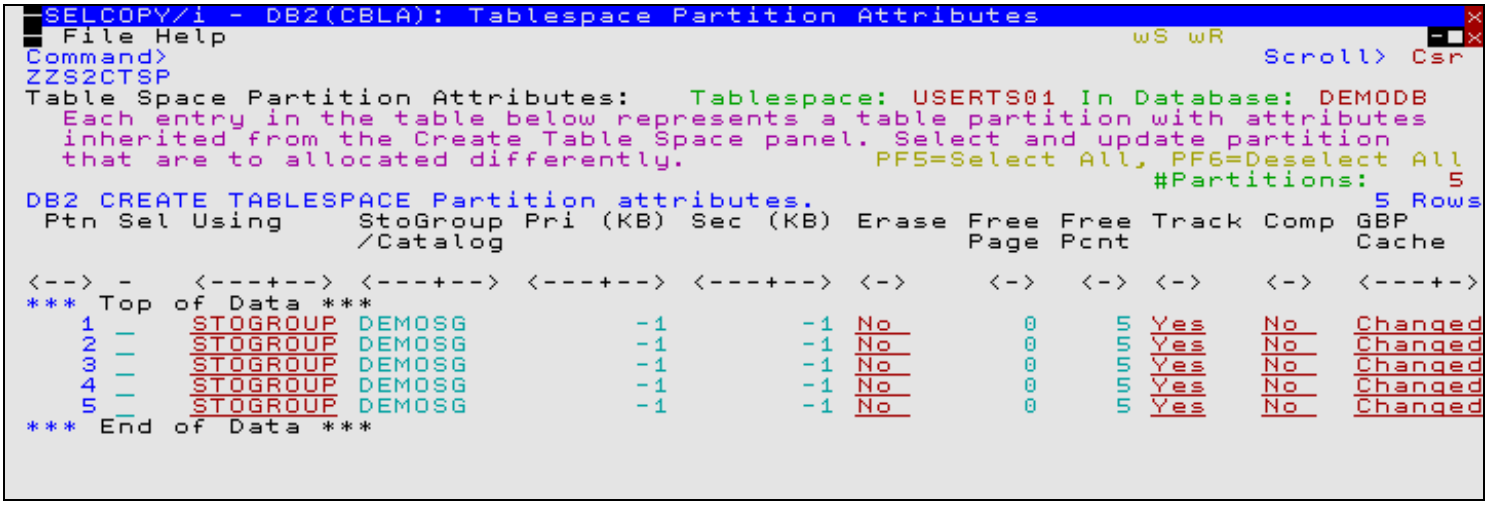


Figure 210. DB2: Create Table Space - Table Space Partition Attributes.

Table Space Partition Attributes - Panel Fields

`TableSpace:`
A non-enterable field displaying the new table space name as supplied in the **Table Name & Type** panel view.

`In DataBase:`
A non-enterable field displaying the database in which the new table space will be created, as supplied in the **Table Name & Location** panel view.

`Partition Number: (Ptn)`
A non-enterable field displaying the table space partition number.

`Sel> (Sel)`
Enter "/" (slash) or any non-blank character in this field to include (select) the attributes for the partition number in the create table space syntax. A blank in this field will exclude (deselect) the partition attributes.

Deselecting a column definition in table view has the same effect as executing the table primary command EXCLUDE (or line command "X") to exclude the table row.

`Table Space data sets: (Using)`
A value of STOGROUP or VCAT representing options "DB2 Managed" or "User Managed" respectively.

Options are as described for the Table Space Data sets: field of the Table Space Type Attributes panel view.

`Storage Group> (StoGroup/Catalog)`
Storage Group value as described for the Storage Group field of the Table Space Type Attributes panel view.

`Primary Alloc> (Pri (KB) )`
Primary allocation value as described for the Primary Alloc field of the Table Space Type Attributes panel view.

`Secondary Alloc> (Sec (KB) )`
Secondary allocation value as described for the Secondary Alloc field of the Table Space Type Attributes panel view.

`Action on DROP> (Erase)`
Data set erase option as described for the Action on DROP field of the Table Space Type Attributes panel view.

`Catalog Alias> (StoGroup/Catalog)`
Catalog alias name as described for the Catalog Alias field of the Table Space Type Attributes panel view.

`Free page frequency> (Free Page)`
Number of pages loaded after which a page of free space will be left. This is as described for the Free page frequency field of the Table Space Options (2/2) panel view.

`Percent free/page> (Free Pcnt)`
Percentage of space left free per page as described for the Percent free/page field of the Table Space Options (2/2) panel view.

`Track Changes> (Track)`
Option to track changes to data as described for the Track Modified Pages option field of the Table Space Options (2/2) panel view.

`Compress Rows> (Comp)`
Option to compress table space rows as described for the Compress Rows option field of the Table Space Options (2/2) panel view.

Group Buffer Pool Caching: (GBP Cache)
For data sharing only, identifies the type of group buffer pool caching as described for the Group Buffer Pool Caching option field of the Table Space Options (1/2) panel view.

Table Space Options (1/2)

The Table Space Options (1/2) panel view is the next view in the sequence, displayed following the Table Space Type Attributes panel view. The contents of the panel view reflect the selected table space type.

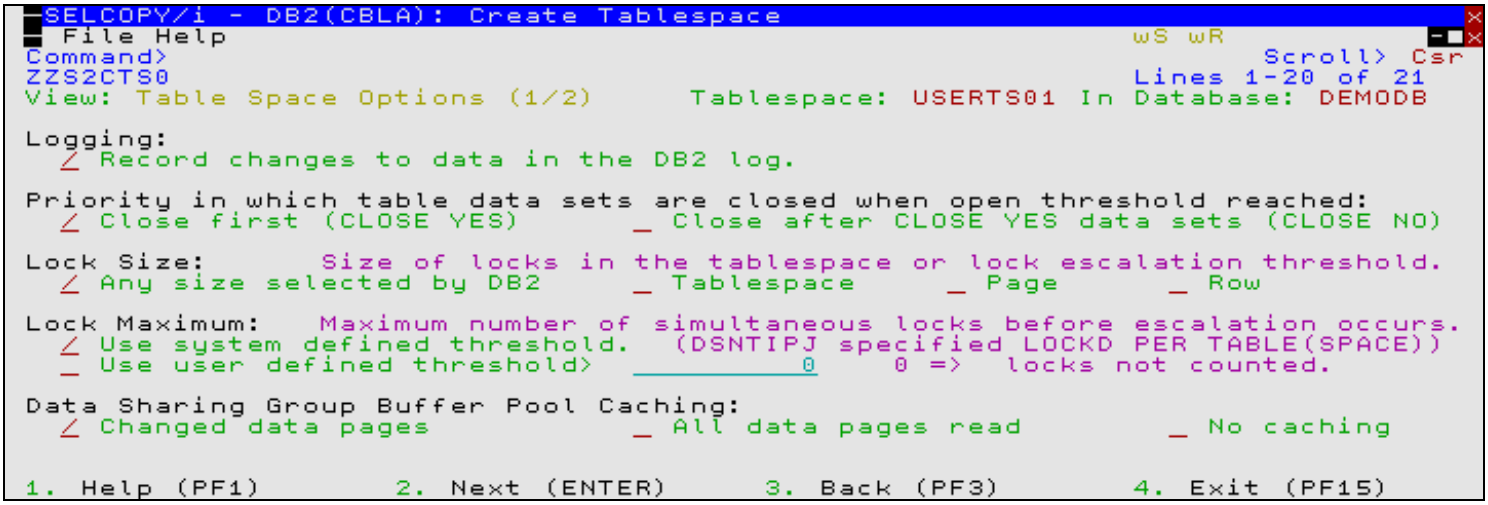


Figure 211. DB2: Create Table Space - Table Options (1/2).

Table Options (1/2) - Panel Fields

TableSpace:
A non-enterable field displaying the new table space name as supplied in the **Table Name & Type** panel view.

In DataBase:
A non-enterable field displaying the database in which the new table space will be created, as supplied in the **Table Name & Location** panel view.

Logged:
Select this option to record in the log all changes to the table and index data in the table space.

Close Priority:
Enter "/" to select the priority in which the table data sets belonging to the table space are closed when the limit of open table space data sets is reached.

Close first (CLOSE YES) indicates that the data sets will be eligible to be closed before data sets belonging to a table space created with CLOSE NO.

Lock Size:
Enter "/" to select the size of locks used in the table space and so the threshold at which lock escalation occurs. Options are as follow:

- ◊ **Any size selected by DB2**
- ◊ **Tablespace**
- ◊ **Table** - Applicable only to exclusively Segmented table space type.
- ◊ **Page** - Not applicable to LOB table spaces.
- ◊ **Row** - Not applicable to LOB table spaces.
- ◊ **LOB** - Applicable only to LOB table spaces.

Lock Maximum:
Enter "/" to select whether the maximum number of locks before escalation is to be determined by the user or by the DB2 system.

If user defined, the threshold value (0-2147483647) may be specified in the accompanying field. A value of 0 (zero) indicates that locks are not to be counted and so no escalation occurs.

Group Buffer Pool Caching:
Applicable only in a data sharing environment, enter "/" to select what pages of the table space or partition are written to the group buffer pool. In a non-data sharing environment, this option will be ignored. Options are as follow:

- ◊ **Changed data pages**
  Unless defined in a group buffer pool that is defined to be used only for cross-invalidation, cache only those table space pages containing table or index data that has been modified.

◊ **ALL data pages read**
Cache all table space pages as they are read from DASD.

◊ **Changed system pages**
Applicable only to LOB table spaces, cache only system pages within the LOB table space that have changed. A system page is a space map page or any other page that does not contain actual data values.

◊ **No caching**
Do not cache table space pages in the group buffer pool.

Table Space Options (2/2)

The Table Space Options (2/2) panel view is the next view in the sequence, displayed following the Table Space Options (1/2) panel view.

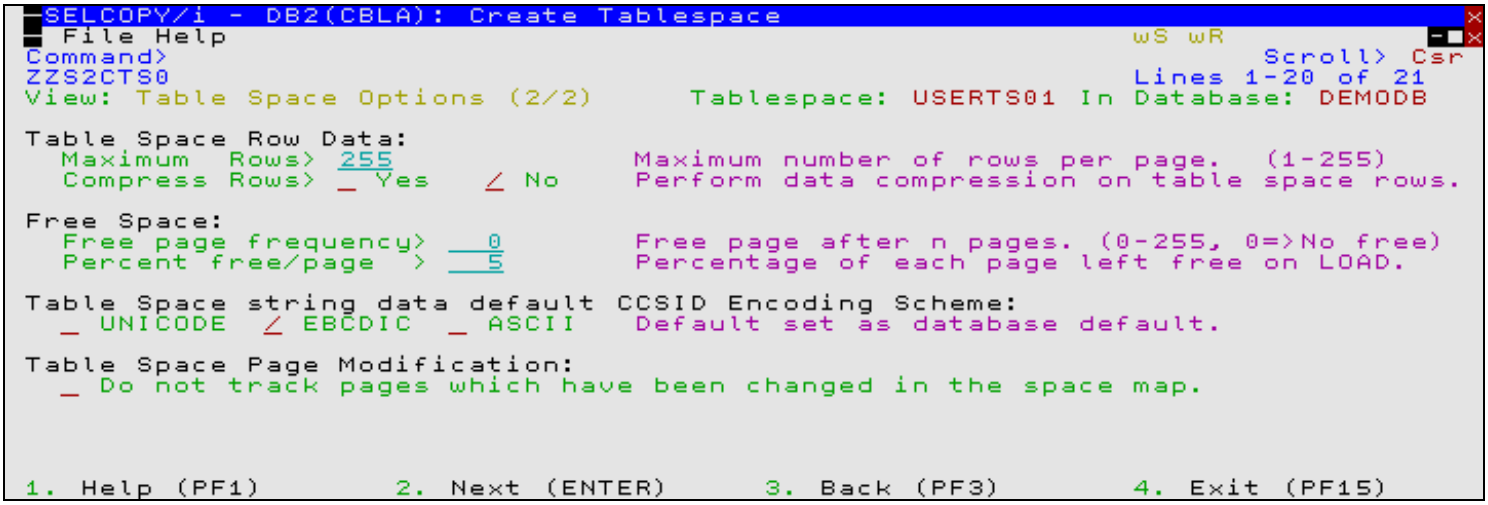


Figure 212. DB2: Create Table Space - Table Options (2/2).

Table Options (2/2) - Panel Fields

TableSpace:
A non-enterable field displaying the new table space name as supplied in the **Table Name & Type** panel view.

In DataBase:
A non-enterable field displaying the database in which the new table space will be created, as supplied in the **Table Name & Location** panel view.

Maximum Rows>
Specifies the maximum number of rows (1-255) that the DB2 system will consider placing on each data page for for insert, LOAD and REORG operations.

Compress Rows>
Enter "/" to select whether or not compression will be performed on rows of the table space or partition. If yes is selected, rows will not be compressed until the LOAD or REORG utility is run on the table in the table space or partition.

Free page frequency>
For a LOAD or REORG operation, specifies the number of pages (0-255) that will be loaded before a leaving a page of free space. The count of pages loaded restarts following the free page. A value of 0 (zero) indicates that no free pages are to be left.

Percent free/page>
For a LOAD or REORG operation, specifies the percentage (0-99) of each page to be left as free space.

CCSID Encoding Scheme:
Enter "/" to select the default encoding scheme (UNICODE, EBCDIC or ASCII) for table stored in the table space. If table space modelling has not occurred, then this field will be initialised to the encoding scheme assigned to the database in which the table space is to be created, otherwise the system default defined in install panel DSNTIPF.

Track Modified Pages:
Select this option if DB2 is to track modified pages in the space map pages of the table space or partition. Tracking modified pages improves the performance of incremental image copy.

Generate SQL

The Generate SQL panel view is the last panel view in the sequence and is common to all SELCOPY/i DB2 Create Objects panel view sequences. For details, please refer to Generate SQL under the description of the DB2 Create Table sequence of panels.

This view allows the user to select how the generated SQL CREATE TABLESPACE statement is to be implemented.

Primary Commands

The following primary commands are supported by selected views in the DB2 Create Table Space sequence of panel views. If issued from a panel view in which the command is not valid, the message "ZZSP102E Primary command not valid in the current context" is displayed.

CMX

```
>>--+- CMX ----------------------+-----------------------------------><
    +- EDITCMX -------------------+
```

Applicable only from the Generate SQL view, CMX generates the SQL statement and copies it to an in-storage output file in a format suitable for subsequent execution using the ACTION (default >F16>) facility.

CMX is assigned to <F17> by default.

JCL

```
>>--+- JCL ----------------------+-----------------------------------><
    +- EDITJCL -------------------+
```

Applicable only from the Generate SQL view, JCL generates the SQL statement and copies it to an in-storage output file with JCL statements that execute the DSNTIAD facility. This job may be submitted to batch using the SELCOPY/i text editor SUBMIT primary command.

JCL is assigned to <F18> by default.

RUN

```
>>--+- RUN ----------------------+-----------------------------------><
    +- EXECSYNtax ----------------+
```

Applicable only from the Generate SQL view, RUN verifies input fields in all applicable panel views and then executes the generated SQL statement. This is the default action on pressing <Enter> from the last panel view in the sequence (i.e. the **Generate SQL** view.)

SELECT

```
>>---- SELect ------------------------------------------------------><
```

SELECT will open a sub-panel that is applicable to the current panel view. Each of the panel views listed below identify the sub-panel opened on executing SELECT. In all other panel views, SELECT is invalid.

- **Table Space Type Attributes**
  For **Partitioned** and **Universal Partioned-by-Range** table spaces only, SELECT opens the Table Space Partition Attributes sub-panel, used to specify individual partition information.

SELECT is assigned to <F5> by default.

TRIGGERACTION

```
>>---- TRIGgeraction -----------------------------------------------><
```

From any of the Trigger Action panel views, TRIGGERACTION will open the Triggered SQL Statements sub-panel, used to define one or more SQL statements that are to be executed when the trigger is actioned.

TRIGGERACTION is assigned to <F6> by default.

Create Table

The DB2 **Create Table** series of panel views (ZZS2CT00) generate an SQL CREATE TABLE statement which may be used to create a new DB2 base or materialised query table in the current DB2 subsystem.

The DB2 Create Table panel views and their sub-panels are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select Base Table option 6. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.6)
- Select Materialized Query Table option 7. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.7)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made within the panel views and also on the type of table space in which the table will be created. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL CREATE TABLE syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

Table Name & Location

Enter the name of the new base or materialized query table and optionally the name of an explicitly defined tablespace within the current DB2 subsystem in which the new table will be created. The current DB2 subsystem is displayed in the panel window title bar.

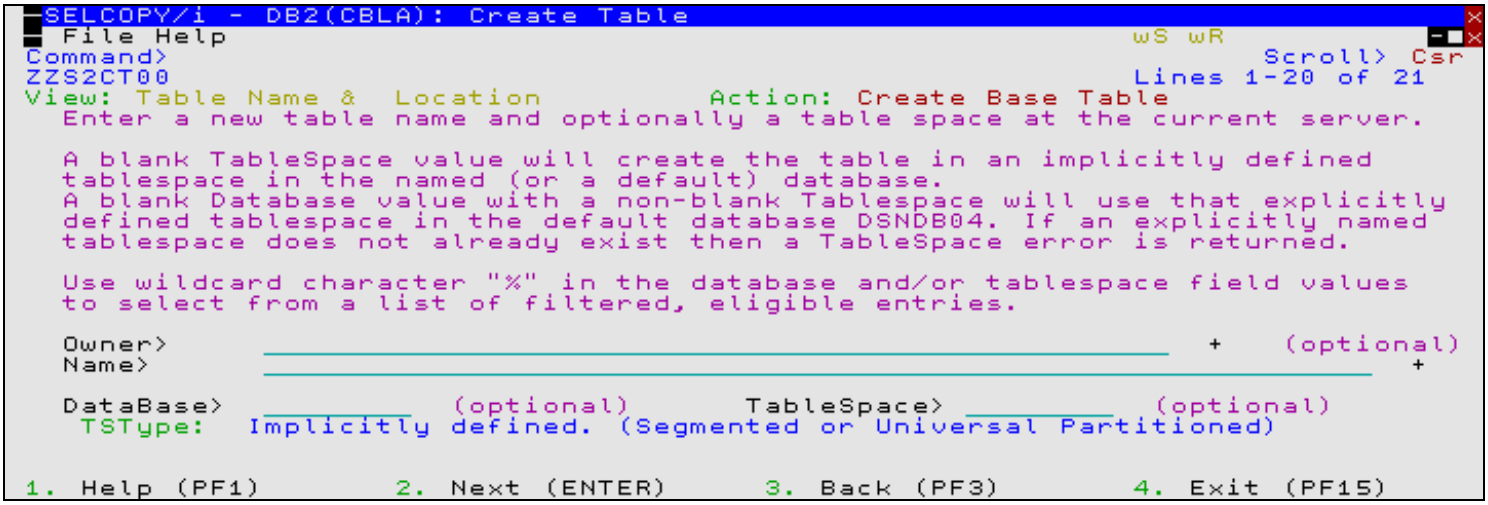


Figure 213. DB2: Create Table - Table Name & Location.

Menu Bar Items

The following menu bar items are common to all Create Table panel views and sub-panels.

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Table Name & Location - Panel Fields

Action:
: A non-enterable field which describes the current action being performed by the Create Table panel. Possible values are:

  ◊ **Create Base Table**
  ◊ **Create Materialized Query Table**

Owner>
Optionally specifies the owner (schema) of the table to be created. If this field is left blank, DB2 will assign an owner id equal to the value of the user's current SQLID special register.

A table owner id has a maximum length of 128 characters.

Name>
Mandatory field which specifies the name of the table to be created.

A table name has a maximum length of 128 characters.

DataBase>
Optionally specifies the name of a database in which the target table space is explicitly or implicitly defined.

The database name may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the database name. If this is the case, a Select Database panel is displayed containing a list of eligible database names which satisfy the database name filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

If no database or table space is specified, then the table space is implicitly defined in database **DSN*nnnnn***.
If no database is specified but a table space is specified, then the table space must have been explicitly defined in database **DSNDB04**.

A database name has a maximum length of 8 characters.

TableSpace>
Optionally specifies the name of a table space in which the new table will be created.

The table space name may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the table space name. If this is the case, a Select Tablespace panel is displayed containing a list of eligible tablespace names which satisfy the table space and database name filters.
Use prefix command "S", or press the <Enter> key on the required entry to select it. Using this method to select a table space name will also update the database name field value.

If no table space is specified, then the table space is implicitly defined in the specified database (or DSN*nnnnn*) with a table space name derived from the table name.

A table space name has a maximum length of 8 characters.

TSType:
A non-enterable field which identifies the type of table space in which the new table will be created. Possible table space type descriptions are as follow:

- **Implicitly defined. (Segmented or Universal Partitioned)**
  Table space will be defined when the table is created. The new, implicitly defined table space will be segmented and, depending on the function mode of the DB2 subsystem, a universal partitioned-by-growth or universal partitioned-by-range.
  If the table space is to be implicitly defined, the Implicit TableSpace Options view will be displayed later in the sequence of DB2 Create Table panel views.

- **Segmented. (*n* x *m*KB pages/segment)**
  The selected table space is an existing segmented (non-partitioned) table space defined with the displayed page size (KB) and number of pages per segment. This type of table space is capable of containing more than one table definition.

- **Partitioned. (*n* parts)**
  The selected table space is an existing, non-universal (i.e. non-segmented) partitioned table space defined as having the displayed number of partitions. This type of table space is capable of containing only one table definition.

- **Universal Partitioned-by-Growth. (Max *n* parts of max size *m*GB)**
  The selected table space is an existing universal (i.e. segmented) partitioned-by-growth table space defined as having the displayed maximum number of partitions, each of the displayed maximum size (GB). This type of table space is capable of containing only one table definition.

- **Universal Partitioned-by-Range. (*n* parts of max size *m*GB)**
  The selected table space is an existing universal (i.e. segmented) partitioned-by-range table space defined as having the displayed number of partitions, each of the displayed maximum size (GB). This type of table space is capable of containing only one table definition.

This field is updated following update of the DataBase/TableSpace fields before progressing to the next panel view.

Table Definition

The Table Definitions panel view is the next view in the sequence, displayed following the Table Name & Location panel view.

This view allows the user to select the method by which table column definitions, table constraints and various table options will be sourced.

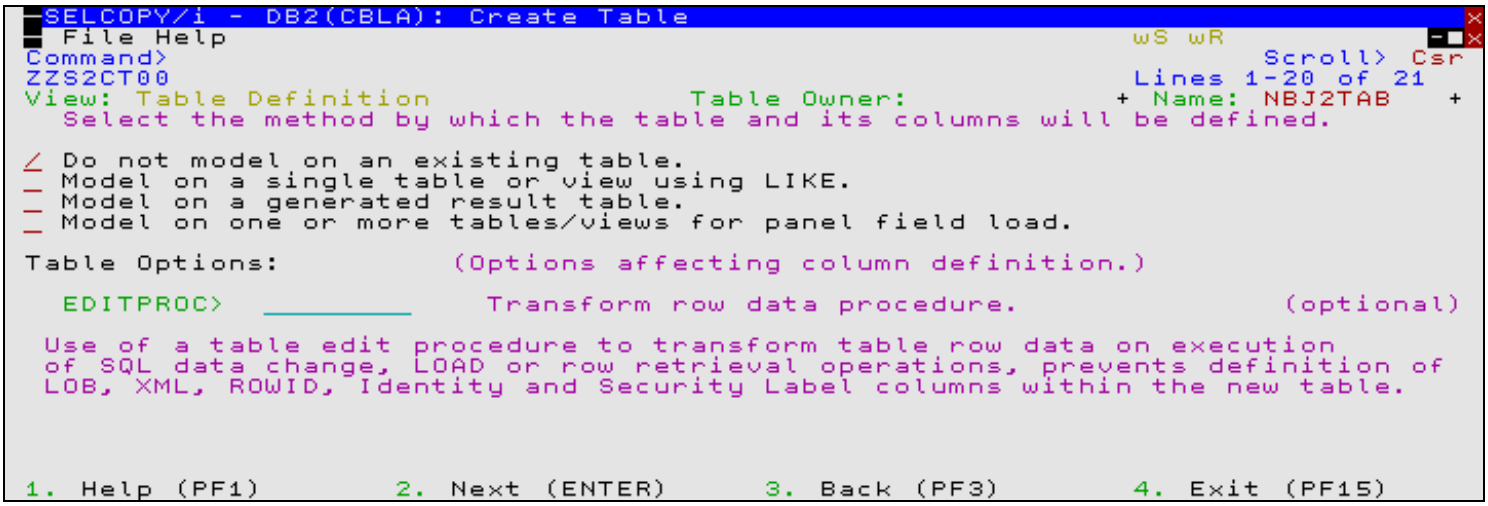


Figure 214. DB2: Create Table - Table Definition.

Table Definition - Panel Fields

`Table Owner:`
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

`Name:`
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

`Modelling Options`
Enter "/" (slash) or any non-blank character against the required entry to select the modelling method to be used to define columns, constraints and other attributes of the new table. Each of the options are mutually exclusive.

The sequence of panel views to be displayed will depend on the method selected.

- ◊ **Do not model on an existing table.**
  The table will **not** be modelled on an existing table in the current DB2 server. Instead, the user will be required to enter column definitions and any other table attributes manually.

- ◊ **Model on a single table or view using LIKE.**
  Use the SQL CREATE TABLE LIKE syntax to model the new table on a table or view which exists in the current DB2 server. See "DB2 for z/OS SQL Reference" for information on table attributes that are not modelled from the table or view specified using LIKE.
  Having selected this option, the Model using LIKE view will be displayed.

- ◊ **Model on a generated result table.**
  Use the SQL CREATE TABLE AS (*fullselect*) syntax to model the new table on the result of a query expression. See "DB2 for z/OS SQL Reference" for table attributes that are not modelled from a result table.
  Having selected this option, the Model on Result Table view will be displayed.

- ◊ **Model on one or more tables/views for panel field load.**
  Populate fields in subsequent Create Table panel views with values modelled on one or more tables and/or views defined in the current DB2 server.
  Having selected this option, the Model on Tables/Views view will be displayed.

`EDITPROC>`
Optionally specify the name of a table edit procedure to be used to transform table row data on execution of SQL data change, LOAD or row retrieval operations. Specification of a table edit procedure invalidates definition of LOB, XML, ROWID, Identity and Security Label columns within the new table.

A procedure name has a maximum length of 8 characters.

Columns & Constraints

The Columns & Constraints panel view is displayed if the user has selected not to model on an existing table or following any table modelling performed to load panel field values. In the latter case, the column and constraint definitions will have been updated as appropriate with the modelled column and/or constraint attributes as selected by the user in panel view Model on Tables/Views (Load Values).

This view allows the user to display column and constraint definition sub-panels to optionally alter/add/delete/select/deselect individual column and constraint definitions.

At least one column definition must exist and so, regardless of the current sub-panel selection, if no columns have been defined and selected the Column Definitions sub-panel will be displayed next in the panel sequence.

Having selected, displayed and then closed a definitions sub-panel, the focus returns to this panel view and, by default, the next definitions sub-panel is automatically selected for display. Also, fields displaying the column/constraint definition totals and existance of a primary key constraint are updated accordingly.

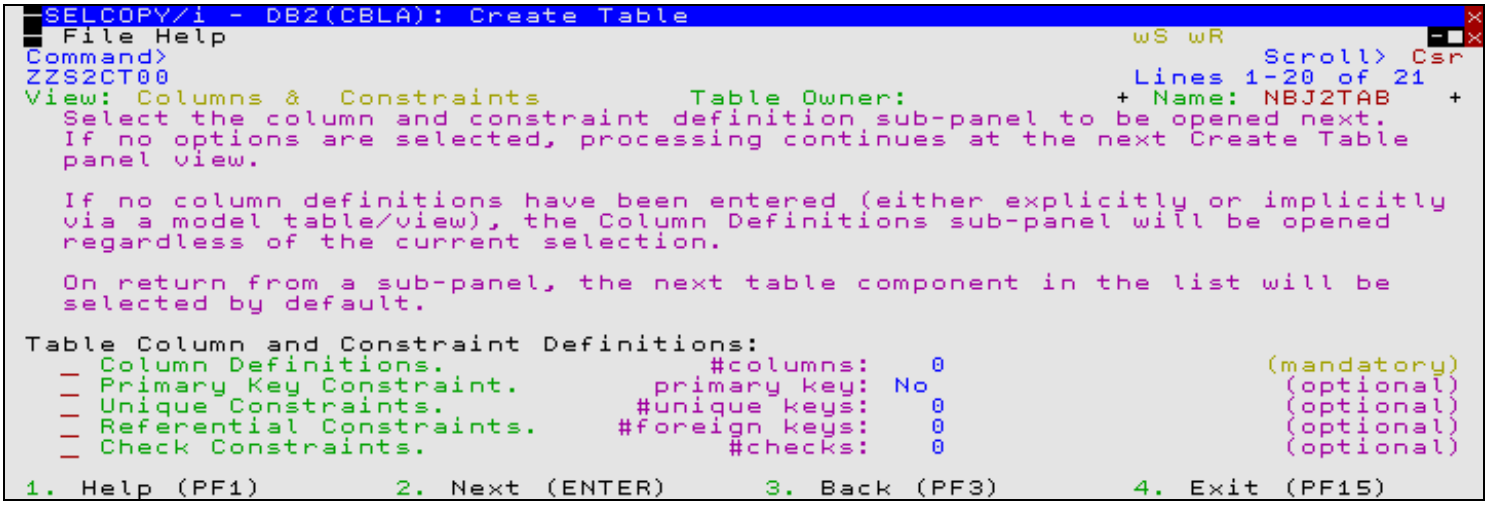


Figure 215. DB2: Create Table - Columns & Constraints.

Columns & Constraints - Panel Fields

`Table Owner:`
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

`Name:`
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

`Table Column and Constraint Definitions:`
Enter "/" (slash) or any non-blank character against the columns/constraints definitions sub-panel to be displayed next. Each of these sub-panel selections are mutually exclusive.

If no definitions sub-panel is selected and at least one column has been defined and selected, then focus will progress to the next panel view in the sequence (Table Options (1/2).)

On return from a definitions sub-panel, the next definitions sub-panel in the list is automatically selected for display. If the last sub-panel selected was **Check Constraints**, then no further sub-panels are selected for display.

- ◊ **Column Definitions.**
  Display the Column Definitions sub-panel next.
- ◊ **Primary Constraint.**
  Display the Primary Key Definition sub-panel next.
- ◊ **Unique Constraints.**
  Display the Unique Key Constraint Definitions sub-panel next.
- ◊ **Referential Constraints.**
  Display the Referential Constraint Definitions sub-panel next.
- ◊ **Check Constraints.**
  Display the Check Constraint Definitions sub-panel next.

`#columns:`
A non-enterable field displaying the total number of columns that have been defined **and** selected. This value is updated on return from the Column Definitions sub-panel.

prime key:
A non-enterable field displaying either "Yes" or "No" to indicate whether or not a primary key constraint has been defined This value is updated on return from the Primary Key Definition sub-panel.

#unique keys:
A non-enterable field displaying the total number of unique key constraints (excluding any primary key constraint) that have been defined **and** selected. This value is updated on return from the Unique Key Constraint Definitions sub-panel.

#foreign keys:
A non-enterable field displaying the total number of referential key constraints (parent/foreign key dependencies) that have been defined **and** selected. This value is updated on return from the Referential Constraint Definitions sub-panel.

#checks:
A non-enterable field displaying the total number of check constraints that have been defined **and** selected. This value is updated on return from the Check Constraint Definitions sub-panel.

Column Definitions

The Columns Definitions sub-panel is displayed on selecting **Column Definitions** from the Columns & Constraints panel view or if column definition validation fails when generating the SQL CREATE TABLE statement from the Generate SQL panel view.

This sub-panel contains an embedded table of DB2 table column definitions. Standard table edit primary and line commands must be used to INSERT, DELETE, COPY, MOVE or EXCLUDE table rows as appropriate, to scroll the table display UP, DOWN, LEFT and RIGHT and also to ZOOM the display of an individual table row.

In addition to the standard primary table commands, this sub-panel also supports primary command **SELECTAll** (assigned to <F6> by default) to toggle between selection and deselection of all column definitions in the table.

Each table row defines one DB2 table column which incorporates the following column attributes:

1. Name.
2. Type.
3. Data Length or Precision.
4. Data Scale. (DECIMAL only)
5. Default value support.
6. Null value support.
7. Field Procedure name.
8. Field Procedure parameters.
9. Distinct Type Schema & Name.
10. Sub-type. (Character data types only)
11. Implicitly hidden.

On inserting a new column definition, a unique name should be entered in the Column Name field and the Type field should be updated to reflect the required column type. All subsequent column definition validation is based on the selected column definition type value.

Following update of a table row (column definition) or on returning to the table view from a zoomed view, validation occurs for the column attributes entered by the user. Likewise, validation is performed for all column definitions on exiting the Column Definitions sub-panel. If an error is detected in a column definition that does not relate to the chosen column type, or if an updated column definition is of column type IDENTITY, RCT, ROWID or SECURITY, then that table row is automatically zoomed. This allows the user to correct the error or to enter column definition attributes which are not displayed in the table view, before continuing.

If the width of a field in the table view is not sufficient to enter the required input value, then the table row should be zoomed and, if necessary, the appropriate field expanded to accomodate the input value. To display a table row in single row (zoomed) format, execute primary command ZOOM (assigned to <F17> by default) with the cursor positioned on the required table row entry. EXPAND (assigned to <F14> by default) may then be used to expand an input field and so enter a value which is longer than the visible input field area.

Zoom of an individual row displays a view of that row containing only those input fields and field options that are pertinent to the column definition type. e.g. Zoom of a DATE type column will display fields which do **not** include column data length, field procedure name nor any other irrelevant input fields and options.

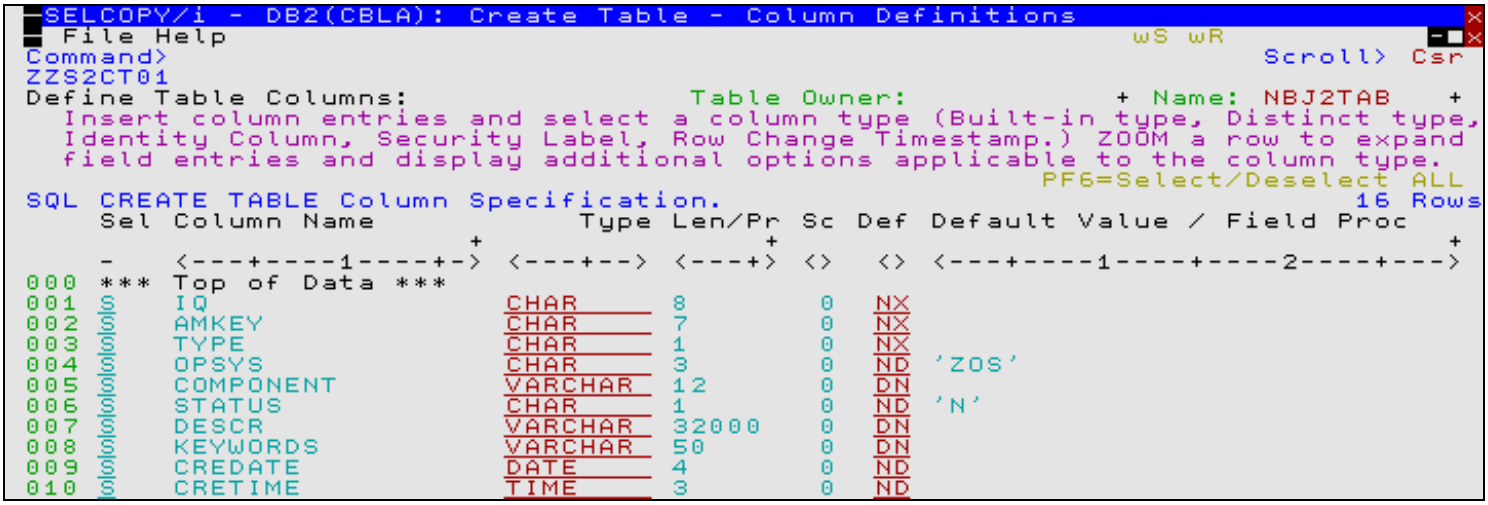


Figure 216. DB2: Create Table - Column Definitions.

Column Definitions - Panel Fields

Field names that follow are as appear in the zoomed view of a table row. Names in parentheses correspond to the equivalent column name in table view.

Column Name> (Column Name)
The unique name to be assigned to the column which is automatically upper cased.

A column name has a maximum length of 30 characters.

Column Type: (Type)
Identifies the column data type.

The value selected or entered in this field when in table view governs the view displayed when the table row is subsequently zoomed. In zoomed view, the contents of this field is non-enterable and is included for display purposes only.

In addition to the standard DB2 built-in data types, this field supports type DISTINCT if a distinct (user-defined) type is to be used to define the column.

One of the built-in or DISTINCT column data types should be specified unless the the column is to be defined as an identity column, a security label or a row change change timestamp column in which case one of the non-standard columns types of IDENTITY, SECURITY, ROWID or RCT must be specified respectively. The source type of these types of column, whether built-in or distinct, may be selected when the column definition is viewed in zoomed format.

Restrictions:

1. Graphic (DBCS) types (GRAPHIC, VARG and DBCLOB) are valid only if the table encoding is Unicode or the DB2 server supports double and mixed byte character sets.
2. Large object (CLOB, DBCLOB and BLOB), XML, ROWID, IDENTITY and SECURITY types are valid only if no EDIT Procedure has been assigned to the table.

Enter a blank or invalid value in this field to display a scrollable list of valid, selectable entries for this field.

CHAR
A fixed length character column.
Note that type SECURITY should be used if this column is to be a security label column.

VARCHAR
A variable length character column.

CLOB
A character large object column.

GRAPHIC
A fixed length graphic character (DBCS) column.

VARG
A variable length graphic character (DBCS) column.

DBCLOB
A graphic character (DBCS) large object column.

BINARY
A fixed length binary column.

VARBIN
A variable length binary column.

BLOB
A binary large object column.

SMALLINT
A small integer column. The range of a small integer is -32768 to +32767.
Note that type IDENTITY should be used if this column is to be an identity column.

INTEGER
A large integer column. The range of a large integer is -2147483648 to +2147483647.
Note that type IDENTITY should be used if this column is to be an identity column.

BIGINT
A big integer column. The range of a big integer is -9223372036854775808 to +9223372036854775807.
Note that type IDENTITY should be used if this column is to be an identity column.

REAL
A single precision (32 bit) floating-point column.

DOUBLE
A double precision (64 bit) floating-point column.

DECIMAL
A packed decimal fixed-point column.
Note that type IDENTITY should be used if this column is to be an identity column.

DECFLOAT
A decimal floating-point column. Precision may be 16 or 34.

DATE
A 3-part (year, month and day) Gregorian date column.

TIME
A 3-part (hour, minute and second) 24-hour clock time of day column.

TIMESTMP
A 7-part (year, month, day, hour, minute, second and microsecond) date and time column.
Note that type RCT should be used if this column is to be a row change timestamp column.

XML
A well-formed XML document column.

ROWID
A unique, DB2 maintained table row identifier column.

Validation checking will return an error if more than one ROWID column definition exists and has been selected.

SECURITY
A security label column. Requires that the RACF SECLABEL class is active and that, on executing the generated SQL statement, its primary authorization ID must have a valid security label.

On selecting type SECURITY, the table row is zoomed so that the underlying CHARACTER data type of fixed length 8 may be selected as either the built-in type or an equivalent distinct type.

Validation checking will return an error if more than one SECURITY column definition exists and has been selected.

Corresponds to SQL CREATE TABLE syntax AS SECURITY LABEL.

RCT
A row change timestamp column. The value of this column within an individual DB2 table row is updated by DB2 with the timestamp at which the table row is inserted or any value within it gets updated.

On selecting type RCT, the table row is zoomed so that the underlying TIMESTAMP data type may be selected as either the built-in type or an equivalent distinct type. Similarly, to set other options relevant to row change timestamp management.

Validation checking will return an error if more than one RCT column definition exists and has been selected.

Corresponds to SQL CREATE TABLE syntax `FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP.`

IDENTITY
An identity column. A column in which an integer value is automatically generated for each inserted DB2 table row. The generated integer value is determined by a defined sequence.

On selecting type IDENTITY, the table row is zoomed so that the underlying SMALLINT, INTEGER, BIGINT or DECIMAL (scale zero) data type may be selected as either the built-in type or an equivalent distinct type. Similarly, to set the sequence parameters and other options relevant to identity column management.

Validation checking will return an error if more than one IDENTITY column definition exists and has been selected.

Corresponds to SQL CREATE TABLE syntax AS IDENTITY.

DISTINCT
A user-defined distinct type column which is based on one of the built-in data types.
Note that type IDENTITY, SECURITY, RCT or ROWID should be used if this column is to be an identity column, security label, row change timestamp or ROWID column respectively.

Default Value/FIELDPROC Name> (Default Value / Field Proc)
Applicable only for Null value / Default value / Field Proc combination codes DD, DF, ND and NF, this field identifies either the default value constant or the field procedure name as applicable. For all other codes, the contents of this field is ignored.

If a default value is to be used which is not the default for the column data type, the session user name or current SQLID, then code DD or ND must be specified and the columns default value constant specified in this field. If the column is a character data type, then the default value must be enclosed within SQL delimiter characters as defined for the DB2 server. If these character string delimiters have been omitted, then they will be added automatically by the panel, escaping any delimiter character that exists as data within the string.

Where a field procedure is to be used, then code DF or NF must be specified and the field procedure name specified in this field.

Field procedure parameters may be specified by the Column Definition - FIELDPROC Parameters sub-panel opened on selecting *FieldProc Parms* in table view or selecting the Manage FIELDPROC Parms> option in zoomed view. Alternatively, a comma separated list of procedure parameters may be specified in parentheses immediately following the field procedure name within this field. Specification of parameters in this field will override any parameters specified in the sub-panel.

Validation checking is performed on this field and an error returned if the default value constant does not adhere to the column type and length or if the field procedure name and its parameteres are invalid.

Corresponds to SQL CREATE TABLE syntax FIELDPROC *program-name*.

DB2 Generated Values>
Applicable to columns of type IDENTITY, RCT and ROWID only, this option specifies whether, on insert of a table row, DB2 will always generate the column value or generate one only if no other value is specified.

Not that DB2 will always generate values on an update operation if the DEFAULT clause is specified.

Corresponds to SQL CREATE TABLE syntax GENERATED ALWAYS and GENERATED BY DEFAULT.

Distinct Type Fields
For column types DISTINCT, IDENTITY, SECURITY, RCT and ROWID a user-defined distinct type may be specified. Fields that follow relate specifically to use of a column distinct type.

On selecting type DISTINCT or selecting distinct type as the source specification for IDENTITY, SECURITY, RCT and ROWID, the Select a Distinct Type panel is opened displaying a selectable list of distinct types defined in the DB2 server.

This selection list includes only those distinct types that have a source type which is valid for the column type being defined. e.g. A distinct type used as the source of a column of type SECURITY must itself have a source type of CHARACTER(8). If no distinct types exist that satisfy the column type, then only error ZZSP077E is displayed without displaying the selection list.

Corresponds to SQL CREATE TABLE syntax *distinct-type-name*.

Distinct Type Schema> (DistSchema)
Specifies the schema of the required distinct type.

The distinct schema may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the schema. If no schema is specified, then wildcard character "%" is implied.

Distinct Type Name> (DistName)
Specifies the name of the required distinct type.

The distinct name may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the name. If no name is specified, then wildcard character "%" is implied.

If more than one distinct type matches the combined schema and name wildcard specification, the Select a Distinct Type panel is opened.

Source Type:
A non-enterable field displaying the built-in source type of the distinct type.

Len/Pr:
A non-enterable field displaying the length or precision assigned to the distinct type.

Sc:
A non-enterable field displaying the scale value assigned to the distinct type.

SubType:
A non-enterable field displaying the character subtype assigned to the distinct type. Possible values are blank (for non-character types), S (SBCS), M (Mixed) or B (Bit).

Hidden> (Hide)
Indicates that the column is to be implicitly hidden so that it is not visible in the result for SQL statements unless it is explicitly referenced by name.

Corresponds to SQL CREATE TABLE syntax IMPLICITLY HIDDEN.

Identity Column Options
Applicable only to column type IDENTITY, these fields combine to define a sequence of integer values so that, for each newly inserted table row, this column will be populated with the next value in the sequence.

An IDENTITY column can only contain numeric integer values, therefore the underlying built-in data type can be SMALLINT, INTEGER, BIGINT or DECIMAL (with scale 0).

Note that identity column values need not be unique unless a unique index is defined which references only the identity column.

Use start value>
Enter "/" (slash) or any non-blank character in this field to indicate that the specified start value is to be used. The start value identifies the first value in the sequence and is a positive or negative numeric integer value that satisfies the column's underlying build-in data type.

In an ascending sequence, the default start value is the specified minimum value or 1 if no minimum value is defined. In a descending sequence, the default start value is the specified maximum value or -1 if no maximum value is defined.

Corresponds to SQL CREATE TABLE syntax `START WITH` *`numeric-constant`*.

Increment By value>
Specifies an integer constant that defines the interval between consecutive values in the sequence. A zero or positive value will define an ascending sequence, a negative value will define a descending sequence.

Corresponds to SQL CREATE TABLE syntax `INCREMENT BY` *`numeric-constant`*.

Use minimum value>
Enter "/" (slash) or any non-blank character in this field to indicate that the specified minimum value is to be used. The minimum value is a positive or negative numeric integer value that satisfies the column's underlying build-in data type and is less than or equal to the maximum value.

For an ascending sequence, the minimum value is the value to which the sequence will cycle on reaching the maximum value. For a descending sequence, the minimum value is the value at which the sequence will either cycle or stop generating values.

In an ascending sequence, the default minimum value is the specified start value or 1 if no start value is defined. In a descending sequence, the default minimum value is the minimum value for the column's underlying built-in data type.

Corresponds to SQL CREATE TABLE syntax `NO MINVALUE` and `MINVALUE` *`numeric-constant`*.

Use maximum value>
Enter "/" (slash) or any non-blank character in this field to indicate that the specified maximum value is to be used. The maximum value is a positive or negative numeric integer value that satisfies the column's underlying build-in data type and is greater than or equal to the minimum value.

For an ascending sequence, the maximum value is the value at which the sequence will either cycle or stop generating values. For a descending sequence, the maximum value is the value to which the sequence will cycle on reaching the minimum value.

In an ascending sequence, the default maximum value is the maximum value for the column's underlying built-in data type. In a descending sequence, the default maximum value is the specified start value or -1 if no start value is defined.

Corresponds to SQL CREATE TABLE syntax `NO MAXVALUE` and `MAXVALUE` *`numeric-constant`*.

Cycle values at Max/Min Value.
Enter "/" (slash) or any non-blank character in this field to indicate that sequence values are to continue to be generated after reaching the maximum value (ascending sequence) or minimum value (descending sequence).

If this option is set, an ascending sequence on reaching the maximum value will cycle and continue the sequence from the minimum value. Similarly, a descending sequence on reaching the minimum value will cycle and continue the sequence from the maximum value.

Corresponds to SQL CREATE TABLE syntax `NO CYCLE` and `CYCLE`.

Generate values in order of request.
Enter "/" (slash) or any non-blank character in this field to indicate that sequence values are to be generated in strict sequential order when this identity column is shared between multiple DB2 members. Selecting this option may disable the caching of sequence values.

Corresponds to SQL CREATE TABLE syntax `NO ORDER` and `ORDER`.

Preallocate sequence values and store in memory.
Enter "/" (slash) or any non-blank character in this field to indicate that a number of sequence values are to be generated and cached in memory to improve performance of multiple table row insert.

Corresponds to SQL CREATE TABLE syntax `NO CACHE` and `CACHE`.

#Cached values>
Specifies the maximum number of identity column sequence value that may be preallocated and cached in memory. The minimum value is 2.

Corresponds to SQL CREATE TABLE syntax `CACHE` *`integer-constant`*.

Length>/Precision> (Len/Pr)
For columns of character, graphic character and binary types, this field displays as Length> in zoomed views and defines the length of the data within the column.

For columns of type DECIMAL and DECFLOAT, this field displays as Precision> in zoomed views and defines the precision (number of decimal digits) represented by the decimal value.

For all other column types, this field is not displayed in zoomed view and, in table view, displays the internal column length for the chosen data type.

For LOB types only, the length value may be suffixed with K, M or G representing a value wich is a multiple of 1024, 1048576 and 1073741824 respectively.

The range of allowable values for a length field is displayed in parentheses and depends on the column type and, for variable length fields, the maximum record size and presence of an EDIT procedure.

Manage FIELDPROC Parms> (FieldProc Parms)
Applicable only for Null value / Default value / Field Proc combination codes DF and NF, this field determines display of the Column Definition - FIELDPROC Parameters sub-panel for this column definition. For all other codes, any entries in this sub-panel will be ignored.

In table view, the number of selected parameters is displayed. Position the cursor on the FieldProc Parms table column entry and press <Enter> or, if configured, double-click the left mouse button to dispaly the sub-panel.

In zoomed view, enter "/" (slash) or any non-blank character in this field and press <Enter> to dispaly the sub-panel.

Corresponds to SQL CREATE TABLE syntax FIELDPROC *program-name*(*constant*, ..) .

NULL Value / Default Value / Field Procedure: (Def)
A 2 character code combination representing support for null values, default value specification and/or use of a field procedure.

Not all of the following code combinations are valid for the different columnn types. In table view, if an invalid code is selected for the column type, an error is returned, the table row is zoomed and the field is set to DX by default.

For type DISTINCT, the generated SQL CREATE TABLE statement will use the distinct type schema and name as the cast function on any default values.

Enter a blank or invalid value in this field to display a complete scrollable list of selectable code entries for this field.

DX (DX: Allow Nulls, No Default Value.)
Null values are permitted.
No default value or field procedure is used.

DN (DN: Allow Nulls, Use Default NULL.)
Null values are permitted.
A default value of NULL is used in the absence of a specified value when a row is inserted, updated or loaded. (WITH DEFAULT NULL)

DD (DD: Allow Nulls, Use Default Value.)
Null values are permitted.
A default value equal to the default value for the column type or, if specified, the value specified in Default Value/FIELDPROC Name is used in the absence of a specified value when a row is inserted, updated or loaded. (WITH DEFAULT)

DU (DU: Allow Nulls, Use Default USER.)
Null values are permitted.
A default value equal to the prevailing value of the special register SESSION_USER is used in the absence of a specified value when a row is inserted, updated or loaded. (WITH DEFAULT USER)

DS (DS: Allow Nulls, Use Default SQLID.)
Null values are permitted.
A default value equal to the prevailing value of the special register CURRENT SQLID is used in the absence of a specified value when a row is inserted, updated or loaded. (WITH DEFAULT CURRENT SQLID)

DF (DF: Allow Nulls, Use Field Procedure.)
Null values are permitted.
The field procedure specified in Default Value/FIELDPROC Name is used to encode/decode column data. (FIELDPROC)

NX (NX: Disallow Nulls, No Default Value.)
Null values are **not** permitted. (NOT NULL)
No default value or field procedure is used.

ND (ND: Disallow Nulls, Use Default Value.)
Null values are **not** permitted. (NOT NULL)
A default value equal to the default value for the column type or, if specified, the value specified in Default Value/FIELDPROC Name is used in the absence of a specified value when a row is inserted, updated or loaded. (WITH DEFAULT)

NU (NU: Disallow Nulls, Use Default USER.)
Null values are **not** permitted. (NOT NULL)
A default value equal to the prevailing value of the special register SESSION_USER is used in the absence of a specified value when a row is inserted, updated or loaded. (WITH DEFAULT USER)

NS (NS: Disallow Nulls, Use Default SQLID.)
Null values are **not** permitted. (NOT NULL)
A default value equal to the prevailing value of the special register CURRENT SQLID is used in the absence of a specified value when a row is inserted, updated or loaded. (WITH DEFAULT CURRENT SQLID)

NF (NF: Disallow Nulls, Use Field Procedure.)
Null values are **not** permitted. (NOT NULL)
The field procedure specified in Default Value/FIELDPROC Name is used to encode/decode column data. (FIELDPROC)

Corresponds to SQL CREATE TABLE syntax NOT NULL, WITH DEFAULT and FIELDPROC.

Scale> (Sc)
Applicable only to columns of type DECIMAL, this field defines the scale (number of fractional digits) represented by the decimal value. This value must be less than or equal to the precision value.

For all other column types, this field is not displayed in zoomed view and any value entered in this field in table view will be ignored.

Corresponds to SQL CREATE TABLE syntax DECIMAL(*integer*,*integer*).

Sel> (Sel)
Enter "/" (slash) or any non-blank character in this field to include (select) the column definition in the create table syntax. A blank in this field will exclude (deselect) the column definition.

Deselecting a column definition in table view has the same effect as executing the table primary command EXCLUDE (or line command "X") to exclude the table row.

Table command SELECTALL (assigned to <F6>) will toggle between selecting and deselecting all table rows (column definitions).

SubType> (SubType)
Applicable only to columns of character types CHAR, VARCHAR and CLOB, this field specifies the character data subtype.

Enter a blank or invalid value in this field to display a complete list of selectable subtype entries for this field.

SBCS
Specifies that the column will contain single byte data.

MIXED
Specifies that the column will contain both single and double byte data. This option is valid only if the table encoding is Unicode or if the DB2 server supports double and mixed byte character sets.

BIT
Valid only for column type CHAR and VARCHAR, specifies that the column will contain non-printable bit data.

Corresponds to SQL CREATE TABLE syntax FOR SBCS|MIXED|BIT DATA.

Name:
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

Table Owner:
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

Column Definition - FIELDPROC Parameters

The Column FIELDPROC Parameters sub-panel is displayed on selecting **Manage FIELDPROC Parms>** or table view entry **FieldProc Parms** from the Column Definitions sub-panel.

This sub-panel contains an embedded table of DB2 table column field procedure parameters. Standard table edit primary and line commands must be used to INSERT, DELETE, COPY, MOVE or EXCLUDE table rows as appropriate and to scroll the table display UP, DOWN, LEFT and RIGHT.

Entries in this table are ignored if a parenthesised list of comma separated constant values are included in the column definitions panel field Default Value/FIELDPROC Name>

Each parameter to be passed to the field procedure must be entered, one in each table row in the required order. The defined SQL delimiter character quote (") or apostrophe (') should be used to enclose values as appropriate.

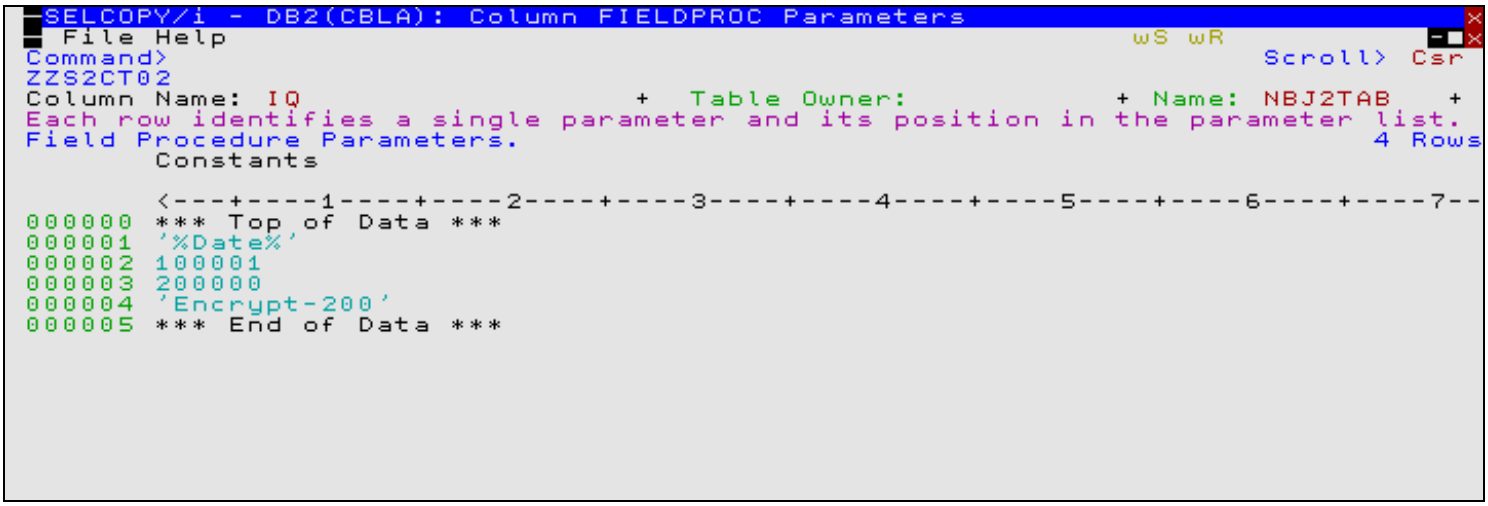


Figure 217. DB2: Column FIELDPROC Parameters.

Primary Key Definition

The Primary Key Definition sub-panel is displayed on selecting **Primary Key Constraint** from the Columns & Constraints panel view.

This sub-panel contains an embedded table of DB2 table primary key column definitions. Standard table edit primary and line commands must be used to INSERT, DELETE, COPY, MOVE or EXCLUDE table rows as appropriate, to scroll the table display UP, DOWN, LEFT and RIGHT and also to ZOOM the display of an individual table row.

In addition to the standard primary table commands, this sub-panel also supports the following primary commands:

- **SELECTAll** (assigned to <F6> by default). This toggles between selection and deselection of all entries in the table.
- **RESTORE** (assigned to <F5> by default). This restores the table entries so that all rows that satisfy the filter criteria are redisplayed. i.e. reset the table view.
  RESTORETAB may be used to include all eligible columns which have been defined or updated since this primary key constraint sub-panel was last displayed.

Each table row identifies a column definition entered in the Column Definitions sub-panel, which satisfies the primary key column criteria and is, therefore, eligible for selection as a primary key column. Apart from Sel entry selection field, all fields in this table are non-enterable and are included for information only.

Select and/or exclude the DB2 column definitions then, if necessary, rearrange them using line command "M" or "MM" so that they occur in the order required for the primary key constraint definition.

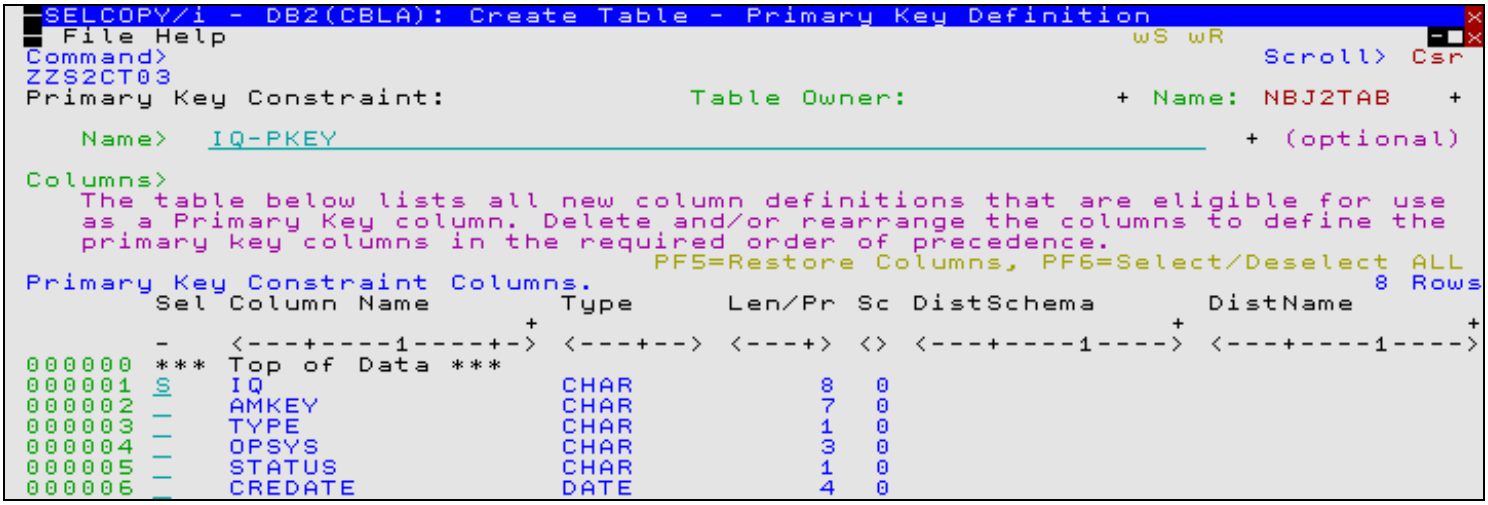


Figure 218. DB2: Create Table - Primary Key Definition.

Primary Key Definition - Panel Fields

Table Owner:
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

Name:
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

Name>
Optionally specifies the name of the primary key constraint which is automatically upper cased. If not specified, DB2 will generate a unique name for the constraint.

A constraint name has a maximum length of 128 characters.

Sel
Enter "/" (slash) or any non-blank character in this field to include (select) the column definition in the create table syntax. A blank in this field will exclude (deselect) the column definition.

Deselecting a column definition in table view has the same effect as executing the table primary command EXCLUDE (or line command "X") to exclude the table row.

Column Name
A non-enterable field displaying the name of the column.

Type
A non-enterable field displaying the type of the column. See Column Type for valid column types.

Len/Pr
A non-enterable field displaying the length or precision of the column data.

Scale
A non-enterable field displaying the scale of the column data.

DistSchema
A non-enterable field displaying the schema of the distinct type assigned to the column. If no distinct type is used, this field will be blank.

DistName
A non-enterable field displaying the name of the distinct type assigned to the column. If no distinct type is used, this field will be blank.

Unique Constraint Definitions

The Unique Constraint Definitions sub-panel is displayed on selecting **Unique Constraints** from the Columns & Constraints panel view.

This sub-panel contains an embedded table of DB2 table unique constraint definitions. Standard table edit primary and line commands must be used to INSERT, DELETE, COPY, MOVE or EXCLUDE table rows as appropriate and to scroll the table display UP, DOWN, LEFT and RIGHT.

In addition to the standard primary table commands, this sub-panel also supports primary command **SELECTAll** (assigned to <F6> by default) to toggle between selection and deselection of all entries in the table.

Insert and select table rows to add new unique constraints to the table definition, then, for each new constraint, open the Unique Constraint Key Columns sub-panel to define the unique key columns.

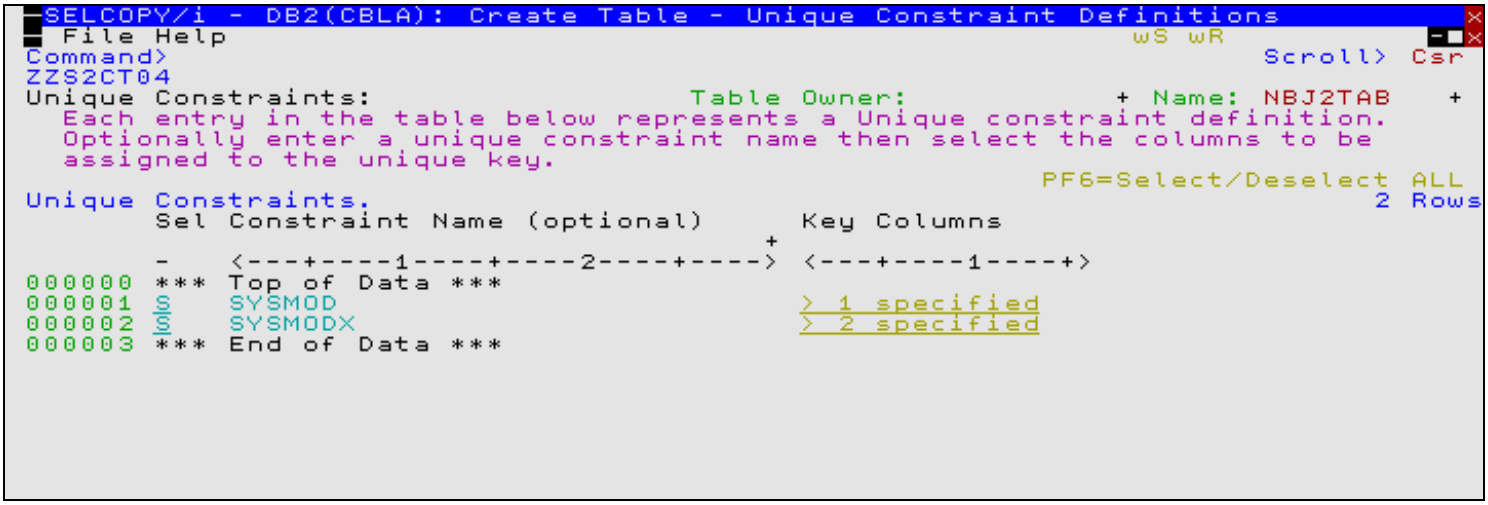


Figure 219. DB2: Create Table - Unique Constraint Definitions.

Unique Constraint Definitions - Panel Fields

`Table Owner:`
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

`Name:`
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

`Sel> (Sel)`
Enter "/" (slash) or any non-blank character in this field to include (select) the unique constraint definition in the create table syntax. A blank in this field will exclude (deselect) the constraint definition.

Deselecting a constraint definition in table view has the same effect as executing the table primary command EXCLUDE (or line command "X") to exclude the table row.

`Constraint Name`
Optionally specifies the name of the unique constraint which is automatically upper cased. If not specified, DB2 will generate a unique name for the constraint.

A constraint name has a maximum length of 128 characters.

`Key Columns`
Displays the number of selected key columns. Position the cursor in the row containing the required unique constraint definition and press <Enter> or, if configured, double-click the left mouse button to display the Unique Constraint Key Columns sub-panel.

Unique Constraint Key Columns

The Unique Constraint Key Columns sub-panel is displayed on selecting a table row entry in the Unique Constraint Definitions sub-panel.

This sub-panel contains an embedded table of DB2 table unique constraint key column definitions. Standard table edit primary and line commands must be used to INSERT, DELETE, COPY, MOVE or EXCLUDE table rows as appropriate, to scroll the table display UP, DOWN, LEFT and RIGHT and also to ZOOM the display of an individual table row.

In addition to the standard primary table commands, this sub-panel also supports the following primary commands:

- **SELECTAll** (assigned to <F6> by default). This toggles between selection and deselection of all entries in the table.
- **RESTORE** (assigned to <F5> by default). This restores the table entries so that all rows that satisfy the filter criteria are redisplayed. i.e. reset the table view.

Each table row identifies a column definition entered in the Column Definitions sub-panel, which satisfies the unique key column criteria and is, therefore, eligible for selection as a unique key column. Apart from Sel entry selection field, all fields in this table are non-enterable and are included for information only.

Select and/or exclude the DB2 column definitions then, if necessary, rearrange them using line command "M" or "MM" so that they occur in the order required for the unique constraint definition.

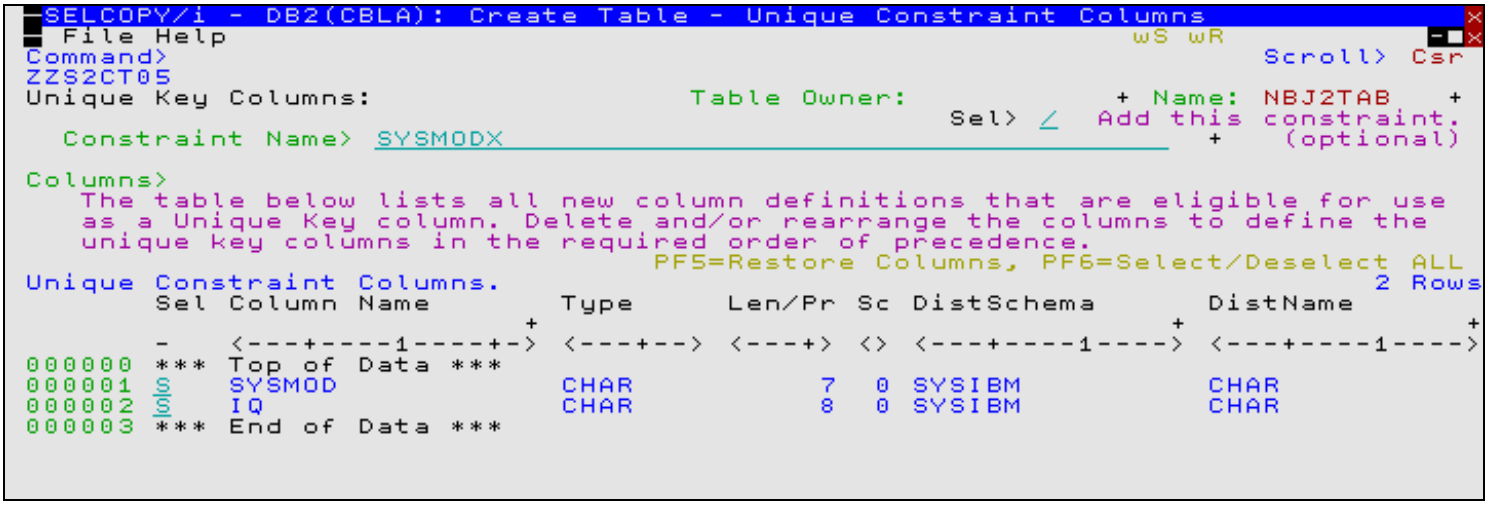


Figure 220. DB2: Create Table - Unique Constraint Columns.

Unique Constraint Key Columns - Panel Fields

Table Owner:
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

Name:
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

Sel>
Optionally enter blank to deselect this unique constraint definition overriding the selection made in the **Unique Constraint Definitions** panel.

Constraint Name>
Optionally override the name of the unique constraint entered in the **Unique Constraint Definitions** panel.

Sel
Enter "/" (slash) or any non-blank character in this field to include (select) the column definition in the create table syntax. A blank in this field will exclude (deselect) the column definition.

Deselecting a column definition in table view has the same effect as executing the table primary command EXCLUDE (or line command "X") to exclude the table row.

Column Name
A non-enterable field displaying the name of the column.

Type
A non-enterable field displaying the type of the column. See Column Type for valid column types.

Len/Pr
A non-enterable field displaying the length or precision of the column data.

Scale
A non-enterable field displaying the scale of the column data.

DistSchema
A non-enterable field displaying the schema of the distinct type assigned to the column. If no distinct type is used, this field will display "SYSIBM" or blanks.

DistName
A non-enterable field displaying the name of the distinct type assigned to the column. If no distinct type is used, this field will display the built-in data type or blanks.

Referential Constraint Definitions

The Referential Constraint Definitions sub-panel is displayed on selecting **Referential Constraints** from the Columns & Constraints panel view.

This sub-panel contains an embedded table of DB2 table referential constraint definitions. Standard table edit primary and line commands must be used to INSERT, DELETE, COPY, MOVE or EXCLUDE table rows as appropriate and to scroll the table display UP, DOWN, LEFT and RIGHT.

In addition to the standard primary table commands, this sub-panel also supports primary command **SELECTAll** (assigned to <F6> by default) to toggle between selection and deselection of all entries in the table.

Insert and select table rows to add new referential constraints to the table definition. For each new constraint, first open the Referential Constraint Parent Key Columns sub-panel to identify the parent DB2 table and parent key columns, then open the Referential Constraint Foreign Key Columns sub-panel to select columns from the new table definition that will constitute the corresponding foreign key.

On exiting the Column Definitions sub-panel, validation is performed for all selected referential constraint definitions. If no parent key and/or foreign key columns have been selected or the number of key columns do not match, then the appropriate sub-panel is opened and an error message returned.

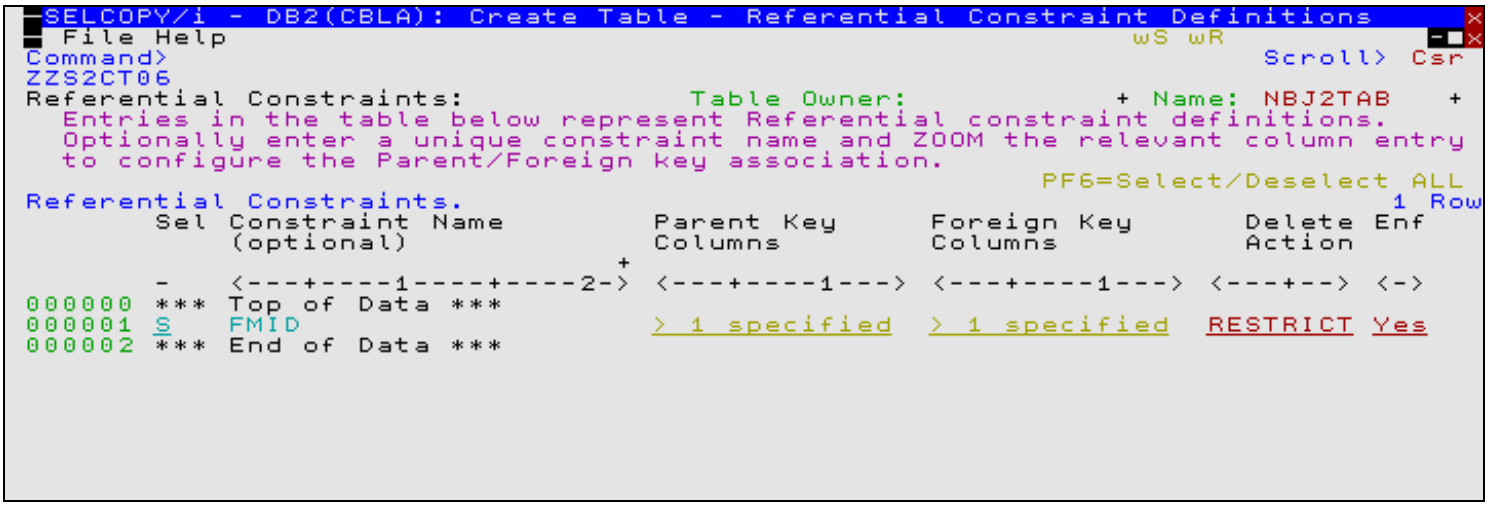


Figure 221. DB2: Create Table - Referential Constraint Definitions.

Referential Constraint Definitions - Panel Fields

Field names that follow are as appear in the zoomed view of a table row. Names in parentheses correspond to the equivalent column name in table view.

`Table Owner:`
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

`Name:`
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

`Sel> (Sel)`
Enter "/" (slash) or any non-blank character in this field to include (select) the referential constraint definition in the create table syntax. A blank in this field will exclude (deselect) the constraint definition.

Deselecting a constraint definition in table view has the same effect as executing the table primary command EXCLUDE (or line command "X") to exclude the table row.

`Constraint Name`
Optionally specifies the name of the referential constraint which is automatically upper cased. If not specified, DB2 will generate a unique name for the constraint.

A constraint name has a maximum length of 128 characters.

`Action> 1. Select Parent Key Columns (Parent Key Columns)`
In zoomed view, enter "1" in the Action> field and press <Enter> to display the Referential Constraint - Parent Key Columns sub-panel for this referential constraint definition.

In table view, the number of selected parent key columns is displayed. Position the cursor on the Parent Key Columns entry of the required table row and press <Enter> or, if configured, double-click the left mouse button to dispaly the sub-panel.

Corresponds to SQL CREATE TABLE syntax REFERENCES *table-name* (*column-name*, ..) .

`Action> 2. Select Foreign Key Columns (Foreign Key Columns)`
In zoomed view, enter "2" in the Action> field and press <Enter> to display the Referential Constraint - Foreign Key Columns sub-panel for this referential constraint definition.

In table view, the number of selected foreign key columns is displayed. This number should match the number of selected parent key columns. Position the cursor on the Foreign Key Columns entry of the required table row and press <Enter> or, if configured, double-click the left mouse button to dispaly the sub-panel.

Corresponds to SQL CREATE TABLE syntax FOREIGN KEY (*column-name*, ..) .

Action on DELETE of Parent Table row: (Delete Action)
Select the action performed on delete of a row in the parent table which has a referential constraint relationship with a row in the new DB2 table (i.e. delete rule).

In table view, enter a blank or invalid value in this field to display a scrollable list of valid, selectable entries for this field. In zoomed view, enter "/" (slash) or any non-blank character to select one of the mutually exclusive options.

1. **Action defined by CURRENT RULES special register.** (Default)
   The default action if no explicit delete rule is defined.
2. **Delete fails if dependent row exists - check performed immediately.** (RESTRICT)
   Corresponds to SQL CREATE TABLE syntax ON DELETE RESTRICT.
3. **Delete fails if dependent row exists - check performed at end of DELETE.** (NOACT)
   Corresponds to SQL CREATE TABLE syntax ON DELETE NO ACTION.
4. **Dependent rows are deleted.** (CASCADE)
   Corresponds to SQL CREATE TABLE syntax ON DELETE CASCADE.
5. **Nullable columns in the foreign key are set to NULL.** (SETNULL)
   Corresponds to SQL CREATE TABLE syntax ON DELETE SET NULL.

DB2 Enforced Constraint: (Enf)
Select whether or not the referential constraint is to be enforced by DB2 during normal operations, such as insert, update, or delete.

In table view, enter a blank or invalid value in this field to display a scrollable list of valid, selectable entries for this field (Yes or No). In zoomed view, enter "/" (slash) or any non-blank character to indicate that the constraint is to be enforced by DB2, otherwise enter blank.

Corresponds to SQL CREATE TABLE syntax ENFORCED or NOT ENFORCED.

Referential Constraint - Parent Key Columns

The Referential Constraint Parent Key Columns sub-panel is displayed on selecting Action 1 or **Parent Key Columns** in the Referential Constraint Definitions sub-panel. Alternativley, following validation if no parent key columns have been selected.

This sub-panel contains an embedded table of DB2 table referential constraint parent key column definitions. The table is browsed so that its contents may not be manipulated using the standard table editing techniques.

The sub-panel also contains fields for parent table name input and/or the name of a unique constraint. Entering the name of a primary key or unique constraint will automatically populate the embedded table with the constraint's associated column names in order of sequence. Since the parent key must be a primary or unique key of the parent table, this ensures that the parent key columns are valid.

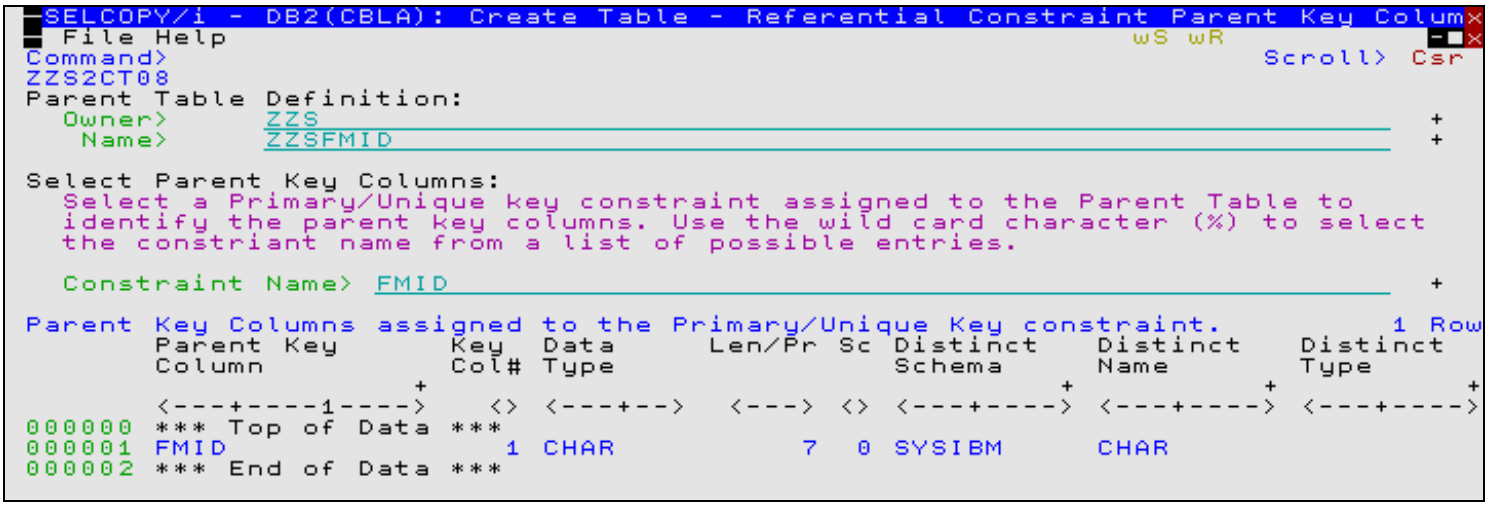


Figure 222. DB2: Create Table - Referential Constraint Parent Key Columns.

Referential Constraint - Parent Key Columns - Panel Fields

Parent Table Definition:
Fields that together identify the parent table to be used in the referential constraint relationship.

These field values may include wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the parent table owner/name specification. If this is the case, a Select Table panel is displayed containing a list of eligible DB2 tables which satisfy the table owner/name filter. Use prefix command "S", or press the <Enter> key on the required entry to select it.

The contents of the parent table input fields are also used as filter fields for constraint name selection.

Owner>
The parent DB2 table owner id.

Name>
The parent DB2 table name.

Constraint Name>
The name of a primary key or unique key constraint associated with the parent table definition. The parent table primary/unique key columns become the parent key columns in the referential constraint definition.

The constraint name may include wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the constraint name. If this is the case, the Select Unique Key Constraint panel is displayed containing a list of eligible primary and unique key constraint definitions which satisfy the combined parent table owner/name and constraint name filter. Note that, if a parent table is not specified or contains wild card characters, then the list will potentially contain constraints from more than one DB2 table.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

Selecting an entry from this list will populate the parent table owner and name fields and also insert the associated key column names and their attributes in the Parent Key Columns table in order of sequence.

Parent Key Columns assigned to the Primary/Unique Key constraint
Embedded table containing a row for each column of the selected primary/unique key.

Parent Key Column
A non-enterable field displaying the name of the column.

Key Col#
A non-enterable field displaying the column sequence number within the key.

Data Type
A non-enterable field displaying the type of the column. See Column Type for valid column types.

Len/Pr
A non-enterable field displaying the length or precision of the column data.

Sc
A non-enterable field displaying the scale of the column data.

Distinct Schema
For data type DISTINCT only, a non-enterable field displaying the schema of the distinct type assigned to the column. If no distinct type is used, this field will display "SYSIBM" or blanks.

Distinct Name
For data type DISTINCT only, a non-enterable field displaying the name of the distinct type assigned to the column. If no distinct type is used, this field will display the built-in data type or blanks.

Distinct Type
For data type DISTINCT only, a non-enterable field displaying the source DB2 built-in data type for the distinct type assigned to the column. If no distinct type is used, this field will display blanks.

Referential Constraint - Foreign Key Columns

The Referential Constraint Foreign Key Columns sub-panel is displayed on selecting Action 2 or **Foreign Key Columns** in the Referential Constraint Definitions sub-panel. Alternatively, following validation if no foreign key columns have been selected or the number of selected foreign key columns does not match the number of selected parent keys columns.

This sub-panel contains an embedded table of DB2 table referential constraint parent key column definitions. The table is for update only so that rows may not be manipulated using the standard table editing techniques. However, primary command **RESTORE** (assigned to <F5> by default) may be executed to restore the table entries if required.

One table row exists for each column of the parent key. A foreign key column name from the newly defined table must be entered for each corresponding parent key column.

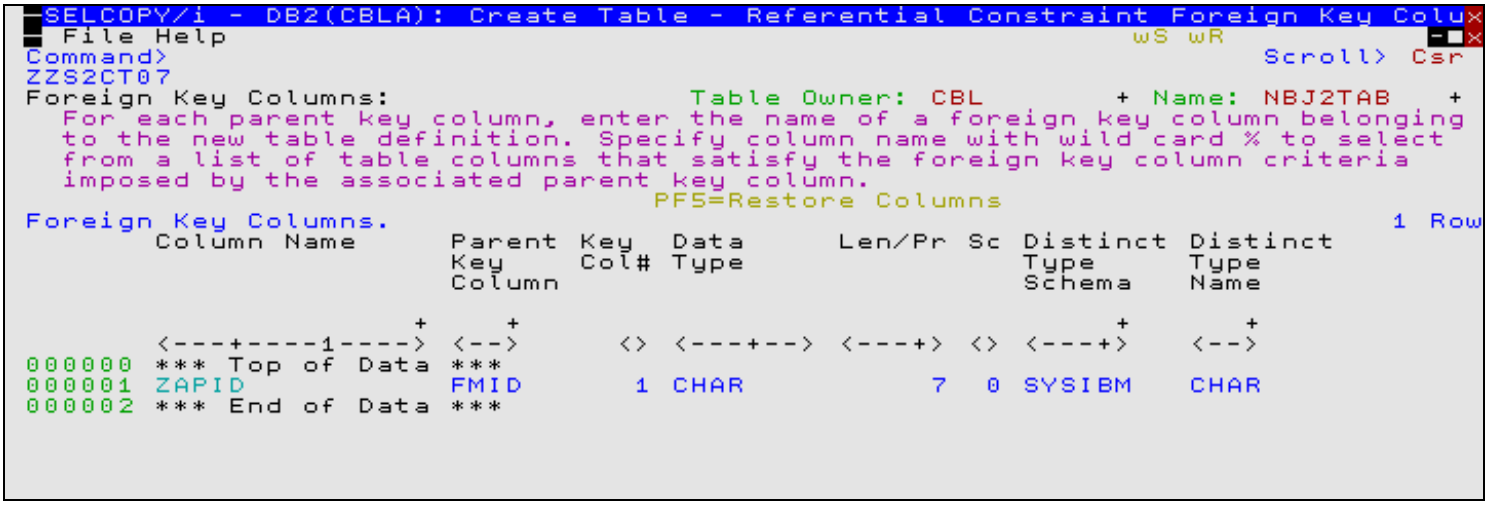


Figure 223. DB2: Create Table - Referential Constraint Foreign Key Columns.

Referential Constraint - Foreign Key Columns - Panel Fields

`Table Owner:`
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

`Name:`
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

`Column Name`
Enter the name of a column in the new DB2 table definition which has the same data type, length/precision and scale attributes as the corresponding parent key column.

The column name may include wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the parent table column name specification. If this is the case, the Select Foreign Key Column panel is displayed containing a list of eligible column names that are defined in the Column Definitions sub-panel and satisfy the column name filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

`Parent Key Column`
A non-enterable field displaying the name of the parent key column.

`Key Col#`
A non-enterable field displaying the column sequence number within the key.

`Data Type`
A non-enterable field displaying the type of the column. See Column Type for valid column types.

`Len/Pr`
A non-enterable field displaying the length or precision of the column data.

`Sc`
A non-enterable field displaying the scale of the column data.

`Distinct Type Schema`
For data type DISTINCT only, a non-enterable field displaying the schema of the distinct type assigned to the column. If no distinct type is used, this field will display "SYSIBM" or blanks.

`Distinct Type Name`
For data type DISTINCT only, a non-enterable field displaying the name of the distinct type assigned to the column. If no distinct type is used, this field will display the built-in data type or blanks.

Check Constraint Definitions

The Check Constraint Definitions sub-panel is displayed on selecting **Check Constraints** from the Columns & Constraints panel view.

This sub-panel contains an embedded table of DB2 table check constraint definitions. Standard table edit primary and line commands must be used to INSERT, DELETE, COPY, MOVE or EXCLUDE table rows as appropriate and to scroll the table display UP, DOWN, LEFT and RIGHT.

Insert and select table rows to add new check constraints to the table definition. Note that no validation is performed for check constraint conditions.

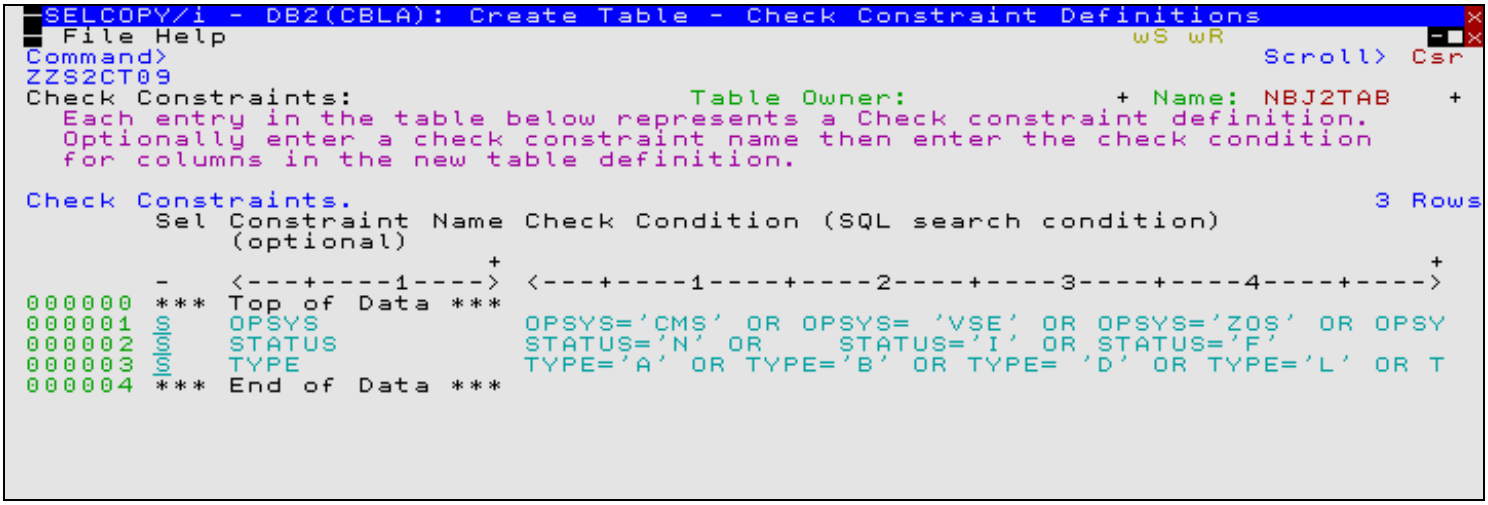


Figure 224. DB2: Create Table - Check Constraint Definitions.

Check Constraint Definitions - Panel Fields

Field names that follow are as appear in the zoomed view of a table row. Names in parentheses correspond to the equivalent column name in table view.

`Table Owner:`
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

`Name:`
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

`Sel> (Sel)`
Enter "/" (slash) or any non-blank character in this field to include (select) the check constraint definition in the create table syntax. A blank in this field will exclude (deselect) the constraint definition.

Deselecting a constraint definition in table view has the same effect as executing the table primary command EXCLUDE (or line command "X") to exclude the table row.

`Constraint Name`
Optionally specifies the name of the check constraint which is automatically upper cased. If not specified, DB2 will generate a unique name for the constraint.

A constraint name has a maximum length of 128 characters.

`Condition> (Check Condition)`
Enter a check condition to apply to columns within the ne DB2 table. A check condition is an SQL search condition.

Note that, if the input field is not sufficiently large enough to type the required check condition, ZOOM the required table row and then EXPAND the panel field.

Model using LIKE

The Model using LIKE panel view is displayed if the user has selected to model the new DB2 table on a single table or view using LIKE in the Table Definition panel view.

This view allows the user to select the source DB2 base table or view from which the new table will be modelled.

Corresponds to SQL CREATE TABLE syntax LIKE *table-name*/*view-name*. See "DB2 for z/OS SQL Reference" for further information on the table attributes that cannot be modelled from another table or view specified using this syntax.

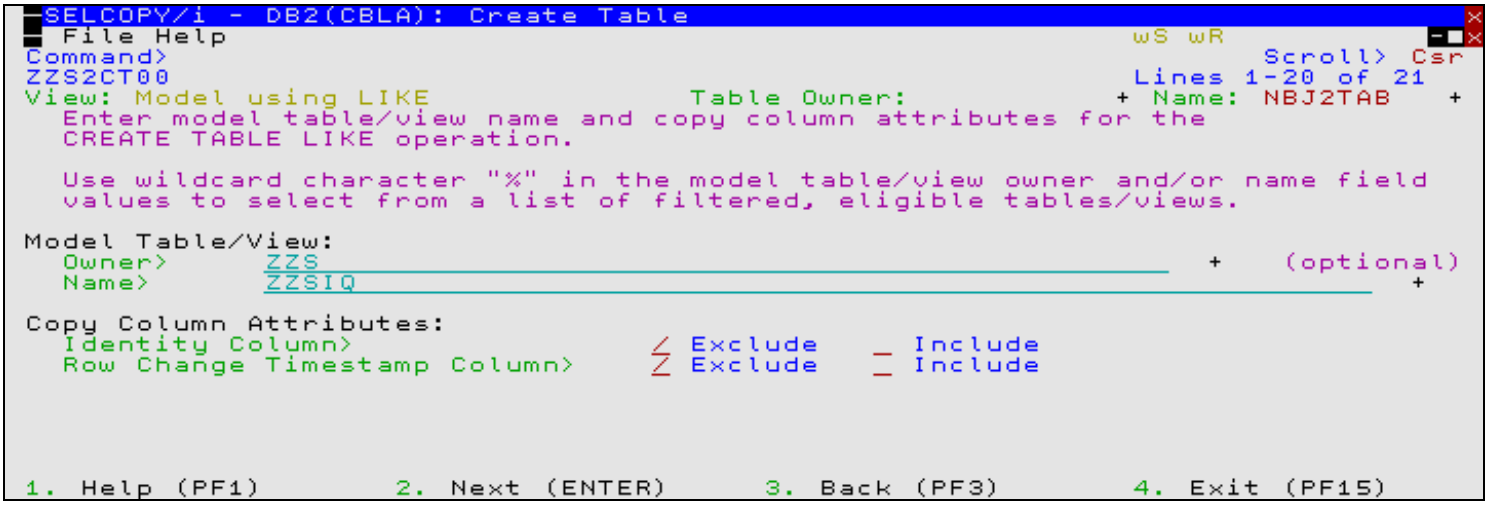


Figure 225. DB2: Create Table - Model using LIKE.

Model using LIKE - Panel Fields

`Table Owner:`
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

`Name:`
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

`Model Table/View:`
Fields that together identify the model DB2 table or DB2 view from which attributes of the new table will be modelled.

These field values may include wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the parent table owner/name specification. If this is the case, a Select Table panel is displayed containing a list of eligible table or view names which satisfy the table owner/name filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

`Owner>`
The model DB2 table or view owner id.

`Name>`
The model DB2 table or view name.

`Copy Column Attributes:`
Attributes of columns belonging to the model DB2 table or view that may be included or excluded when modelling the new DB2 table columns.

`Identity Column>`
Enter "/" (slash) or any non-blank character against the mutually exclusive options (Include or Exclude) to control whether a column in the new table, modelled by an IDENTITY column in the source DB2 table or view, itself becomes an IDENTITY column inheriting the same defined sequence parameters.

Corresponds to SQL CREATE TABLE syntax EXCLUDING or INCLUDING IDENTITY COLUMN ATTRIBUTES.

`Row Change Timestamp Column>`
Enter "/" (slash) or any non-blank character against the mutually exclusive options (Include or Exclude) to control whether a column in the new table, modelled by a ROW CHANGE TIMESTAMP (RCT) column in the source DB2 table or view, itself becomes a ROW CHANGE TIMESTAMP column.

Corresponds to SQL CREATE TABLE syntax EXCLUDING or INCLUDING ROW CHANGE TIMESTAMP COLUMN ATTRIBUTES.

Model on Result Table

The Model on a Result Table panel view is displayed if the user has selected to model the new DB2 table on a generated result table in the Table Definition panel view.

This view allows the user to specify a fullselect SQL query expression to generate a DB2 result table from which the new DB2 table will be modelled.

If column names in the DB2 result table are to be named differently in the new DB2 table definition and the AS clause is not used to name columns in the fullselect clause, then execute primary command COLMAP (assigned to <F5>) to open the Result Table Column Name Remap sub-panel.

Corresponds to SQL CREATE TABLE syntax AS (*fullselect*) WITH NO DATA. See "DB2 for z/OS SQL Reference" for further information on the table attributes that cannot be modelled from a generated result table.

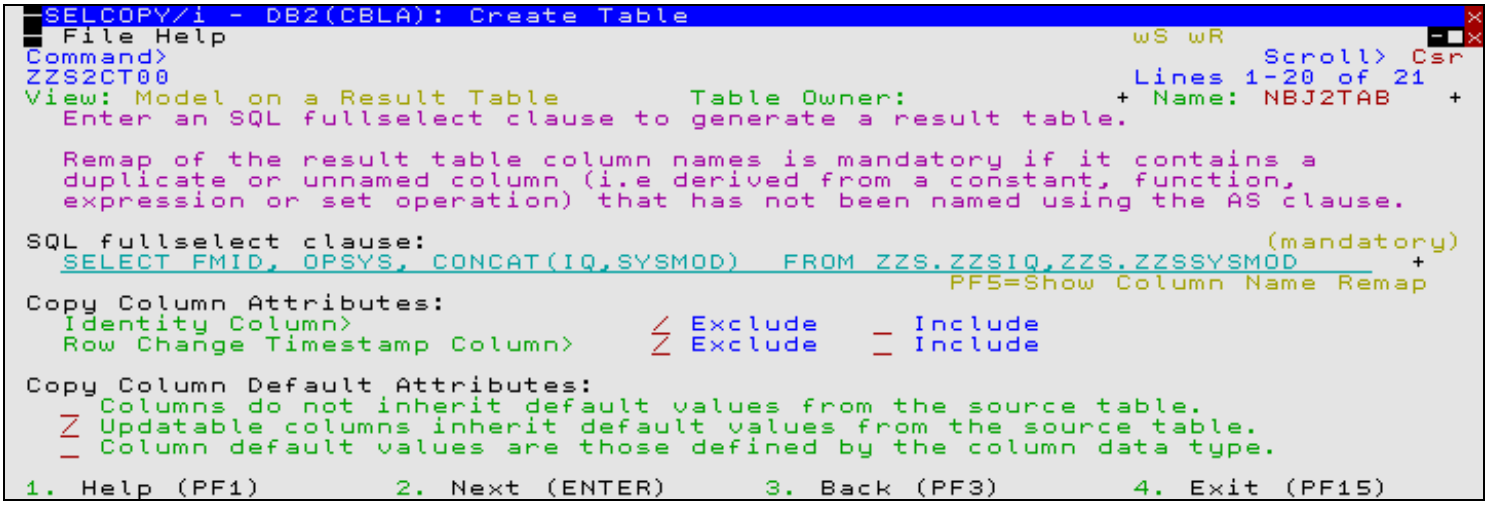


Figure 226. DB2: Create Table - Model on a Result Table.

Model on Result Table - Panel Fields

`Table Owner:`
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

`Name:`
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

`SQL fullselect clause:`
Enter the SQL fullselect query expression. If the size of the input field is not sufficient to enter the complete fullselect clause, then execute primary command EXPAND (default <F14>) to display an editable view of the field.

Note that no validation of the SQL fullselect clause is performed by SELCOPY/i.

`Copy Column Attributes:`
Attributes of columns belonging to the model DB2 result table that may be included or excluded when modelling the new DB2 table columns.

`Identity Column>`
Enter "/" (slash) or any non-blank character against the mutually exclusive options (Include or Exclude) to control whether a column in the new DB2 table, modelled by an IDENTITY column in the DB2 result table, itself becomes an IDENTITY column inheriting the same defined sequence parameters.

Corresponds to SQL CREATE TABLE syntax EXCLUDING or INCLUDING IDENTITY COLUMN ATTRIBUTES.

`Row Change Timestamp Column>`
Enter "/" (slash) or any non-blank character against the mutually exclusive options (Include or Exclude) to control whether a column in the new DB2 table, modelled by a ROW CHANGE TIMESTAMP (RCT) column in the DB2 result table, itself becomes a ROW CHANGE TIMESTAMP column.

Corresponds to SQL CREATE TABLE syntax EXCLUDING or INCLUDING ROW CHANGE TIMESTAMP COLUMN ATTRIBUTES.

`Copy Column Default Attributes:`
Enter "/" (slash) or any non-blank character against one of these mutually exclusive options to control how default values for columns in the new DB2 table are assigned.

- ◊ **Columns do not inherit default values from the source table.**
  Corresponds to SQL CREATE TABLE syntax EXCLUDING COLUMN DEFAULTS.
- ◊ **Updatable columns inherit default values from the source table.**
  Corresponds to SQL CREATE TABLE syntax INCLUDING COLUMN DEFAULTS.
- ◊ **Column default values are those defined by the column data type.**
  Corresponds to SQL CREATE TABLE syntax USING TYPE DEFAULTS.

Result Table Column Name Remap

The Result Table Column Name Remap sub-panel is displayed on executing COLMAP from the Model on Result Table or Materialized Query Table Definition panel view.

This sub-panel contains an embedded table of column names to be assigned, in order of specification, to the columns of the new DB2 table. These are used in place of the names of columns in the generated result table. Column name remap is mandatory if any of the result table columns are unnamed or have a duplicate name. Furthermore, the number of column names specified in this table must equal the number of columns in the result table.

Standard table edit primary and line commands must be used to INSERT, DELETE, COPY, MOVE or EXCLUDE table rows as appropriate and to scroll the table display UP, DOWN, LEFT and RIGHT.

A column name has a maximum length of 30 characters.

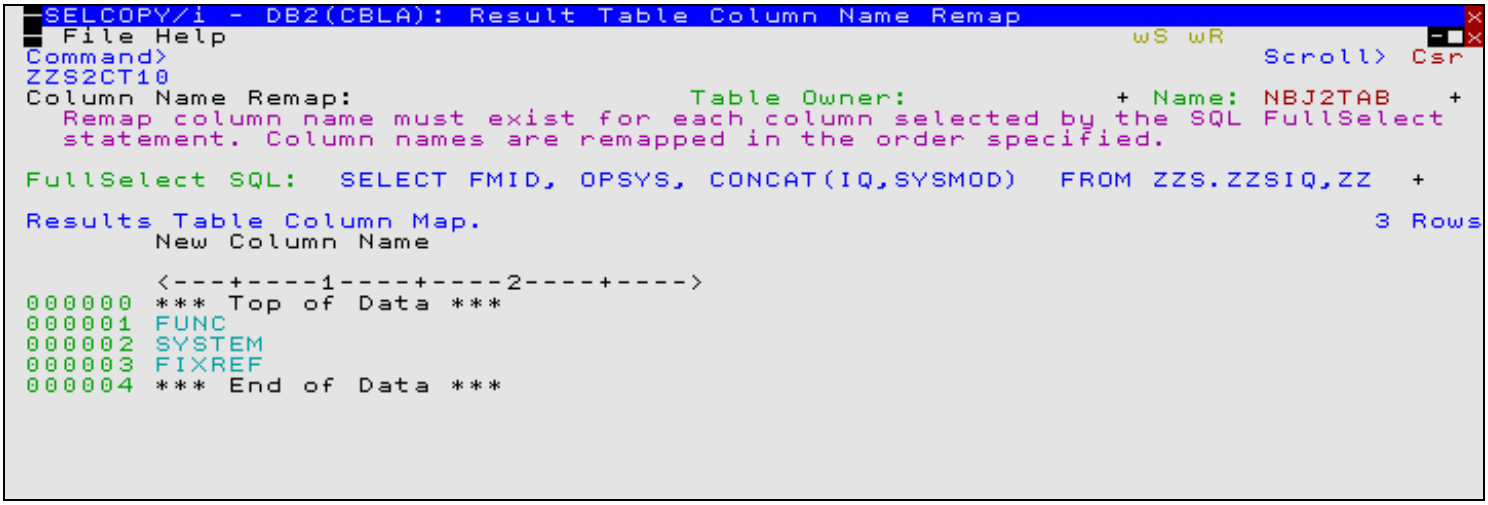


Figure 227. DB2: Result Table Column Name Remap.

Model on Tables/Views (Load Values)

The Table Modelling panel view is the next view displayed if the user has selected to model on one or more tables/views for panel field load in the Table Definition panel view.

Unlike modelling using LIKE or modelling from a result table, this form of modelling will populate the appropriate input fields in the sequence of panel views with table attributes from a specified model DB2 table or view. Furthermore, the model DB2 table may exist at a remote server location.

This view allows the user to select the model DB2 table or view and then select which of the table attributes will be modelled. Modelled column and constraint definitions may be added to definitions that may have already been included in the column and constraint definition panels. These existing definitions may have been entered manually or added by previous modelling performed by this Table Modelling panel view.

The modelling is performed on progressing to the next panel view in the sequence if either of the following is true:

1. It is the first time the Table Modelling panel view has been visited in this sequence of panel views. Note that panel views may be revisited by proceeding backwards through the sequence of panel views.
2. Apart from deselection of a table attribute option field, a change has been made to a field in the Table Modelling panel view. i.e. A change to the model table/view specification, selection of a table attribute option or a change to the modelling action (add or replace).

Once modelling has been performed, the Table Modelling panel view remains in focus to display informational messages and also to allow the user to perform additional modelling on another DB2 table or view if required.

Whether or not modelling is performed, the Columns & Constraints panel view will be displayed in the sequence of panel views, allowing the user to view, deselect, add to, update or delete the modelled column and constraint definitions.

Note that this form of modelling does not yet support modelling of the following table attributes:

1. IDENTITY column sequence parameters. Column definitions, modelled from columns defined as being identity columns, are assigned a column type of IDENTITY but do not inherit the identity sequence used by the model identity column.

   The identity column input field values that define the identity sequence will be set to their default values. See Identity Column Options for the Column Definitions sub-panel.

2. Range partitioned tablespace table partitioning columns and partition limit values.

   This would only be applicable where the model is a DB2 table defined in a partitioned by range tablespace and the new DB2 table is to be defined in an implicitly defined tablespace or one which is explicitly defined with the same number of partitions as the model tablespace.

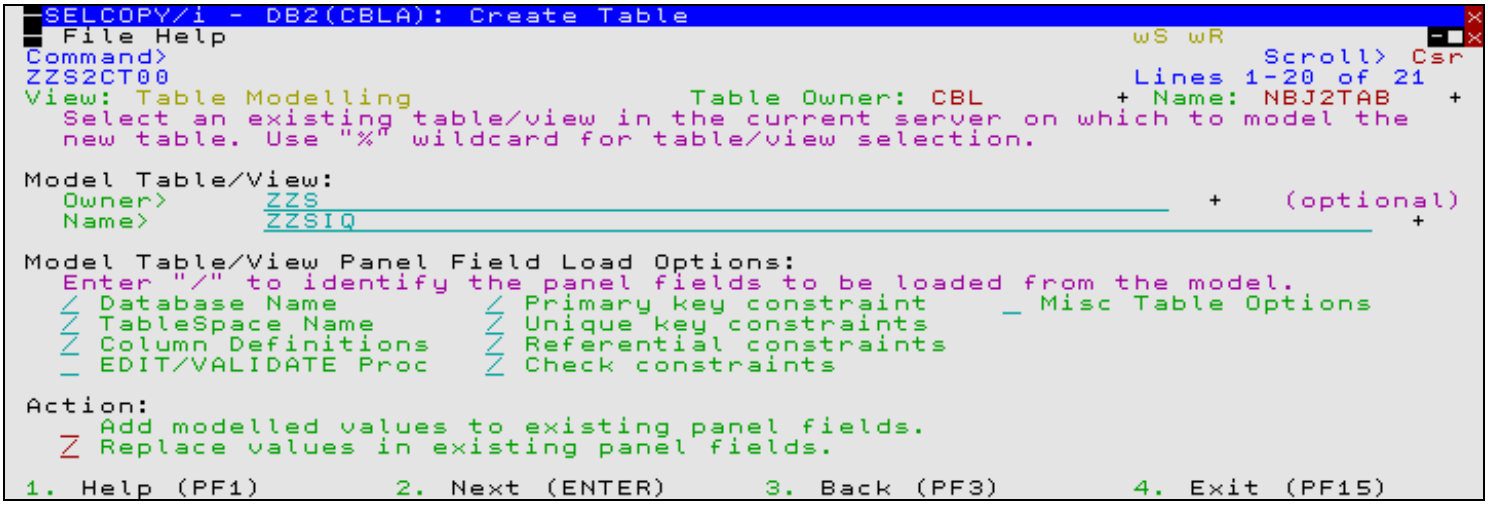


Figure 228. DB2: Create Table - Table Modelling.

Model on Tables/Views - Panel Fields

`Table Owner:`
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

`Name:`
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

`Model Table/View:`
Fields that together identify the model DB2 table or DB2 view from which attributes of the new table will be modelled.

These field values may include wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the parent table owner/name specification. If this is the case, a Select Table panel is displayed containing a list of eligible DB2 table or view names which satisfy the table owner/name filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

`Location>`
The DB2 server location of the model table or view.

`Owner>`
The model DB2 table or view schema (owner).

`Name>`
The model DB2 table or view name.

`Model Table/View Panel Field Load Options:`
Enter "/" (slash) or any non-blank character against each of the required options in order to identify the table attributes to be modelled.

`Database Name`
Insert the name of the database in which the model DB2 table is defined into the DataBase input field of the Table Name & Location panel view.

If the Add action is in effect, modelling of the database name occurs only if the DataBase input field is blank.

`TableSpace Name`
Insert name of the tablespace in which the model DB2 table is defined into the TableSpace input field of the Table Name & Location panel view.

If the Add action is in effect, modelling of the tablespace name occurs only if the TableSpace input field is blank.

`Column definitions`
Insert model DB2 table column definitions into the Column Definitions sub-panel.

All column definition fields, except for IDENTITY column sequence definition fields, are initialised with values inherited from the model column definition. This includes the column name, type, length/precision, scale, default value specification and default value, null value specification, field procedure name and parameters, distinct type schema and name, character sub-type and implicitly hidden option.

Note that the Column definitions option is implied if any of the options to model constraints are selected. (Primary key, Unique key, Referential or Check.)

`EDIT/VALIDATE Proc`
Insert the names of the edit and validate procedures that are assigned to the model DB2 table into the EDITPROC input field of the Table Definition panel view and VALIDPROC input field of the Table Options (1/2)

panel view respectively.

If the Add action is in effect, modelling of procedure names occurs only if the appropriate input field is blank.

Primary key constraint
Insert the model DB2 table primary key definition constraint name and associated column names into the Primary Key Definition sub-panel.

If this option is selected, then column definition modelling will also be performed whether or not the Columns definition option has been set.

If the Add action is in effect, modelling of the primary key occurs only if no primary key columns have already been specified. If so, then the model DB2 table primary key definition will be treated as a unique key definition and included as part of the Unique key constraint modelling if the Unique key constraints option has also been selected.

Unique key constraints
Insert model DB2 table unique key constraint definitions into the Unique Key Constraint Definitions and Unique Constraint Key Columns sub-panels

If this option is selected, then column definition modelling will also be performed whether or not the Columns definition option has been set.

Referential constraints
Insert model DB2 table referential constraint definitions into the Referential Constraint Definitions, Referential Constraint - Parent Key Columns and Referential Constraint - Foreign Key Columns sub-panels.

If this option is selected, then column definition modelling will also be performed whether or not the Columns definition option has been set.

Check constraints
Insert model DB2 table referential constraint definitions into the Check Constraint Definitions sub-panel.

If this option is selected, then column definition modelling will also be performed whether or not the Columns definition option has been set.

Misc Table Options
Set table option flag fields found in panel views Table Options (1/2) and Table Options (2/2), to match table options that have been set for the model DB2 table. These table options and their corresponding option fields are as follow:

- Use of DROP TABLE. (DROP Table)
- Append of data rows on INSERT or LOAD. (Append Data Rows)
- Level of table access auditing. (Audit Table Access)
- Method of table access. (DB2 SQL Table Access)
- Additional logging of changes to table columns or data. (Data Capture Logging)

Modelling of these table options will occur for both Add and Replace actions.

Action:
Enter "/" (slash) or any non-blank character to select whether modelled table attributes are to either be added to existing table attribute values wherever possible, or to replace existing values.

Beware that, when replacing existing table attribute values, those values may not be subsequently restored. Similarly, adding to existing values may result in an invalid table definition. e.g. Table column definitions added to existing column definitions may result in duplicate column names or an invalid column type specification. This would result in an error from SELCOPY/i when column vetting is performed.

Materialized Query Definition

The Materialized Query Definition panel view is the next view in the sequence, displayed following the Table Name & Location panel view, when a materialized query table is to be defined.

This view allows the user to specify a fullselect SQL query expression to generate a DB2 result table from which the new DB2 materialized query table will be modelled and loaded.

If column names in the DB2 result table are to be named differently in the new table definition and the AS clause is not used to name columns within the fullselect clause, then execute primary command COLMAP (assigned to <F5>) to open the Result Table Column Name Remap sub-panel.

Corresponds to SQL CREATE TABLE syntax AS (*fullselect*) DATA INITIALLY DEFERRED REFRESH DEFERRED. See "DB2 for z/OS SQL Reference" for further information on the materialized query table attributes that cannot be modelled from a generated result table.

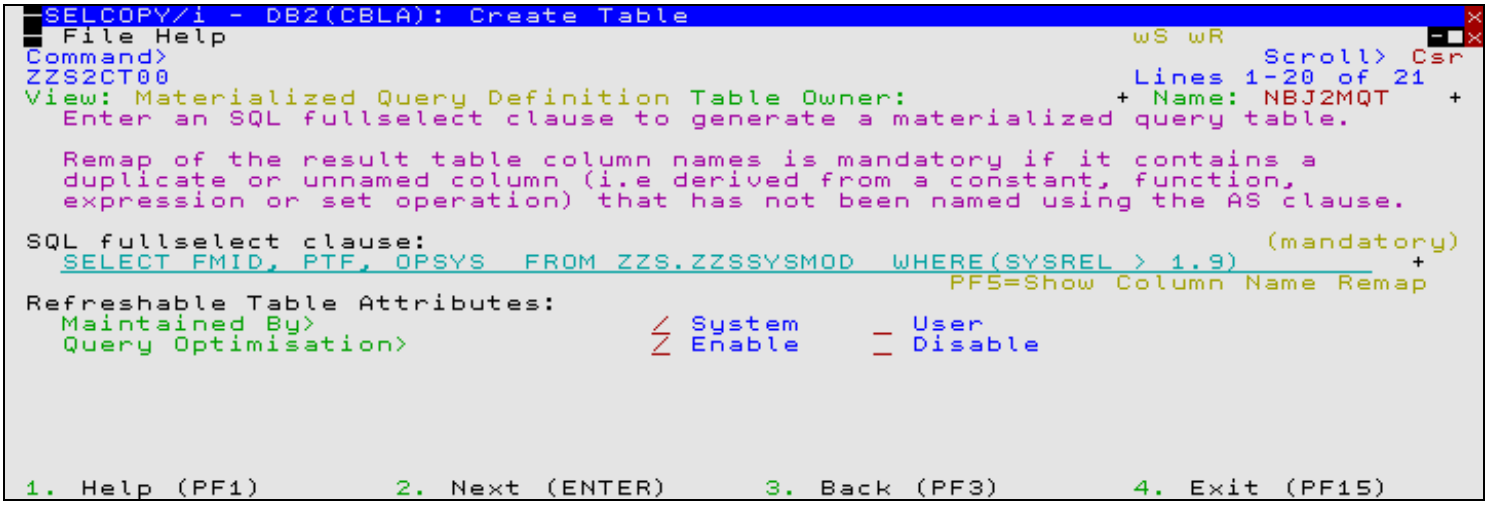


Figure 229. DB2: Create Table - Materialized Query Definition.

Materialized Query Definition - Panel Fields

`Table Owner:`
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

`Name:`
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

`SQL fullselect clause:`
Enter the SQL fullselect query expression. If the size of the input field is not sufficient to enter the complete fullselect clause, then execute primary command EXPAND (default <F14>) to display an editable view of the field.

Note that no validation of the SQL fullselect clause is performed by SELCOPY/i.

`Refreshable Table Attributes:`
Attributes the govern maintenance of materialized query table data and its use for SQL query optimisation.

`Maintained By>`
Enter "/" (slash) or any non-blank character against the mutually exclusive options to control whether the table data can only be updated by REFRESH TABLE SQL statement (System) or if it can also be updated by a user data change statement or the LOAD utility (User).

Corresponds to SQL CREATE TABLE syntax MAINTAINED BY SYSTEM or USER.

`Query Optimisation>`
Enter "/" (slash) or any non-blank character against the mutually exclusive options (Enable or Disable) to control whether or not the table may be used for SQL query optimisation.

Corresponds to SQL CREATE TABLE syntax ENABLE or DISABLE QUERY OPTIMIZATION.

Implicit TableSpace Options

The Implicit TableSpace Options panel view is displayed following table modelling panel views and prior to display of the Columns & Constraints panel view, only if the new DB2 table is to be created in an implicitly defined tablespace.

This view allows the user to select the type of tablespace to be created defined and also its default encoding of string data.

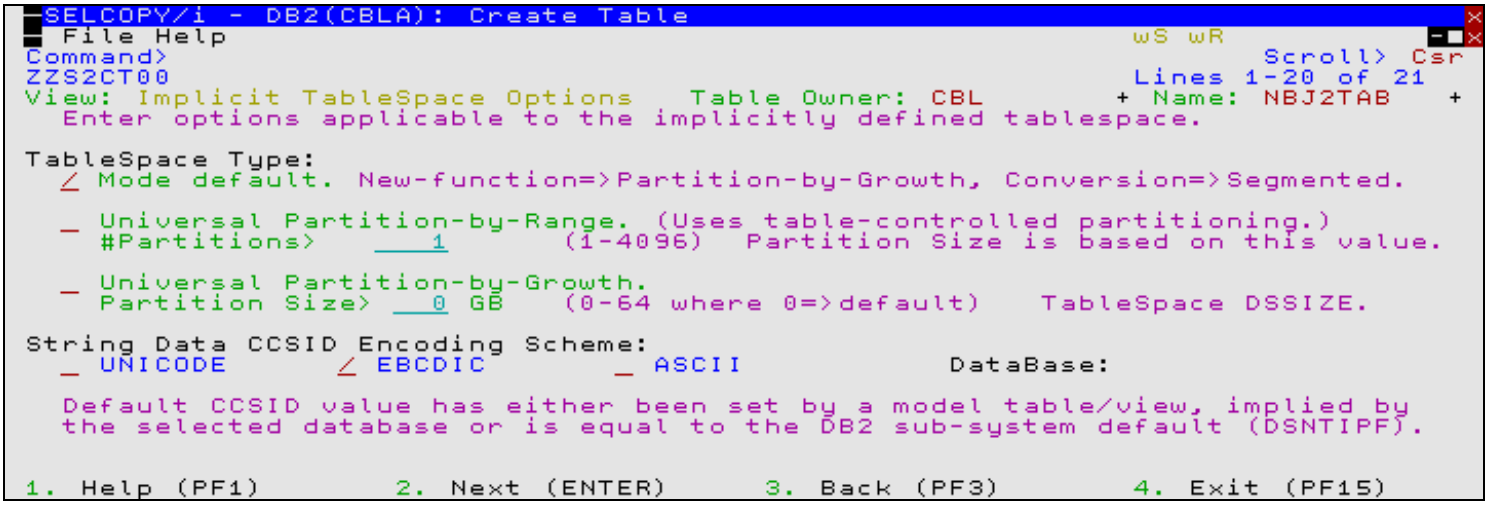


Figure 230. DB2: Create Table - Implicit TableSpace Options.

Implicit TableSpace Options - Panel Fields

`Table Owner:`
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

`Name:`
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

`TableSpace Type:`
Enter "/" (slash) or any non-blank character against the preferred tablespace type. See "DB2 Administration Guide" and "DB2 Installation and Migration Guide" for details on tablespace types and DB2 migration mode.

If Universal Partition-by-Range is selected, then the Range Partitioning Key Columns and Tablespace Range Partitions sub-panels will automatically be displayed later in the panel view sequence. These prompt the user to specify DB2 table controlled partitioning values.

`#Partitions>`
Applicable only to a Universal Partition-by-Range tablespace, this input field specifies the number of partitions (NUMPARTS) that will be assigned to the new table space. Partition limit values will need to be defined for each partition.

The number of partitions determines the maximum size (default DSSIZE) for each partition. See "CREATE TABLESPACE" in "DB2 SQL Reference" for details.

`Partition Size>`
Applicable only to a Universal Partition-by-Growth tablespace, this input field specifies the size of each partition (DSSIZE) in GB that will be assigned to the new table space as it grows.

A zero value will result in use of the default DSSIZE for the partition-by-growth tablespace.

`String Data CCSID Encoding Scheme:`
Enter "/" (slash) or any non-blank character against the preferred encoding scheme (UNICODE, EBCDIC or ASCII) to be assigned to the new DB2 table and, therefore, the implicitly defined tablespace.

The encoding scheme that is already selected for the user by default, is determined in the following order of precedence:

1. The encoding scheme inherited by DB2 table or view modelling.
2. The default encoding scheme of the database in which the table will be created.
3. The DB2 server (sub-system) default encoding scheme (set by install panel DSNTIPF.)

Note that changing the encoding scheme may invalidate column definitions that have been already been added to the Column Definitions sub-panel.

`DataBase:`
A non-enterable field displaying the name of the database, as specified in the **Table Name & Location** panel view, in which the implicit tablespace will be created.

Table Options (1/2)

The Table Options (1/2) panel view is displayed when specification of column definitions, constraint definitions, modelling options and implict tablespace options has been completed.

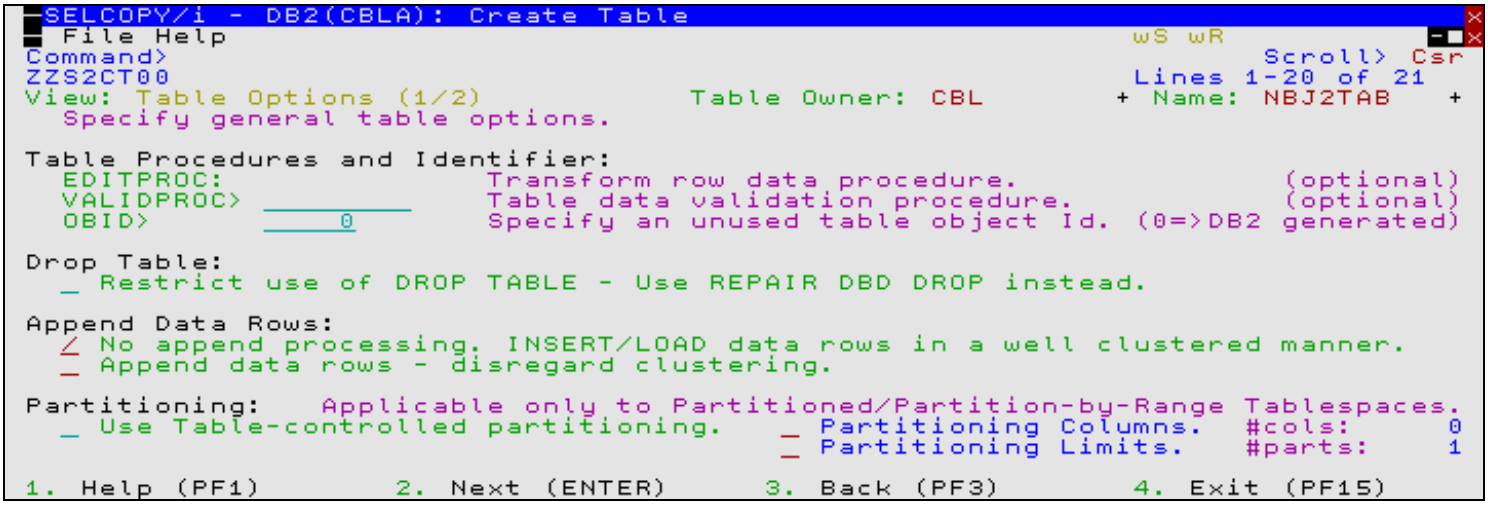


Figure 231. DB2: Create Table - Table Options (1/2).

Table Options (1/2) - Panel Fields

`Table Owner:`
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

`Name:`
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

`EDITPROC:`
A non-enterable field displaying the name of the edit procedure to be assigned to the new table as supplied in the **Table Definition** panel view.

`VALIDPROC>`
Optionally specifies the name of a validation procedure exit routine to be executed before a load, insert, update, or delete operation on any row of the table takes place, and so potentially inhibit the operation.

A procedure name has a maximum length of 8 characters.

`OBID>`
Optionally specifies the integer value to be used as the DB2 table object's internal descriptor. This valuue must not identify an existing or previously used OBID of the database.

By default DB2 will generate a unique OBID for the table.

`DROP Table:`
Enter "/" (slash) or any non-blank character to indicate that use of SQL statement DROP TABLE is to be prohibited for the new DB2 table. (RAPAIR DBD DROP must be used instead.)

Corresponds to SQL CREATE TABLE syntax WITH RESTRICT ON DROP.

`Append Data Rows:`
Enter "/" (slash) or any non-blank character to select the method by which inserted or loaded data rows are placed in the table by DB2.

Corresponds to SQL CREATE TABLE syntax APPEND NO or YES.

`Partitioning:`
Applicable only if the new DB2 table is to be defined in a partitioned or Universal Partitioned-by-Range tablespace, these options define table controlled tablespace partitioning values. (Compare with index controlled tablespace partitioning.)

If the destination tablespace is not range partitioned, then these options are ignored.

`Use Table-controlled partitioning`
Enter "/" (slash) or any non-blank character in this field to indicate that table controlled partitioning is to be configured for the implicitly defined partitioned-by-range tablespace or the named, explicitly defined partitioned tablespace in which the new DB2 table will be created.

If **Universal Partition-by-Range** is selected as the implicitly defined tablespace type in the Implicit TableSpace Options panel view, then, regardless of whether this option has been selected, table-controlled partitioning will be configured.

If this option is selected, then the Range Partitioning Key Columns and Tablespace Range Partitions sub-panels may be displayed before progressing to the next panel view.

Partitioning Columns/Partitioning Limits
Enter "/" (slash) or any non-blank character against the partitioning attributes sub-panel to be displayed next. The two sub-panel selections are mutually exclusive.

Tablespace range partitioning requires that at least one partitioning column must be identified and, for each tablespace partition, limit values must be specified for each partitioning column. Therefore, regardless of the current sub-panel selection, the next sub-panel displayed in the panel sequence will be as follows:

1. If no partitioning columns have been defined and selected, the Range Partitioning Key Columns sub-panel is displayed.
2. If partitioning column limit values have not been defined for **all** tablespace partitions, the Tablespace Range Partitions sub-panel is displayed.

Having selected, displayed and then closed a partitioning attributes sub-panel, the focus returns to this panel view and, by default, the next sub-panel is automatically selected for display. Also, fields displaying the number of partitioning columns is updated accordingly. If the last sub-panel selected was **Partitioning Limits**, then no further sub-panels are selected for display.

If no sub-panel is selected and the tablespace range partitioning conditions have been satisfied, then focus will progress to the next panel view in the sequence (Table Options (2/2).)

- **Partitioning Columns.**
  Display the Range Partitioning Key Columns sub-panel next.
- **Partitioning Limits.**
  Display the Tablespace Range Partitions sub-panel next.

#Cols:
A non-enterable field displaying the total number of partitioning columns that have been selected. This value is updated on return from the **Range Partitioning Key Columns** sub-panel.

#parts:
A non-enterable field displaying the number of tablespace partitions. This value is the NUMPARTS value assigned to the explicitly defined tablespace specified in the **Table Name & Location** panel view.

If the tablespace is implicitly defined, then this value is the **#Partitions** value specified in the Implicit TableSpace Options panel view.

Range Partitioning Key Columns

The Partitioning Key Columns sub-panel is displayed on selecting **Partitioning Columns** from the Table Options (1/2) panel view.

This sub-panel contains an embedded table of DB2 table primary key column definitions. Standard table edit primary and line commands must be used to INSERT, DELETE, COPY, MOVE or EXCLUDE table rows as appropriate, to scroll the table display UP, DOWN, LEFT and RIGHT and also to ZOOM the display of an individual table row.

In addition to the standard primary table commands, this sub-panel also supports the following primary commands:

- **SELECTAll** (assigned to <F6> by default). This toggles between selection and deselection of all entries in the table.
- **RESTORE** (assigned to <F5> by default). This restores the table entries so that all rows that satisfy the filter criteria are redisplayed. i.e. reset the table view.

Each table row identifies a column definition entered in the Column Definitions sub-panel, which also satisfies partitioning column criteria and is, therefore, eligible for selection as a partitioning key column. Apart from Sel (entry selection) and Sort columns, all other columns in this table are non-enterable and are included for information only.

Select and/or exclude the DB2 column definitions then, if necessary, rearrange them using line command "M" or "MM" so that they occur in the order required for the primary key constraint definition.

Corresponds to SQL CREATE TABLE syntax PARTITION BY RANGE *partition-expression*.

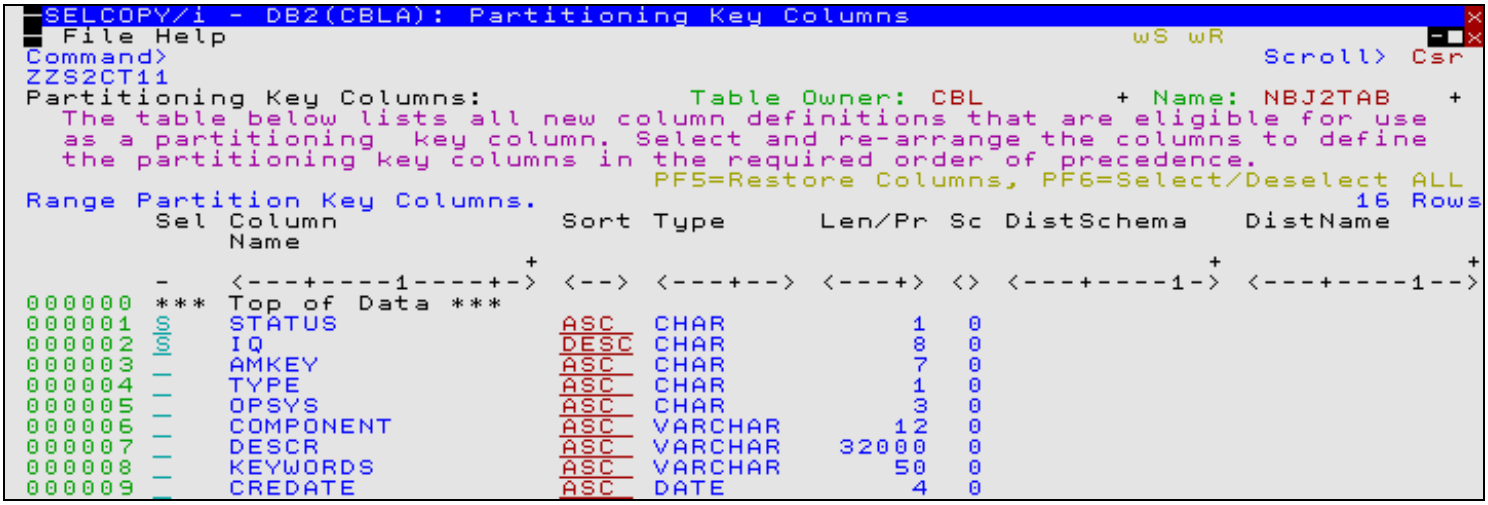


Figure 232. DB2: Create Table - Partitioning Key Columns.

Range Partitioning Key Columns - Panel Fields

`Table Owner:`
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

`Name:`
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

`Sel`
Enter "/" (slash) or any non-blank character in this field to include (select) the column definition in the create table syntax. A blank in this field will exclude (deselect) the column definition.

Deselecting a column definition in table view has the same effect as executing the table primary command EXCLUDE (or line command "X") to exclude the table row.

`Column Name`
A non-enterable field displaying the name of the column.

`Null`
Indicates whether null values in the column fields are treated as being low (negative infinity) or high (positive infinity) for the prposes of comparison.

Enter a blank or invalid value in this field to display a complete scrollable list of selectable entries.

`LOW`
Null values are low.

`HI`
Null values are high. Corresponds to SQL CREATE TABLE syntax NULLS LAST.

`Sort`
Indicates whether column entries are put in ascending or descending order. Whether sorting column entries in ascending or descending order, null values will be treated as having a value of poitive infinity for the purposes of comparison.

Enter a blank or invalid value in this field to display a complete scrollable list of selectable entries.

`ASC`
Puts the entries in ascending order by this key column.

`DESC`
Puts the entries in descending order by this key column.

Corresponds to SQL CREATE TABLE syntax NULLS LAST ASC or DESC.

`Type`
A non-enterable field displaying the type of the column. See Column Type for valid column types.

`Len/Pr`
A non-enterable field displaying the length or precision of the column data.

`Scale`
A non-enterable field displaying the scale of the column data.

`DistSchema`
A non-enterable field displaying the schema of the distinct type assigned to the column. If no distinct type is used, this field will be blank.

DistName
A non-enterable field displaying the name of the distinct type assigned to the column. If no distinct type is used, this field will be blank.

Tablespace Range Partitions

The Tablespace Range Partitions sub-panel is displayed on selecting **Partitioning Limits** from the Table Options (1/2) panel view.

This sub-panel contains an embedded table of DB2 table space partitions. The table is for update only so that rows may not be manipulated using the standard table editing techniques.

Each table row identifies a partition of the tablespace for which partition column limits must be entered.

Validation is performed for all partition definitions on exiting the Tablespace Range Partitions sub-panel. If limit values have not been specified for all partitioning columns of a tablespace partition entry, then the Range Partitioning Key Values sub-panel will be opened for that tablespace partition. to bypass validation, use CANCEL to exit the panel.

Corresponds to SQL CREATE TABLE syntax PARTITION *integer*.

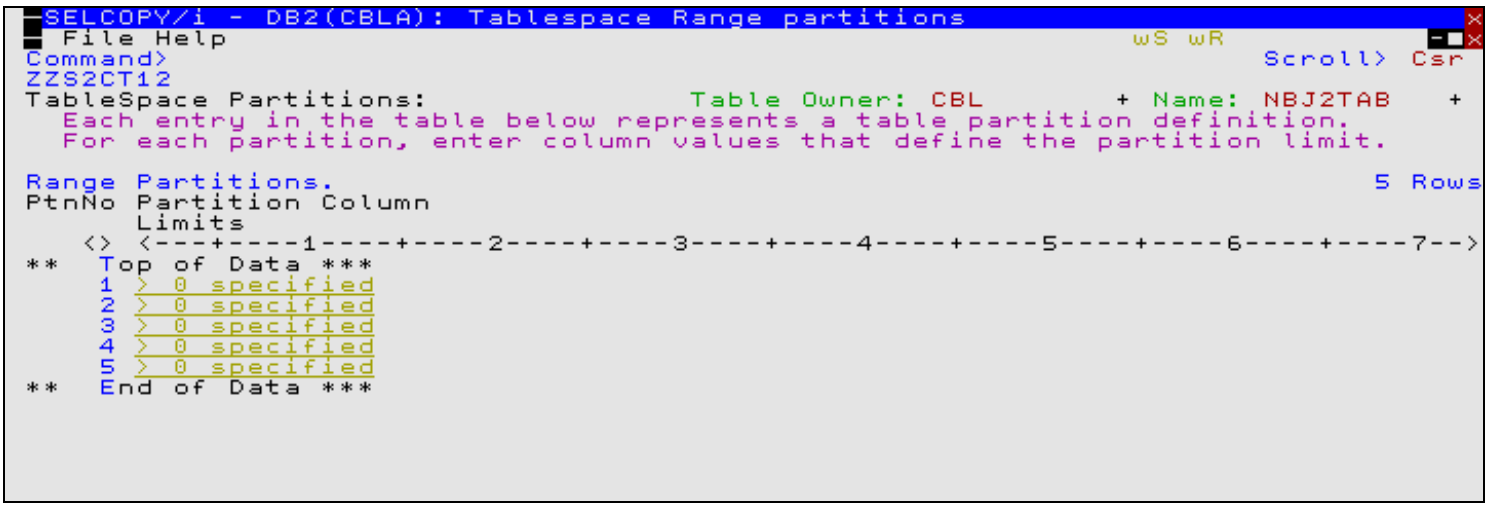


Figure 233. DB2: Create Table - Tablespace Range Partitions.

Tablespace Range Partitions - Panel Fields

Table Owner:
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

Name:
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

PtnNo
A non-enterable field displaying the table space partition number.

Partition Column Limits
Displays the number of partitioning columns for which a limit value has been specified. To enter limit values, position the cursor on the Partition Column Limits entry of the required tablespace partition and press <Enter> or, if configured, double-click the left mouse button to dispaly the Range Partitioning Key Values sub-panel.

Range Partitioning Key Values

The Partitioning Key Column Values sub-panel is displayed on selecting **Partition Column Limits** from the Tablespace Range Partitions sub-panel.

This sub-panel contains an embedded table of DB2 table space partitioning key columns. The table is for update only so that rows may not be manipulated using the standard table editing techniques.

Each table row identifies a partitioning column definition which was selected in the Range Partitioning Key Columns sub-panel. Apart from the Limit Value column, all columns in this table are non-enterable and are included for information only.

Following update of a table row (limit value) or on returning to the table view from a zoomed view, validation occurs for the limit value entered by the user. Likewise, validation is performed for all column limit values on exiting the Partitioning Key Column Values sub-panel. If an error is detected, then the table row containing the limit value in error is automatically zoomed. This allows the user to correct the error before continuing.

If the width of the Limit Value field in the table view is not sufficient to enter the required limit value, then the table row should be zoomed and, if necessary, the appropriate field expanded to accomodate the input value. To display a table row in single row (zoomed) format, execute primary command ZOOM (assigned to <F17> by default) with the cursor positioned on the required table row entry. EXPAND (assigned to <F14> by default) may then be used to expand an input field and so enter a value which is longer than the visible input field area.

Corresponds to SQL CREATE TABLE syntax ENDING AT (*constant*, MAXVALUE, or MINVALUE, ...).

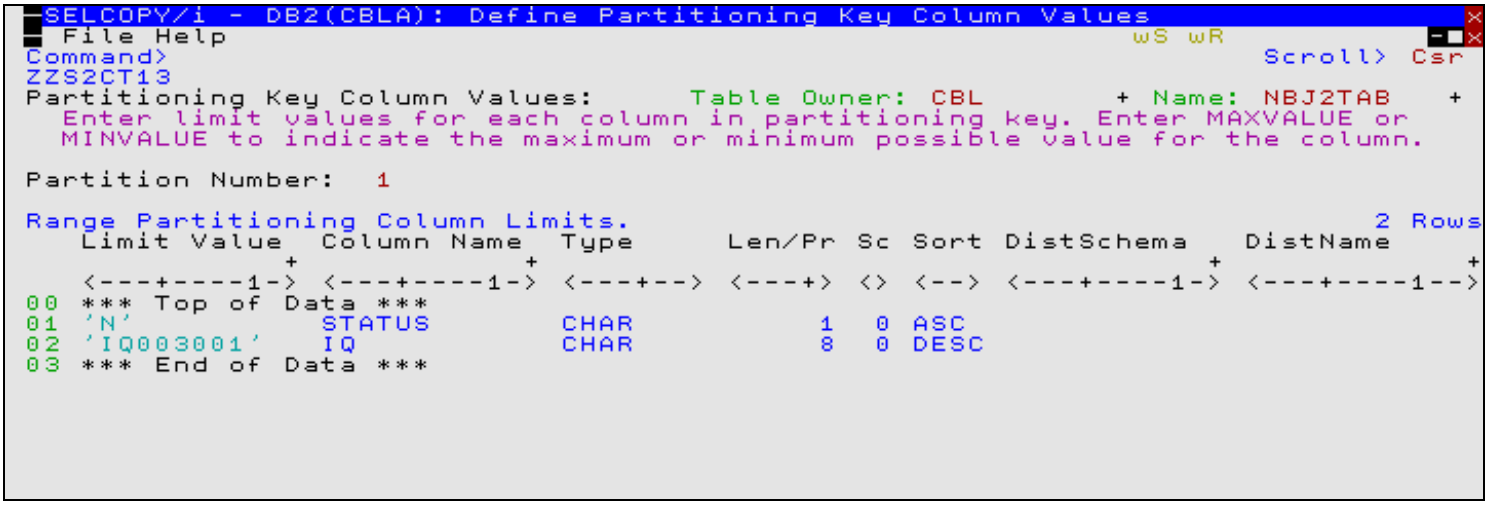


Figure 234. DB2: Create Table - Partitioning Key Column Values.

Range Partitioning Key Values - Panel Fields

Field names that follow are as appear in the zoomed view of a table row. Names in parentheses correspond to the equivalent column name in table view.

Table Owner:
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

Name:
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

Partition Number:
A non-enterable field displaying the current table space partition number.

Limit Value> (Limit Value)
Specifies the partition limit value for the column within the partitioning key.

If the column is a character data type, then the limit value must be enclosed within SQL delimiter characters as defined for the DB2 server. If these character string delimiters have been omitted, then they will be added automatically by the panel, escaping any delimiter character that exists as data within the string.

Exceptions to this rule are values MAXVALUE and MINVALUE (which may be entered with a minimum abbreviation of MAX and MIN) which represent maximum and minimum possible values for the column regardless of the column's defined data type. To enter MAXVALUE or MINVALUE (or their abbreviations) as string data, they must explicitly be entered within string delimiter characters.

Column Name: (Column Name)
A non-enterable field displaying the name of the column.

DataType: (Type)
A non-enterable field displaying the type of the column. See Column Type for valid column types.

Precision/Length: (Len/Pr)
A non-enterable field displaying the length or precision of the column data.

Scale (Sc)
A non-enterable field displaying the scale of the column data.

Value Sort Order: (Sort)
A non-enterable field displaying ASC or DESC, indicating the order in which entries are sorted by this column (ascending or descending).

DistType Schema: (DistSchema)
A non-enterable field displaying the schema of the distinct type assigned to the column. If no distinct type is used, this field will be blank.

DistType Name: (DistName)
A non-enterable field displaying the name of the distinct type assigned to the column. If no distinct type is used, this field will be blank.

Table Options (2/2)

The Table Options (2/2) panel view is the next view in the sequence, displayed following the Table Options (1/2) panel view.

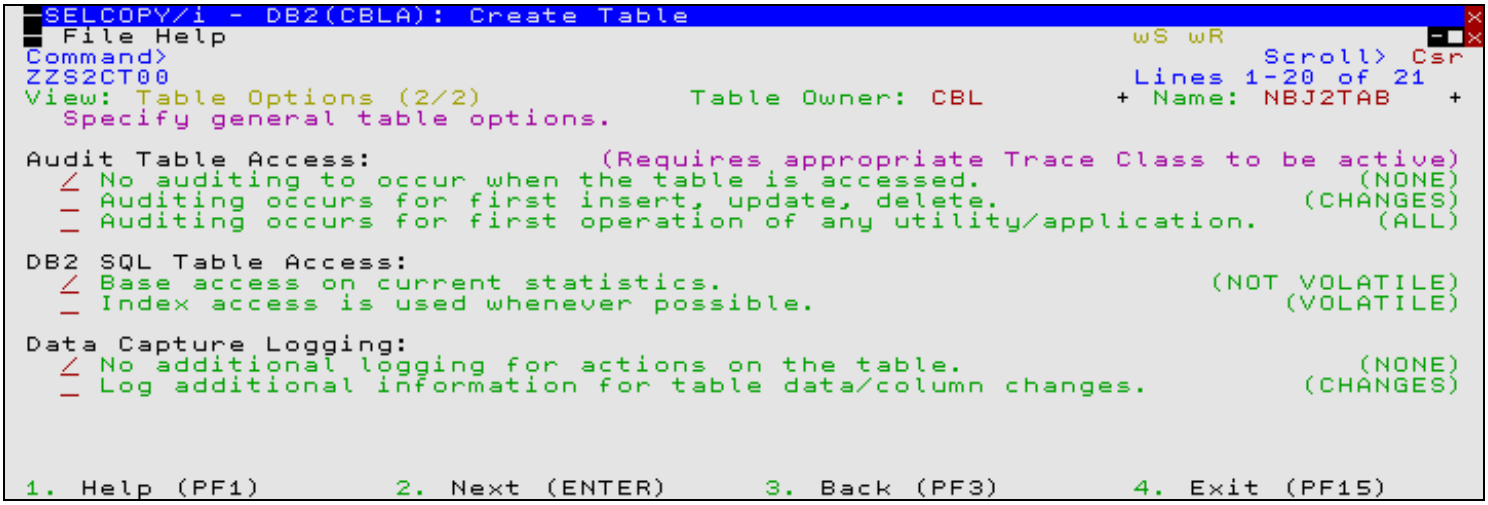


Figure 235. DB2: Create Table - Table Options (2/2).

Table Options (2/2) - Panel Fields

Table Owner:
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

Name:
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

Audit Table Access:
Enter "/" (slash) or any non-blank character to select the the types of access to this table that causes auditing to be performed.

Corresponds to SQL CREATE TABLE syntax AUDIT NONE, CHANGES or ALL.

DB2 SQL Table Access:
Enter "/" (slash) or any non-blank character to select how DB2 will choose access to this table.

Corresponds to SQL CREATE TABLE syntax VOLATILE/NOT VOLATILE CARDINALITY.

Data Capture Logging:
Enter "/" (slash) or any non-blank character to select whether the logging of the certain actions on this table will be enhanced with additional information.

Corresponds to SQL CREATE TABLE syntax DATA CAPTURE NONE or CHANGES.

Generate SQL

The Generate SQL panel view is the next view in the sequence, displayed following the Table Options (2/2) panel view.

This view allows the user to select how the generated SQL CREATE TABLE statement is to be implemented.

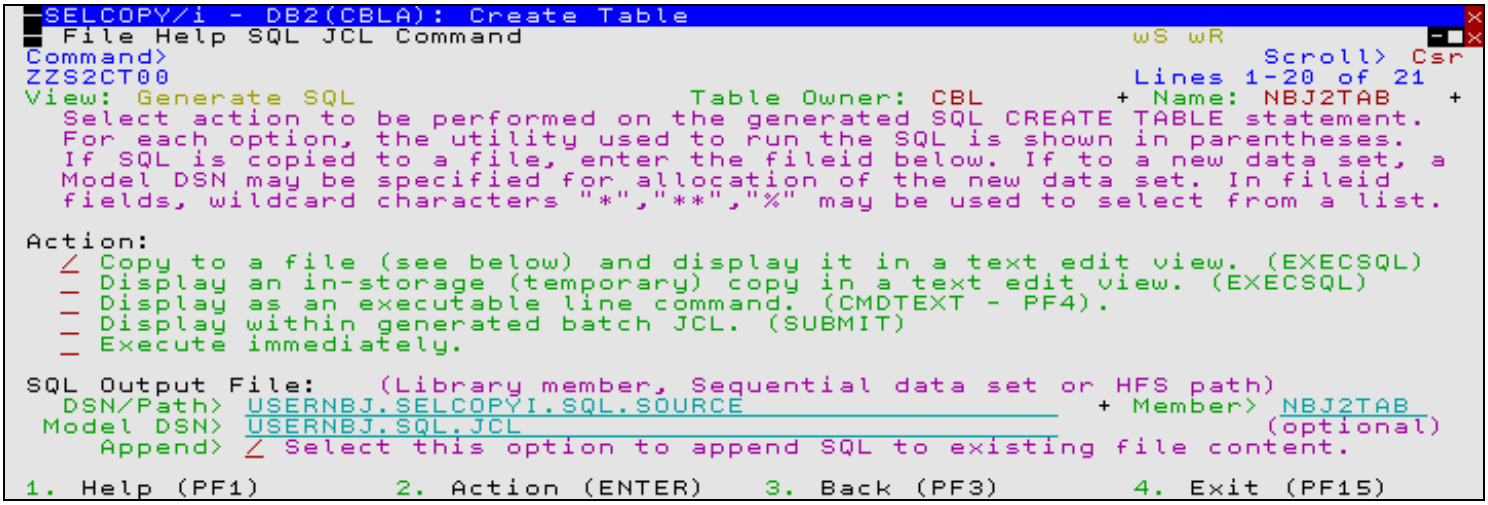


Figure 236. DB2: Create Table - Generate SQL.

Generate SQL - Panel Fields

`Table Owner:`
A non-enterable field displaying the new table owner id as supplied in the **Table Name & Location** panel view.

`Name:`
A non-enterable field displaying the new table name as supplied in the **Table Name & Location** panel view.

`Action:`
Enter "/" (slash) or any non-blank character to select the the action to be performed with the generated SQL statement on completion of the create table panel view sequence.

For all the actions below, except Execute immediately, the SQL will be displayed in an edit view. The appropriate command or facility may be issued by the user to subsequently execute the generated SQL statement.

`Copy to a file`
Copy the generated SQL statement to the output file specified by the SQL Output File fields below.

The SQL statement may subsequently be executed using the EXECSQL primary command.

`Display an in-storage copy`
Copy the generated SQL statement to an in-storage output file with a temporary DSN.

The SQL statement may subsequently be executed using the EXECSQL primary command.

`Display as an executable line command`
Copy the generated SQL statement to an in-storage output file in a format suitable for subsequent execution using the ACTION (default >F16>) facility.

`Display within generated batch JCL`
Copy the generated SQL statement to an in-storage output file and enclose it within JCL which executes the DSNTIAD load module.

The SQL statement may subsequently be executed using the SUBMIT primary command.

`Execute immediately`
Opens the Execute SQL Statements panel and immediately executes the generated SQL statement to create the new DB2 table. DB2 SQL messages are displayed in this panel.

`SQL Output File:`
Applicable only if the Copy to file action has been selected.

Input fields which together identify a single output file (sequential data set, HFS file path or PDS/PDSE library member) to which the generated SQL statement will be copied. This output file may be a new or existing data set, HFS file or library member.

`DSN/Path>`
Identifies the fully qualified data set name or an absolute or relative HFS file path.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

If a DSN is specified for a data set (of orgaisation PS or PO) that does not already exist, the Allocate NonVSAM data set dialog window will be opened to create the new output file.

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcard characters "*" (asterisk) or "%" (percent), or is blanked out.

Model Dsn>
Applicable only if the Copy to file action has been selected and **DSN/Path>** specifies a new data set or PDS/PDSE library name.

This field specifies the DSN of an existing sequential or PDS/PDSE library that will be used to model a new data set in the Allocate NonVSAM dialog window.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

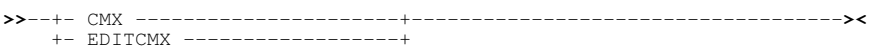
Append>
Applicable only if the Copy to file action has been selected.

Enter "/" (slash) or any non-blank character in this field to append the generated SQL statement to existing text in the output file. If not, existing text will be replaced by the SQl statement.

Primary Commands

The following primary commands are supported by selected views in the DB2 Create Table sequence of panel views. If issued from a panel view in which the command is not valid, the message "ZZSP102E Primary command not valid in the current context" is displayed.

CMX



Applicable only from the Generate SQL view, CMX generates the SQL statement and copies it to an in-storage output file in a format suitable for subsequent execution using the ACTION (default >F16>) facility.

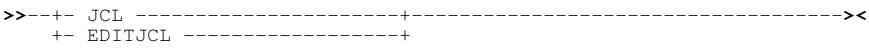
CMX is assigned to <F17> by default.

COLMAP

Applicable to both the Materialized Query Table Definition and Model on Result Table panel views, COLMAP opens the Result Table Column Name Remap sub-panel, used to define a column name AS new column name clause for each column in the DB2 result table.

COLMAP is assigned to <F5> by default.

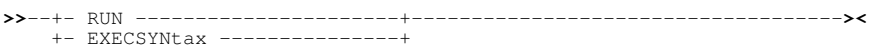
JCL



Applicable only from the Generate SQL view, JCL generates the SQL statement and copies it to an in-storage output file with JCL statements that execute the DSNTIAD facility. This job may be submitted to batch using the SELCOPY/i text editor SUBMIT primary command.

JCL is assigned to <F18> by default.

RUN



Applicable only from the Generate SQL view, RUN verifies input fields in all applicable panel views and then executes the generated SQL statement. This is the default action on pressing <Enter> from the last panel view in the sequence (i.e. the **Generate SQL** view.)

Create Index

The DB2 **Create Index** series of panel views (ZZS2CI00) generate an SQL CREATE INDEX statement which may be used to create a new DB2 index, XML index or Auxiliary index in the current DB2 subsystem.

The DB2 Create Index panel views and their sub-panels are interactive panel windows (window class WINWIPO0) and may be started by selecting Index option 8. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.8)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made within the panel views and also on the type of table space in which the index will be created. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL CREATE INDEX syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

Index & Table Name

Enter the name of the new index and the name of the table within the current DB2 subsystem on which the index will be defined. The current DB2 subsystem is displayed in the panel window title bar.

Following selection of a DB2 table, the type of table and table space in which the table is defined determines the index creation options that are presented in the panel views that follow. (e.g. Index partition options for Partitioned and Universal Partitioned-By-Range table spaces.)

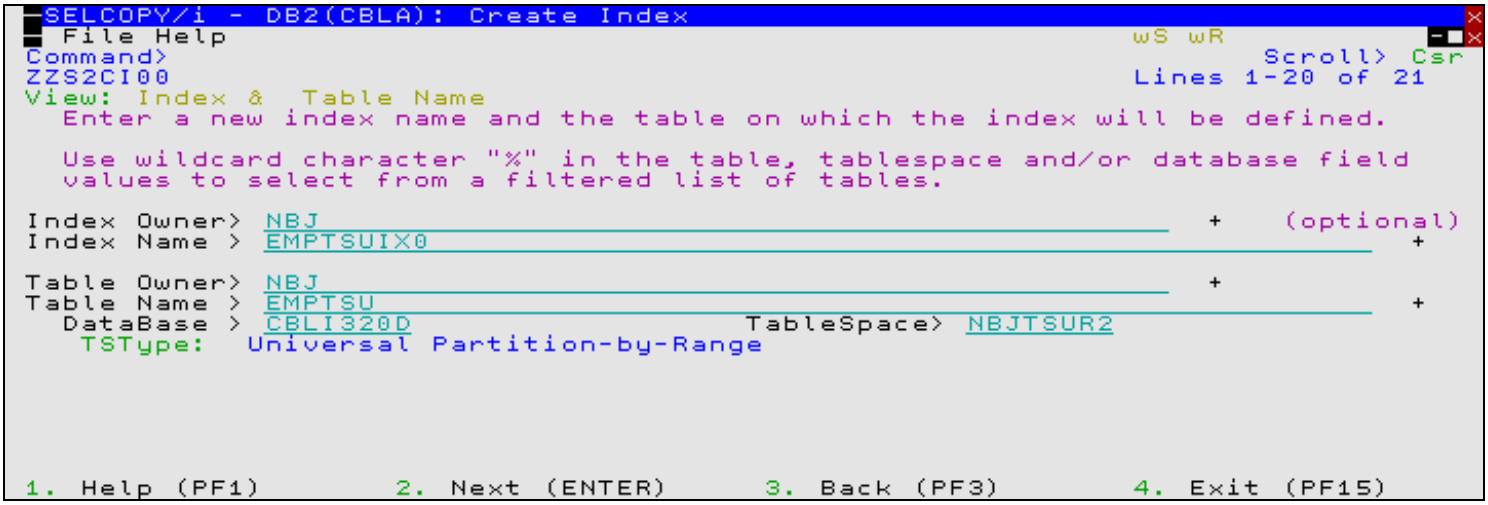


Figure 237. DB2: Create Index - Index & Table Name.

Menu Bar Items

The following menu bar items are common to all Create Index panel views and sub-panels.

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Index & Table Name - Panel Fields

Index Owner>
: Optionally specifies the owner id (schema) of the index to be created. If this field is left blank, DB2 will assign an owner id equal to the value of the user's current SQLID special register.

  A index owner id has a maximum length of 128 characters.

Index Name>
: Mandatory field which specifies the name of the index to be created.

An index name has a maximum length of 128 characters.

Table Owner>
Specifies the owner id (schema) of the table on which the index is created.
This field will be populated on specification of a DB2 table. See Table Name for details on table selection.

A table owner id has a maximum length of 128 characters.

Table Name>
Mandatory field which specifies the name of the table on which the index is created.

A blank entry for Table Owner, Table Name, DataBase or TableSpace is treated as wildcard character "%" (percent).

Wildcard character "%" (percent) or "*" (asterisk), representing zero or more characters, or wildcard character "_" (underscore), representing a single character, in one, but not all, of the Table Owner, Table Name, DataBase and TableSpace fields will open a Select Table panel. This panel will contain a list of tables that satisfy the filter derived from the 4 input fields.
Use prefix command "S", or press the <Enter> key on the required entry to select it and so populate the 4 input fields in the Index & Table Name panel view.

A table name has a maximum length of 128 characters.

DataBase>
Specifies the name of the database containing the table on which the index will be defined.
This field will be populated on specification of a DB2 table. See Table Name for details on table selection.

A database name has a maximum length of 8 characters.

TableSpace>
Specifies the name of the table space containing the table on which the index will be defined.
This field will be populated on specification of a DB2 table. See Table Name for details on table selection.

A table space name has a maximum length of 8 characters.

TSType:
A non-enterable field which identifies the type of table space in which the specified table is defined. Possible table space type descriptions are as follow:

- ◊ **Segmented.**
  The selected table space is a segmented (non-partitioned) table space.

- ◊ **Partitioned. (*n* parts)**
  The selected table space is a non-universal (i.e. non-segmented) partitioned table space defined as having the displayed number of partitions.

- ◊ **Universal Partition-by-Growth.**
  The selected table space is a universal (i.e. segmented) partitioned-by-growth table space.

- ◊ **Universal Partition-by-Range.**
  The selected table space is a universal (i.e. segmented) partitioned-by-range table space.

- ◊ **LOB (Large Object).**
  The selected table space is a LOB (large object) table space.

This field is updated following update of the TableSpace field before progressing to the next panel view.

Index Type

The Index Type panel view is the next view in the sequence, displayed following the Index & Table Name panel view for non-auxiliary indexes.

Depending on the selected table space, this view allows the user to identify the new index as being unique, clustering, partitioned or partitioning. It also identifies whether or not the index will be extended (containing key expressions.)

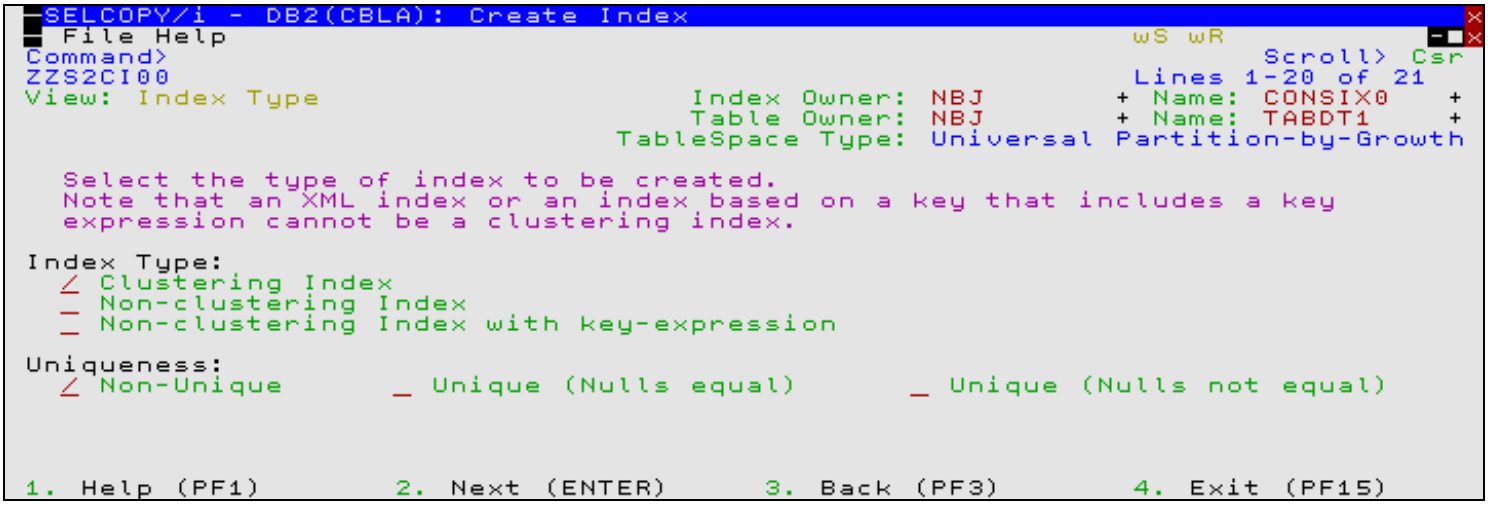


Figure 238. DB2: Create Index - Index Type. (Segmented or Universal Partition-by-Growth)

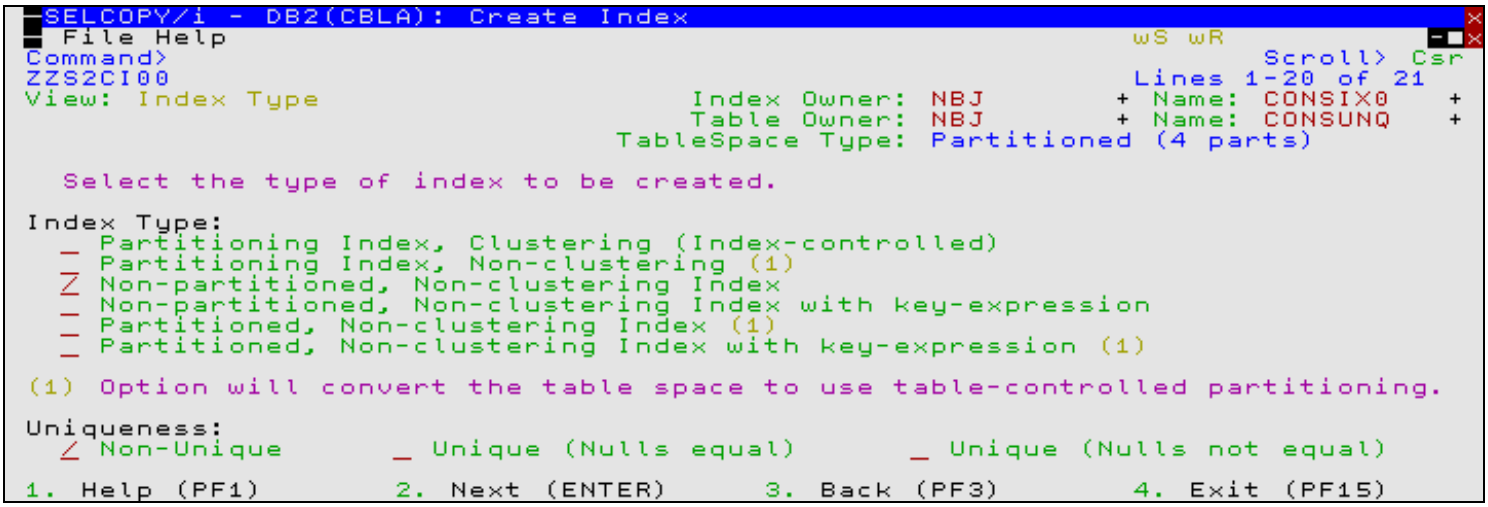


Figure 239. DB2: Create Index - Index Type. (Partitioned)

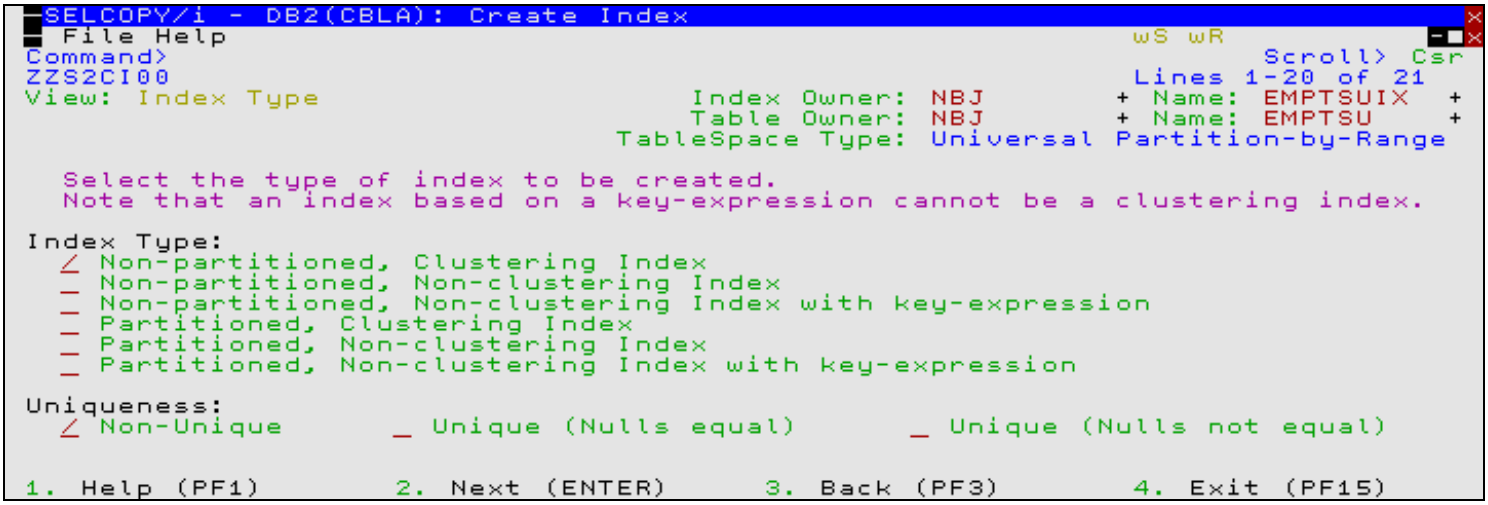


Figure 240. DB2: Create Index - Index Type. (Universal Partition-by-Range)

Index Type - Panel Fields

`Index Owner:`
A non-enterable field displaying the owner id of the new index as supplied in the **Index & Table Name** panel view.

Name:
A non-enterable field displaying the new index name as supplied in the **Index & Table Name** panel view.

Table Owner:
A non-enterable field displaying the table owner id as supplied in the **Index & Table Name** panel view.

Name:
A non-enterable field displaying the table name as supplied in the **Index & Table Name** panel view.

Tablespace Type:
A non-enterable field displaying the type of table space as identified in the **Index & Table Name** panel view.

Index Type:
Enter "/" (slash) or any non-blank character against the entry which describes the required attributes of the new index. Each of the options are mutually exclusive.

- ◊ **Clustering Index** or **Non-partitioned, Clustering Index**
  The index will be non-partitioned, clustering and have a key that consists solely of column name specifications.

- ◊ **Non-clustering Index** or **Non-partitioned, Non-clustering Index**
  The index will be non-partitioned, non-clustering and have a key that consists solely of column name specifications.

- ◊ **Non-clustering Index with key-expression** or **Non-partitioned, Non-clustering Index with key-expression**
  The index will be non-partitioned, non-clustering and have a key that may contain a scalar key expression (extended index). Note that an extended index cannot be a clustering index.

- ◊ **Partitioned, Clustering Index**
  Applicable only to indexes on tables in a **Universal Partition-by-Range** table space, the index will be partitioned, clustering and have a key that consists solely of column name specifications.

- ◊ **Partitioned, Non-clustering Index**
  The index will be partitioned, non-clustering and have a key that consists solely of column name specifications.

- ◊ **Partitioned, Non-clustering Index with key-expression**
  The index will be partitioned, non-clustering and have a key that may contain a scalar key expression (extended index).

- ◊ **Partitioning Index, Clustering (Index-controlled)**
  Applicable only to indexes on tables in a **Partitioned** table space, the index will be a clustering, table partitioning index. i.e. Table partition column limits will be index-controlled. This option is invalid if a partitioning index is already defined on the partitioned table space.

- ◊ **Partitioning Index, Non-clustering**
  Applicable only to indexes on tables in a **Partitioned** table space, the index will be a non-clustering, table partitioning index. Although table partition column limits are defined when the new index is created, because the index is non-clustering, DB2 will convert the table space partitioning to be table-controlled. This option is invalid if a partitioning index is already defined on the partitioned table space.

- ◊ **XML Index**
  Create an XML index. If the table does not contain an XML column, then error ZZSP124E will be returned when this option is selected.

Uniqueness:
Enter "/" (slash) or any non-blank character to select the level of uniqueness of data in table rows columns that comprise the index key. Each of the options are mutually exclusive.

- ◊ **Non-Unique**
  The table may contain two or more rows that have the same value in the index key columns.

- ◊ **Unique (Nulls equal)**
  The table may **not** contain two or more rows that have the same value in the index key columns. In determining whether key column values are unique, DB2 will treat null values in an index key column as being equal.

- ◊ **Unique (Nulls not equal)**
  The table may **not** contain two or more rows that have the same value in the index key columns. In determining whether key column values are unique, DB2 will treat null values in an index key column as being not equal.

Index Key Columns

The Index Key Columns sub-panel is automatically displayed on progressing from the Index Type panel view when index key columns have not yet been configured and an index type has been selected that is not based on a key-expression or an XML column.

The sub-panel may also be opened from any subsequent create index panel view, using the primary command, **KEYcolumns** (assigned to <F6> by default).

This sub-panel contains an embedded table of eligible column names that belong to the selected DB2 table. Standard table edit primary and line commands must be used to DELETE, MOVE or EXCLUDE table rows as appropriate, to scroll the table display

UP, DOWN, LEFT and RIGHT and also to ZOOM the display of an individual table row.

In addition to the standard primary table commands, this sub-panel also supports primary commands **SELECTAll** and **REFRESH**, to toggle selection and deselection of all columns and reset display of column definitions. (By default, SELECTALL is assigned to <F5>, SELECTALL DESELECT is assigned to <F6> and REFRESH is assigned to <F16>.)

Each selected table row identifies an index column, its sequence order within the new index key and the order (ASCending, DESCending or RANDom) in which the column's values will be indexed.

The sub-panel also identifies the padding status for data in all variable length character columns that comprise the index key.

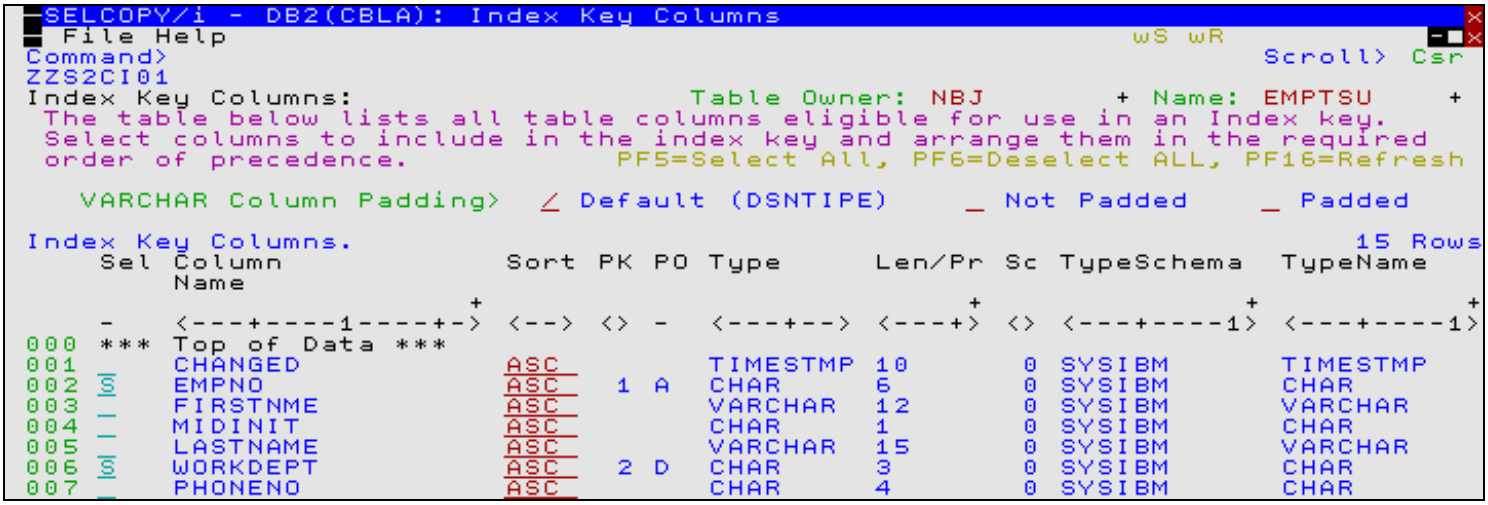


Figure 241. DB2: Create Index - Index Key Columns.

Index Key Columns - Panel Fields

Field names that follow are as appear in the zoomed view of a table row. Names in parentheses correspond to the equivalent column name in table view.

`Table Owner:`
A non-enterable field displaying the table owner id as supplied in the **Index & Table Name** panel view.

`Name:`
A non-enterable field displaying the table name as supplied in the **Index & Table Name** panel view.

`VARCHAR Column Padding>`
Enter "/" (slash) or any non-blank character against the padding status required for indexed data in variable length character or graphic data columns.

- ◊ **Default (DSNTIPE)**
  Uses the default padding for the DB2 server as defined in the DSNTIPE customisation panel.
- ◊ **Not Padded**
  Suppress padding of variable length character strings. Generated CREATE INDEX syntax will include NOT PADDED.
- ◊ **Padded**
  Perform padding of variable length character strings. Generated CREATE INDEX syntax will include PADDED.

`Sel> (Sel)`
Enter "/" (slash) or any non-blank character in this field to include (select) the column definition in the index key. A blank in this field will exclude (deselect) the column.

Deselecting a column in table view has the same effect as executing the table primary command EXCLUDE (or line command "X") to exclude the table row.

`Column Name: (Column Name)`
A non-enterable field displaying a column name belonging to the selected table.

`Sort> (Sort)`
Specifies the order in which values belonging to a selected column will be indexed. Index entries will be arranged in ASCending, DESCending or RANDom order by the column.

Enter a blank or invalid value in this field to display the list of valid, selectable entries.

`Partitioning Key Column Sequence: (PK)`
Applicable only to tables in table spaces for which table-controlled partitioning is used, this is a non-enterable field displaying the numeric position of the column within the partitioning key. If the column is not included a the partitioning key, this field will be blank.

If index key columns are selected that match or comprise a superset of the partitioning key columns and, furthermore, the selected index ordering (ascending/descending) for these columns match the partitioning key column ordering, then the new index will be a partitioning index. Otherwise, it will be a secondary index (DPSI or NPSI).

Partitioning Key Column Ordering: (PO)
Applicable only to tables in table spaces for which table-controlled partitioning is used, this is a non-enterable field displaying the ordering (Ascending or Descending) of the column within the partitioning key. If the column is not included a the partitioning key, this field will be blank.

DataType: (Type)
A non-enterable field displaying the column data type.

Precision/Length: (Len/Pr)
A non-enterable field displaying the length or precision assigned to the column data type.

Scale: (Sc)
A non-enterable field displaying the scale value assigned to the column data type.

DistType Schema: (TypeSchema)
A non-enterable field which displays the source type schema of the column data type. This is particularly relevant where the column is assigned a defined distinct type.

DistType Name: (TypeName)
A non-enterable field which displays the source type name of the column data type. This is particularly relevant where the column is assigned a defined distinct type.

Index Key Columns & Expressions

The Index Key Columns & Expressions sub-panel is automatically displayed on progressing from the Index Type panel view when an index key involving an expression has not yet been configured and an index type has been selected that is based on a key-expression.

The sub-panel may also be opened from any subsequent create index panel view, using the primary command, **KEYcolumns** (assigned to <F6> by default).

This sub-panel contains an embedded table of DB2 table column names and/or expressions based on table columns that define the index key. The order of the table rows determines the sequence order in which the column names and key-expressions occur within the index key.

Standard table edit primary and line commands must be used to INSERT, DELETE, COPY, MOVE or EXCLUDE table rows as appropriate, to scroll the table display UP, DOWN, LEFT and RIGHT and also to ZOOM the display of an individual table row.

In addition to the standard primary table commands, this sub-panel also supports primary command **SELECTAll** to toggle selection and deselection of all columns and expressions. (By default, SELECTALL is assigned to <F5> and SELECTALL DESELECT is assigned to <F6>.)

Each selected table row must specify either a DB2 table column name or a scalar expression referencing at least one DB2 table column. Which of these index key element types is specified is determined by the key element format field value (COLumn or EXPRession.)

Unlike the Index Key Columns sub-panel, the sort order in which index entries will be arranged by the column is not a customisable field since indexes involving key-expressions can only have values arranged in ascending order.

If the width of the Column/Expression field in the table view is not sufficient to enter the required input value, then the table row should be zoomed (default <F17>) and, if necessary, the field expanded (default <F14>) to accomodate the input value.

The sub-panel also identifies the padding status for data in all variable length character columns that comprise the index key.

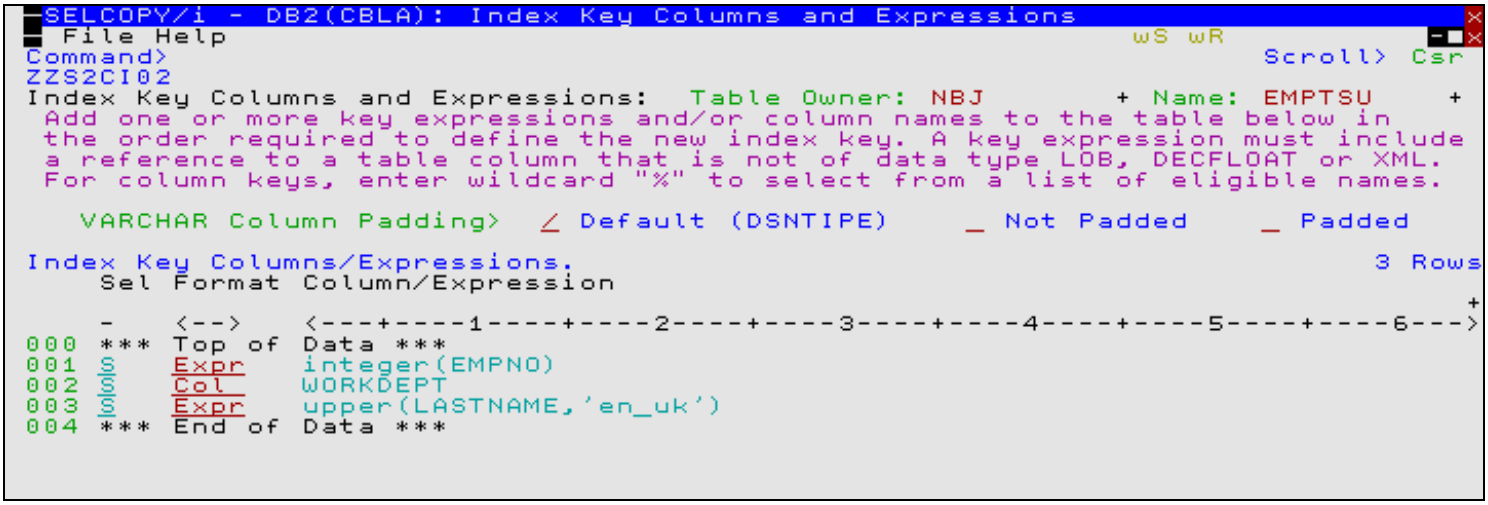


Figure 242. DB2: Create Index - Index Key Columns & Expressions.

Index Key Columns & Expressions - Panel Fields

Field names that follow are as appear in the zoomed view of a table row. Names in parentheses correspond to the equivalent column name in table view.

`Table Owner:`
A non-enterable field displaying the table owner id as supplied in the **Index & Table Name** panel view.

`Name:`
A non-enterable field displaying the table name as supplied in the **Index & Table Name** panel view.

`VARCHAR Column Padding>`
Enter "/" (slash) or any non-blank character against the padding status required for indexed data in variable length character or graphic data columns.

- ◊ **Default (DSNTIPE)**
  Uses the default padding for the DB2 server as defined in the DSNTIPE customisation panel.
- ◊ **Not Padded**
  Suppress padding of variable length character strings. Generated CREATE INDEX syntax will include NOT PADDED.
- ◊ **Padded**
  Perform padding of variable length character strings. Generated CREATE INDEX syntax will include PADDED.

`Sel> (Sel)`
Enter "/" (slash) or any non-blank character in this field to include (select) the column name/key-expression definition in the index key. A blank in this field will exclude (deselect) the definition.

Deselecting a column in table view has the same effect as executing the table primary command EXCLUDE (or line command "X") to exclude the table row.

`Key Format> (Format)`
Identifies the format (COLumn or EXPRession) of the index key element specified in the Index Column Name or Expression field.

If Column is selected, the column name entry must be the name of a valid column belonging to the selected DB2 table. Furthermore, if a blank or wildcard column name entry is specified, a column selection panel is displayed allowing the user to select a valid column name.

If Expression is selected, no validation is performed for the value entered in the Index Column Name or Expression field.

`Index Column Name or Expression> (Column/Expression)`
Depending on the selected index key element format, this field specifies the name of a valid column or a scalar expression based on a column belonging to the selected DB2 table.

For an entry of format COLumn, a blank or wildcard in this field will open the column selection panel displaying valid index columns from the selected DB2 table. Doing this before later changing the format to EXPRession, is a good starting point for constructing a key-expression on a particular column.

`Index Column Attributes:`
Applicable only where a column has been selected from the column selection panel, this block of non-enterable fields provides useful information on the column selected.

`DataType:`
A non-enterable field displaying the column data type.

Precision/Length:
A non-enterable field displaying the length or precision assigned to the column data type.

Scale:
A non-enterable field displaying the scale value assigned to the column data type.

Type Schema:
A non-enterable field which displays the source type schema of the column data type. This is particularly relevant where the column is assigned a defined distinct type.

Type Name:
A non-enterable field which displays the source type name of the column data type. This is particularly relevant where the column is assigned a defined distinct type.

PartKey Seq#
Applicable only to tables in table spaces for which table-controlled partitioning is used, this is a non-enterable field displaying the numeric position of the column within the partitioning key. If the column is not included a the partitioning key, this field will be blank.

If index key columns are selected that match or comprise a superset of the partitioning key columns and, furthermore, the selected index ordering (ascending/descending) for these columns match the partitioning key column ordering, then the new index will be a partitioning index. Otherwise, it will be a secondary index (DPSI or NPSI).

PartKey Order:
Applicable only to tables in table spaces for which table-controlled partitioning is used, this is a non-enterable field displaying the ordering (Ascending or Descending) of the column within the partitioning key. If the column is not included a the partitioning key, this field will be blank.

Index Options

The Index Options panel view is the next view in the sequence following the Index Type or XML Index panel views.

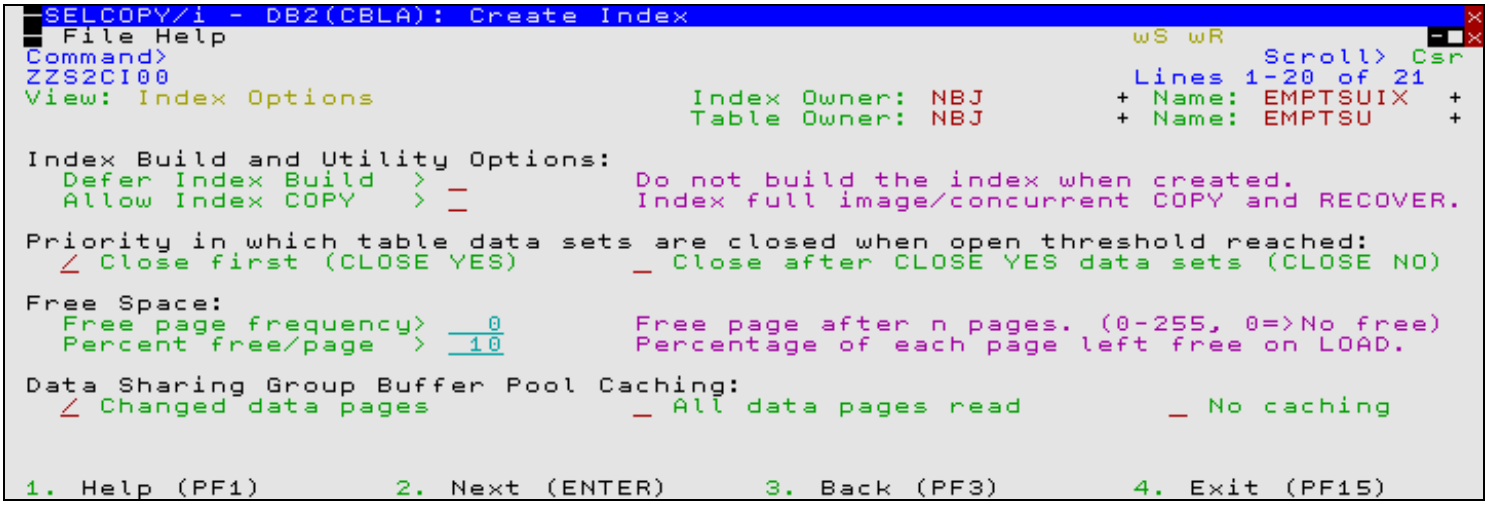


Figure 243. DB2: Create Index - Index Options.

Index Options - Panel Fields

Index Owner:
A non-enterable field displaying the owner id of the new index as supplied in the **Index & Table Name** panel view.

Name:
A non-enterable field displaying the new index name as supplied in the **Index & Table Name** panel view.

Table Owner:
A non-enterable field displaying the table owner id as supplied in the **Index & Table Name** panel view.

Name:
A non-enterable field displaying the table name as supplied in the **Index & Table Name** panel view.

Defer Index Build>
Select this option to suppress build of the index when CREATE INDEX is executed. Build of the index is deferred until REBUILD INDEX is executed.

This option determines parameter DEFER NO/YES in the generated CREATE INDEX statement and is ignored if the index is for an Auxiliary table.

Allow Index COPY>
Select this option to allow the COPY utility to perform concurrent or full-image copies of this index and also to allow use of the RECOVER utility on this index.

This option determines parameter COPY NO/YES in the generated CREATE INDEX statement.

Data Set Close Priority>
Enter "/" (slash) or any non-blank character against the option which describes the close rule to be applied to data sets belonging to this index when the index is not in use and the number of open data sets threshold limit has been reached.

This option determines parameter CLOSE YES/NO in the generated CREATE INDEX statement.

- ◊ **Close first**
  Index data sets are always eligible for close when the index is not in use. (CLOSE YES)

- ◊ **Close after CLOSE YES data sets**
  Index data sets are eligible for close when the index is not in use, only after data sets for indexes assigned as being CLOSE YES have been closed first. (CLOSE NO)

Free page frequency>
For a LOAD or REORG operation, specifies the number of pages (0-255) that will be loaded before a leaving a page of free space. The count of pages loaded restarts following the free page. A value of 0 (zero) indicates that no free pages are to be left.

Percent free/page>
For a LOAD or REORG operation, specifies the percentage (0-99) of each page to be left as free space.

Data Sharing Group Buffer Pool Caching:
Applicable only in a data sharing environment, enter "/" to select what pages of the index space or partition are written to the group buffer pool. In a non-data sharing environment, this option will be ignored. Options are as follow:

- ◊ **Changed data pages**
  Unless defined in a group buffer pool that is defined to be used only for cross-invalidation, cache only those index space pages containing index data that has been modified.

- ◊ **ALL data pages read**
  Cache all index space pages as they are read from DASD.

- ◊ **No caching**
  Do not cache index space pages in the group buffer pool.

Index Space

The Index Space panel view is the next view in the sequence following the Index Options panel view when the index is non-partitioned, an XML index or an index on an Auxiliary table.

This panel view specifies attributes of the single partition index data sets and of the storage into which they are loaded when the index is used.

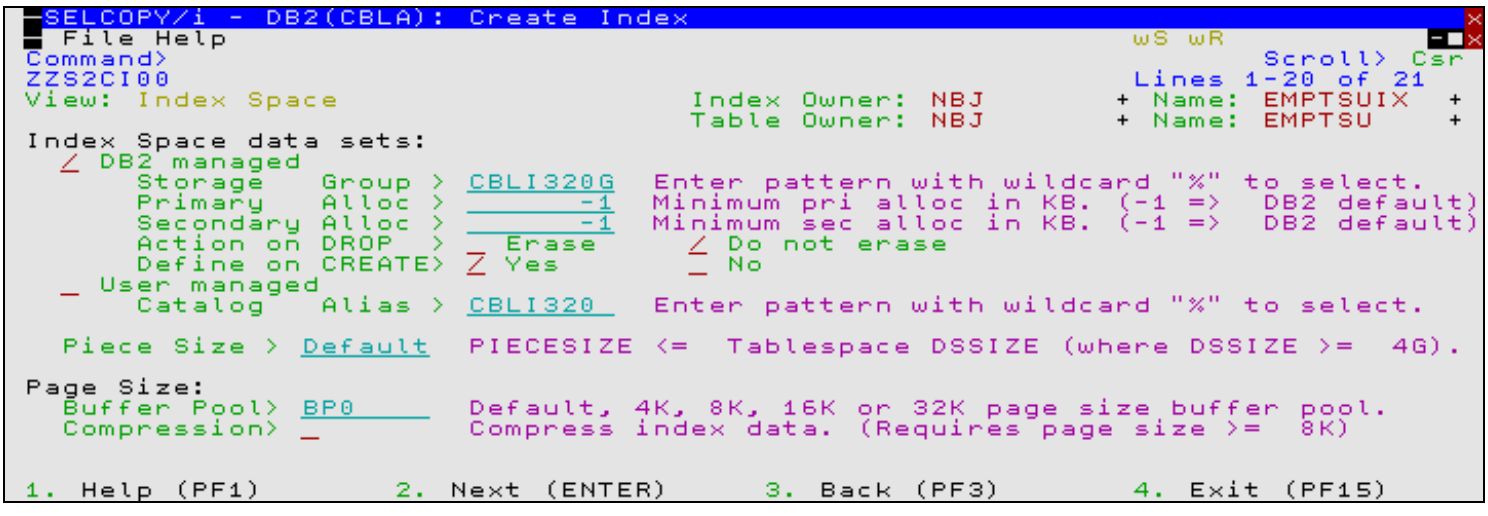


Figure 244. DB2: Create Index - Index Space.

Index Space - Panel Fields

Index Owner:
A non-enterable field displaying the owner id of the new index as supplied in the **Index & Table Name** panel view.

Name:
A non-enterable field displaying the new index name as supplied in the **Index & Table Name** panel view.

Table Owner:
A non-enterable field displaying the table owner id as supplied in the **Index & Table Name** panel view.

Name:
A non-enterable field displaying the table name as supplied in the **Index & Table Name** panel view.

Index Space Data sets:
Enter "/" to select the method by which index space data sets will be managed and enter values appropriate to the selected method. Options are as follow:

- ◊ **DB2 managed**
  Specifies that DB2 will define and manage the index data sets. Each data set will be defined on a volume of the identified storage group with primary and secondary allocations sizes defined by the Primary and Secondary Alloc fields respectively.

- ◊ **User managed**
  Specifies that index space data sets are to be managed by the user. Index data sets are linear VSAM data sets cataloged in an ICF catalog identified by the catalog name/alias defined by the Catalog Alias field.

Storage Group>
Mandatory field for DB2 managed data sets which specifies the name of a storage group defined in the current DB2 server.

The storage group name may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the storage group name. If this is the case, a Select Storage Group panel is displayed containing a list of eligible storage groups which satisfy the name filter. Use prefix command "S", or press the <Enter> key on the required entry to select it.

Primary Alloc>
Mandatory field for DB2 managed data sets which specifies the minimum primary allocation size in KB for the index data sets.

A value of -1 indicates to DB2 that it should use a value based on system defaults. A value other than -1 may be adjusted by DB2 to satisfy minimum requirements.

Secondary Alloc>
Mandatory field for DB2 managed data sets which specifies the minimum secondary allocation size in KB for the index data sets.

A value of -1 indicates to DB2 that it should use a value based on system defaults.

Action on DROP>
Enter "/" to select whether or not index data sets will be erased when the index is deleted on execution of a utility or dropped using an SQL statement.

Define on CREATE>
This field is not displayed for an XML Index or an index on an Auxiliary table.

Enter "/" to select whether index data sets will be allocated when the index is created (Yes) or delayed until data is inserted into the index (No).

Catalog Alias>
Mandatory field for User managed data sets which specifies the name or alias of the ICF catalog in which the table space data sets are to be cataloged.

The catalog alias may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the storage group name. If this is the case, a Select Catalog Alias panel is displayed containing a list of eligible aliases that satisfy the model catalog alias filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

Piece Size>
Specifies the maximum addressability of each index data set.

This field is initialised as "Default" which results in no specific PIECESIZE value being specified in the resultant CREATE INDEX statement.
Enter blanks or any invalid value to select from a list of valid piece size values.

Buffer Pool>
Specifies the 4K, 8K, 16K or 32K buffer pool name to be used for the index and so determines the index space page size.

This field is initialised as "Default" which results in no specific BUFFERPOOL value being specified in the resultant CREATE INDEX statement.
Enter blanks or any invalid value to select from a list of valid buffer pool names.

Compression>
Enter "/" to specify that index data sets will be compressed. Compression requires a buffer pool page size of 8K, 16K or 32K.

Partitioned Index Space

The Partitioned Index Space panel view is the next view in the sequence following the Index Options panel view when the index is partitioned.

This panel view specifies default attributes of each partitioned index data set, the storage into which thay are loaded when the index is used and an option to specify individual partition or index-controlled partitioning index data set attributes.

For partitioned (non-universal) table spaces for which partitioning column limit values have not yet been defined, customising individual partition data set attributes is mandatory. Therfore, in this case, the Index Partitioning Attributes sub-panel is opened automatically so that partitioning column limits may be entered for the index key columns. i.e. Table space partitioning will be index-controlled.

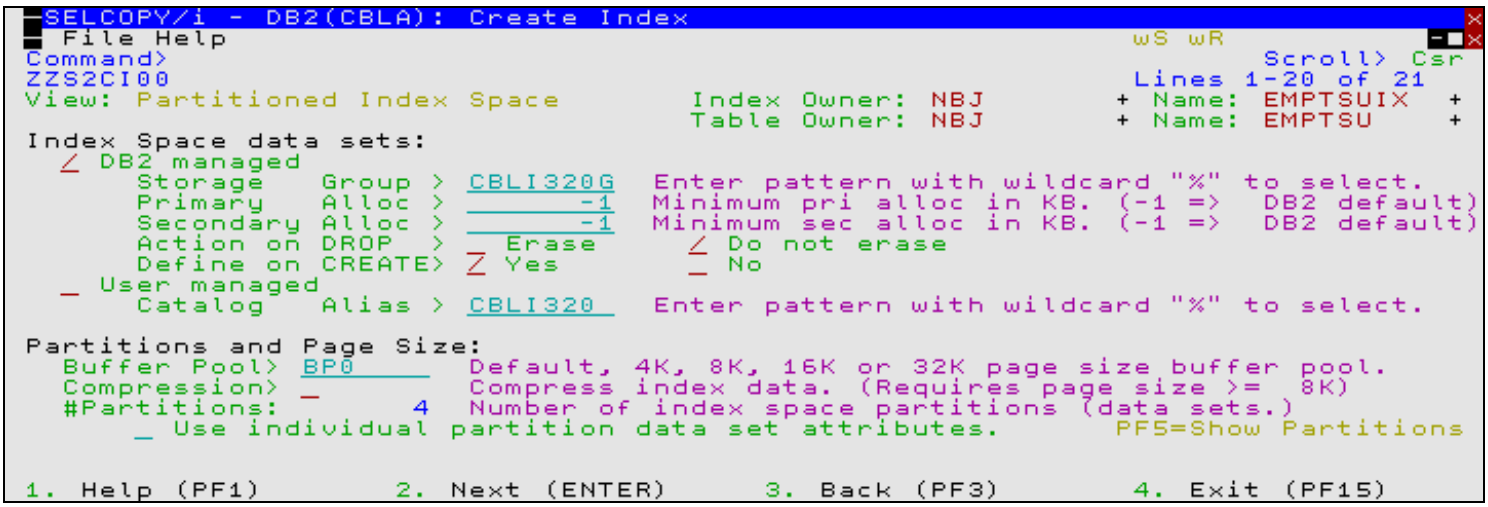


Figure 245. DB2: Create Index - Partitioned Index Space.

Partitioned Index Space - Panel Fields

Index Owner:
A non-enterable field displaying the owner id of the new index as supplied in the **Index & Table Name** panel view.

Name:
A non-enterable field displaying the new index name as supplied in the **Index & Table Name** panel view.

Table Owner:
A non-enterable field displaying the table owner id as supplied in the **Index & Table Name** panel view.

Name:
A non-enterable field displaying the table name as supplied in the **Index & Table Name** panel view.

Index Space Data sets:
Enter "/" to select the method by which index space data sets will be managed and enter values appropriate to the selected method. Options are as follow:

- ◊ **DB2 managed**
  Specifies that DB2 will define and manage the index data sets. By default, each data set will be defined on a volume of the identified storage group with primary and secondary allocations sizes defined by the Primary and Secondary Alloc fields respectively.

- ◊ **User managed**
  Specifies that index space data sets are to be managed by the user. Index data sets are linear VSAM data sets cataloged in an ICF catalog which, by default, is identified by the catalog name/alias defined by the Catalog Alias field.

Storage Group>
Mandatory field for DB2 managed data sets which specifies the name of a storage group defined in the current DB2 server.

The storage group name may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the storage group name. If this is the case, a Select Storage Group panel is displayed containing a list of eligible storage groups which satisfy the name filter. Use prefix command "S", or press the <Enter> key on the required entry to select it.

Primary Alloc>
Mandatory field for DB2 managed data sets which specifies the minimum primary allocation size in KB for the index data sets.

A value of -1 indicates to DB2 that it should use a value based on system defaults. A value other than -1 may be adjusted by DB2 to satisfy minimum requirements.

Secondary Alloc>
Mandatory field for DB2 managed data sets which specifies the minimum secondary allocation size in KB for the index data sets.

A value of -1 indicates to DB2 that it should use a value based on system defaults.

Action on DROP>
Enter "/" to select whether or not index data sets will be erased when the index is deleted on execution of a utility or dropped using an SQL statement.

Define on CREATE>
Enter "/" to select whether index data sets will be allocated when the index is created (Yes) or delayed until data is inserted into the index (No).

Catalog Alias>
Mandatory field for User managed data sets which specifies the name or alias of the ICF catalog in which the table space data sets are to be cataloged.

The catalog alias may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the storage group name. If this is the case, a Select Catalog Alias panel is displayed containing a list of eligible aliases that satisfy the model catalog alias filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

Buffer Pool>
Specifies the 4K, 8K, 16K or 32K buffer pool name to be used for the index and so determines the index space page size.

This field is initialised as "Default" which results in no specific BUFFERPOOL value being specified in the resultant CREATE INDEX statement.

Enter blanks or any invalid value to select from a list of valid buffer pool names.

Compression>
Enter "/" to specify that index data sets will be compressed. Compression requires a buffer pool page size of 8K, 16K or 32K.

#Partitions:
A non-enterable field displaying the number of defined table space partitions and, therefore, the number of index space partitions for the index.

Use individual partition data set attributes.
Enter "/" to indicate that data set attributes are to be entered for one or more partitions in the index definition.

These may override the default values entered in this panel view for Storage Group, Primary Alloc, Secondary Alloc, Action on Drop and Catalog Alias, and in the Index Options panel view for Free page frequency, Percent free/page and Group Buffer Pool Caching.

If this option is selected and individual partition data set overrides have not yet been selected, then the Index Partition/Partitioning Attributes sub-panel is automatically opened.

Once partition overrides have been selected, thereafter, this sub-panel may be opened from this panel view using primary command, **SELect** (assigned to <F5> by default). If SELECT is actioned before partition overrides have been entered then the "Use individual partition data set attributes" option is automatically selected.

Index Partition/Partitioning Attributes

The Index Partition Attributes or Index Partitioning Attributes sub-panel is displayed from the Partitioned Index Space panel view via one of the following methods:

1. Automatically when **Use individual partition data set attributes** is selected and partition attribute overrides have not yet been entered and selected.
2. Enter primary command **SELect** (default for <F5>).

This sub-panel contains an embedded table of a pre-configured number of rows, one for each partition in the index space. The table is for update only so that the rows may not be manipulated using the standard table editing techniques.

Each table row identifies a single partition and has data set attribute columns initialised to be the default values set in previously displayed panel views. Zoom of an individual row displays a formatted view of the row data.

In addition to data set attribute columns, the **Index Partitioning Attributes** sub-panel has the table column, **Partitioning Column Limits**, in which partition limit values must be entered for each column in the index key. Since provision of partitioning limit values is mandatory, this column replaces the SELect column found in the **Index Partition Attributes** sub-panel.

The Index Partition Attributes sub-panel supports primary commands **SELECTAll** and **REFRESH**, to toggle selection and deselection of all columns and reset display of partition dataset attributes. (By default, SELECTALL is assigned to <F5>, SELECTALL DESELECT is assigned to <F6> and REFRESH is assigned to <F16>.)

Validation is performed for all partition definitions on exiting the Index Partitioning Attributes sub-panel. If limit values have not been specified for all partitioning columns of a partition entry, then the Partitioning Key Column Values sub-panel will be opened for that

partition. To bypass validation, use CANCEL to exit the panel.

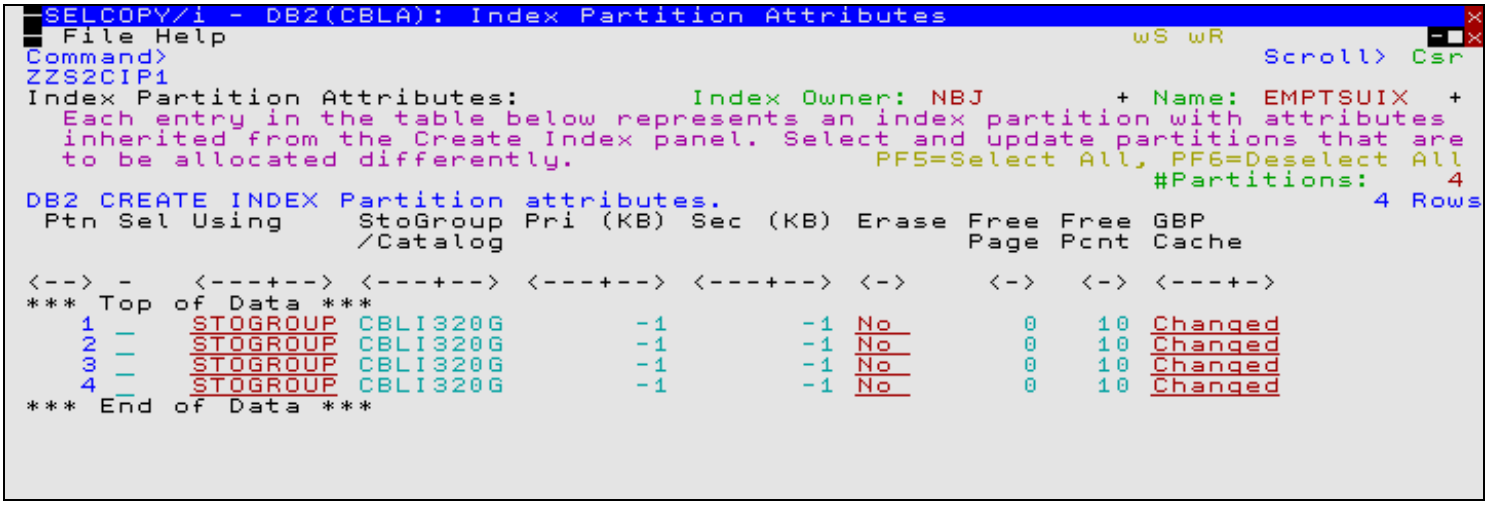


Figure 246. DB2: Create Index - Index Partition Attributes.

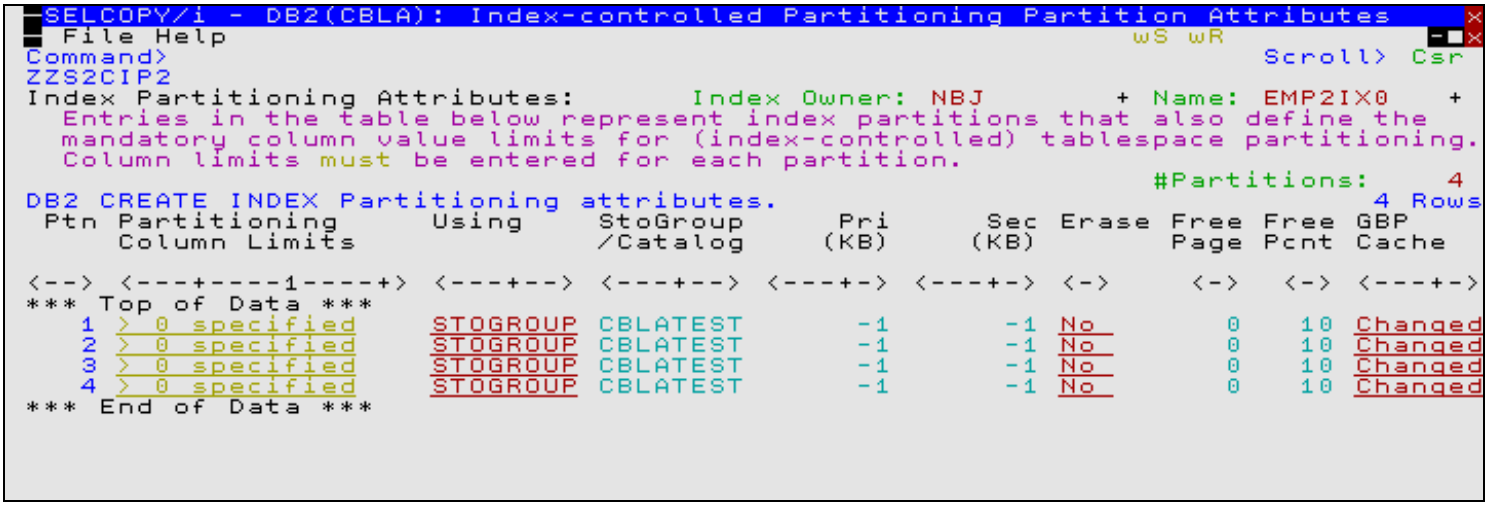


Figure 247. DB2: Create Index - Index Partitioning Attributes.

Index Partition/Partitioning Attributes - Panel Fields

Field names that follow are as appear in the zoomed view of a table row. Names in parentheses correspond to the equivalent column name in table view.

`Index Owner:`
A non-enterable field displaying the owner id of the new index as supplied in the **Index & Table Name** panel view.

`Name:`
A non-enterable field displaying the new index name as supplied in the **Index & Table Name** panel view.

`#Partitions:`
A non-enterable field displaying the number of defined table space partitions and, therefore, the number of index space partitions for the index.

`Partition Number: (Ptn)`
A non-enterable field displaying the table space partition number.

`Sel> (Sel)`
This field is applicable only to the **Index Partition Attributes** sub-panel.

Enter "/" (slash) or any non-blank character in this field to include (select) the attributes for the partition number in the CREATE INDEX syntax. A blank in this field will exclude (deselect) the partition attributes.

Deselecting a column definition in table view has the same effect as executing the table primary command EXCLUDE (or line command "X") to exclude the table row.

`Column Limits> (Partition Column Limits)`
This field is applicable only to the **Index Partitioning Attributes** sub-panel.

In table view, this field displays the number of partitioning columns for which a partitioning limit value has been specified. Position the cursor on an entry in this column for the required tablespace partition and press <Enter> or, if configured, double-click the left mouse button to display the Partitioning Key Column Values sub-panel.

In single row view, this field is an option which, when selected will display the **Partitioning Key Column Values** sub-panel.

Cols:
This field is applicable only to the **Index Partitioning Attributes** sub-panel.

Displayed in single row view only, this is a non-enterable field displaying the number of columns that comprise the index key and for which limit values must be specified.

Index Space data sets: (Using)
A value of STOGROUP or VCAT representing options "DB2 Managed" or "User Managed" respectively.

Options are as described for the Index Space Data sets: field of the Partitioned Index Space panel view.

Storage Group> (StoGroup/Catalog)
Storage Group value as described for the Storage Group field of the **Partitioned Index Space** panel view.

Primary Alloc> (Pri (KB))
Primary allocation value as described for the Primary Alloc field of the **Partitioned Index Space** panel view.

Secondary Alloc> (Sec (KB))
Secondary allocation value as described for the Secondary Alloc field of the **Partitioned Index Space** panel view.

Action on DROP> (Erase)
Data set erase option as described for the Action on DROP field of the **Partitioned Index Space** panel view.

Catalog Alias> (StoGroup/Catalog)
Catalog alias name as described for the Catalog Alias field of the **Partitioned Index Space** panel view.

Free page frequency> (Free Page)
Number of pages loaded after which a page of free space will be left. This is as described for the Free page frequency field of the Index Options panel view.

Percent free/page> (Free Pcnt)
Percentage of space left free per page as described for the Percent free/page field of the **Index Options** panel view.

Group Buffer Pool Caching: (GBP Cache)
For data sharing only, identifies the type of group buffer pool caching as described for the Group Buffer Pool Caching option field of the **Index Options** panel view.

Partitioning Key Column Values

The Partitioning Key Column Values sub-panel is displayed on selecting **Partition Column Limits** from the Index Partitioning Attributes sub-panel. Alternatively, the sub-panel may be opened if limit values have not been entered for each partitioning column in each partition when the Index Partitioning Attributes sub-panel is closed.

This sub-panel contains an embedded table of DB2 table space partitioning key columns. The table is for update only so that rows may not be manipulated using the standard table editing techniques.

Each table row identifies a partitioning column definition that correspond to the columns selected as the index key. Apart from the Limit Value column, all columns in this table are non-enterable and are included for information only.

Following update of a table row (limit value) or on returning to the table view from a zoomed view, validation occurs for the limit value entered by the user. Likewise, validation is performed for all column limit values on exiting the Partitioning Key Column Values sub-panel. If an error is detected, then the table row containing the limit value in error is automatically zoomed. This allows the user to correct the error before continuing. To bypass validation, use CANCEL to exit the panel.

If the width of the Limit Value field in the table view is not sufficient to enter the required limit value, then the table row should be zoomed (default <F17>) and, if necessary, the appropriate field expanded (default <F14>) to accomodate the input value.

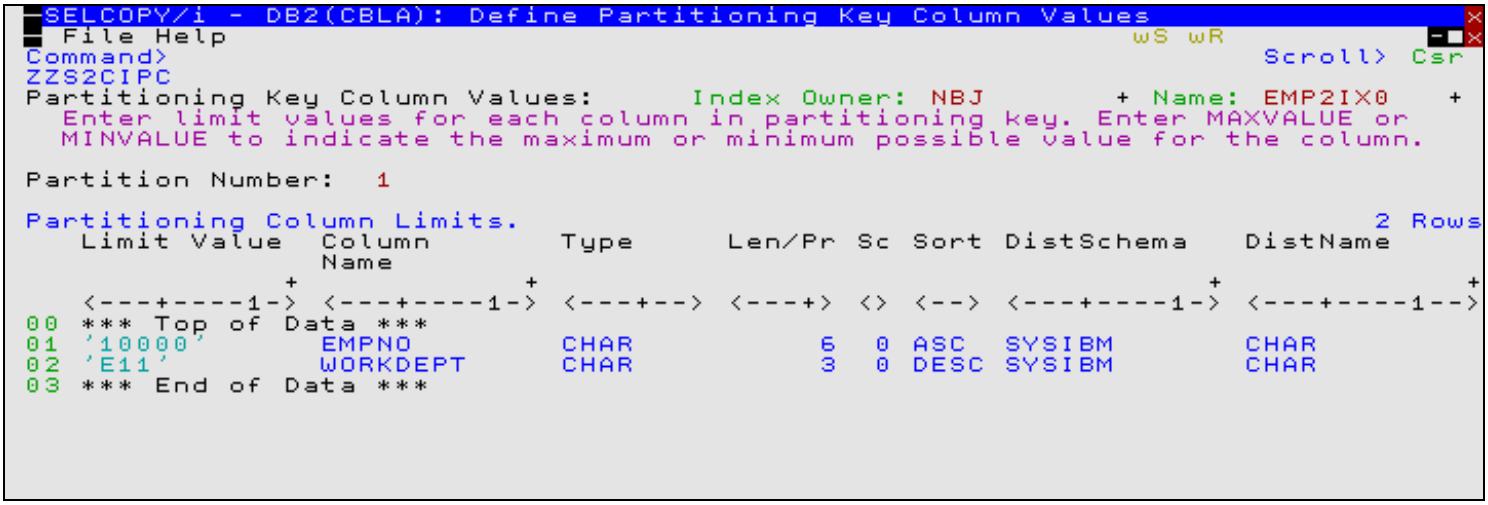


Figure 248. DB2: Create Index - Partitioning Key Column Values.

Partitioning Key Column Values - Panel Fields

Field names that follow are as appear in the zoomed view of a table row. Names in parentheses correspond to the equivalent column name in table view.

Index Owner:
A non-enterable field displaying the index owner id as supplied in the **Index & Table Name** panel view.

Name:
A non-enterable field displaying the index name as supplied in the **Index & Table Name** panel view.

Partition Number:
A non-enterable field displaying the current table space partition number.

Limit Value> (Limit Value)
Specifies the partition limit value for the column within the partitioning key.

If the column is a character data type, then the limit value must be enclosed within SQL delimiter characters as defined for the DB2 server. If these character string delimiters have been omitted, then they will be added automatically by the panel, escaping any delimiter character that exists as data within the string.

Exceptions to this rule are values MAXVALUE and MINVALUE (which may be entered with a minimum abbreviation of MAX and MIN) which represent maximum and minimum possible values for the column regardless of the column's defined data type. To enter MAXVALUE or MINVALUE (or their abbreviations) as string data, they must explicitly be entered within string delimiter characters.

Column Name: (Column Name)
A non-enterable field displaying the name of the column.

DataType: (Type)
A non-enterable field displaying the type of the column.

Precision/Length: (Len/Pr)
A non-enterable field displaying the length or precision of the column data.

Scale (Sc)
A non-enterable field displaying the scale of the column data.

Value Sort Order: (Sort)
A non-enterable field displaying ASC or DESC, indicating the order in which entries are sorted by this column (ascending or descending).

DistType Schema: (DistSchema)
A non-enterable field displaying the schema of the distinct type assigned to the column. If no distinct type is used, this field will be blank.

DistType Name: (DistName)
A non-enterable field displaying the name of the distinct type assigned to the column. If no distinct type is used, this field will be blank.

XML Index

The XML Index panel view is the next view in the sequence following the Index Type panel view if XML has been selected.

This view allows the user to specify an XML Pattern (consisting of optional XML Namespace declarations and an XML Path) that identifies the element, attribute ot text node on which indexing will occur. It also identifies the SQL data type used to index the XML node values and the uniqueness rule enforced on indexed entries.

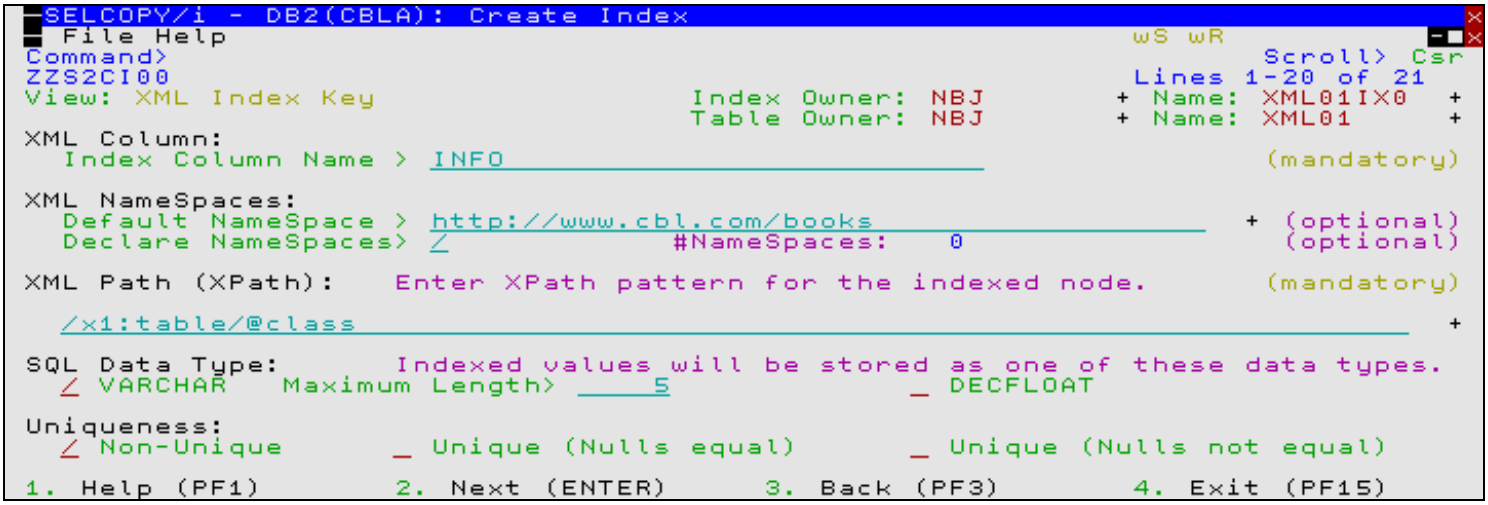


Figure 249. DB2: Create Index - XML Index.

XML Index - Panel Fields

Index Owner:
A non-enterable field displaying the owner id of the new index as supplied in the **Index & Table Name** panel view.

Name:
A non-enterable field displaying the new index name as supplied in the **Index & Table Name** panel view.

Table Owner:
A non-enterable field displaying the table owner id as supplied in the **Index & Table Name** panel view.

Name:
A non-enterable field displaying the table name as supplied in the **Index & Table Name** panel view.

Index Column Name>
Specifies the name of the XML column on which the index will be based.

If this field is blank, the name of the XML column will be inserted automatically. Otherwise, if the field contains a wildcard character, a column selection panel is displayed allowing the user to select the XML column name.

Default NameSpace>
Optionally specifies the name (URI) of the default NameSpace to be applied to any non-prefixed element or attribute node name.

Wildcard character "%" (percent) or "*" (asterisk), representing zero or more characters, or wildcard character "_" (underscore), representing a single character, may be entered in this field. This opens an XML namespace selection panel which allows selection of a namespace URI that has previous been defined to the current DB2 server.

Declare NameSpaces>
Enter "/" to indicate that XML names are to be defined for NameSpace URIs. Thes XML names are used as prefixes to element and attribute node names in the document text of the XML column data.

If this option is selected and XML NameSpace prefixes have not yet been defined and selected, then the XML Index Namespace Prefixes sub-panel is automatically opened.

Once an XML NameSpace prefix has been defined and selected, thereafter, this sub-panel may be opened from this panel view using primary command, **SELect** (assigned to <F5> by default). If SELECT is actioned before partition overrides have been entered then the "Declare NameSpaces" option is automatically selected.

#NameSpaces:
A non-enterable field displaying the number of NameSpace prefix that have been defined and selected in the XML Index Namespace Prefixes sub-panel.

XML Path:
Specifies the name of the XML path (XPath) to the element, attribute or text node(s) on which the index entries are derived.

The XPath is comprised of forward-axis steps and node specifications as described by CREATE INDEX *pattern-expression* in the IBM manual, *"DB2 SQL Reference"*.

SQL Data Type:
Enter "/" to select the SQL data type used to store indexed values. Options are as follow:

- ◊ **VARCHAR**
  Values will be stored as variable length character data with a maximum length defined by the Maximum Length input field.

- ◊ **DECFLOAT**
  Values will be stored as a decimal floating point value of precision 34.

Maximum Length>
Applicable only if SQL data type VARCHAR has been selected, this field specifies the defined maximum length (1-1000) of the variable length character value.

Uniqueness:
Enter "/" (slash) or any non-blank character to select the level of uniqueness of data in the indexed XML node values. Each of the options are mutually exclusive.

- ◊ **Non-Unique**
  The table may contain two or more rows that have the same XML indexed node value.

- ◊ **Unique (Nulls equal)**
  The table may **not** contain two or more rows that have the same XML indexed node value. In determining whether the node values are unique, DB2 will treat null values in an indexed node as being equal.

- ◊ **Unique (Nulls not equal)**
  The table may **not** contain two or more rows that have the same XML indexed node value. In determining whether the node values are unique, DB2 will treat null values in an indexed node as being not equal.

XML Index Namespace Prefixes

The XML Index Namespace Prefixes sub-panel is displayed from the XML Index panel view via one of the following methods:

1. Automatically when **Declare NameSpace** is selected and Namespace prefixes have not yet been entered and selected.
2. Enter primary command **SELect** (default for <F5>).

This sub-panel contains an embedded table of XML names and Namespace URIs to which they are assigned.

Standard table edit primary and line commands must be used to INSERT, DELETE, COPY, MOVE or EXCLUDE table rows as appropriate, to scroll the table display UP, DOWN, LEFT and RIGHT and also to ZOOM the display of an individual table row.

In addition to the standard primary table commands, this sub-panel also supports primary command **SELECTAll** to toggle selection and deselection of all columns and expressions. (By default, SELECTALL is assigned to <F5> and SELECTALL DESELECT is assigned to <F6>.)

If the width of the Namespace URI field in the table view is not sufficient to enter the required input value, then the table row should be zoomed (default <F17>) and, if necessary, the field expanded (default <F14>) to accomodate the input value.

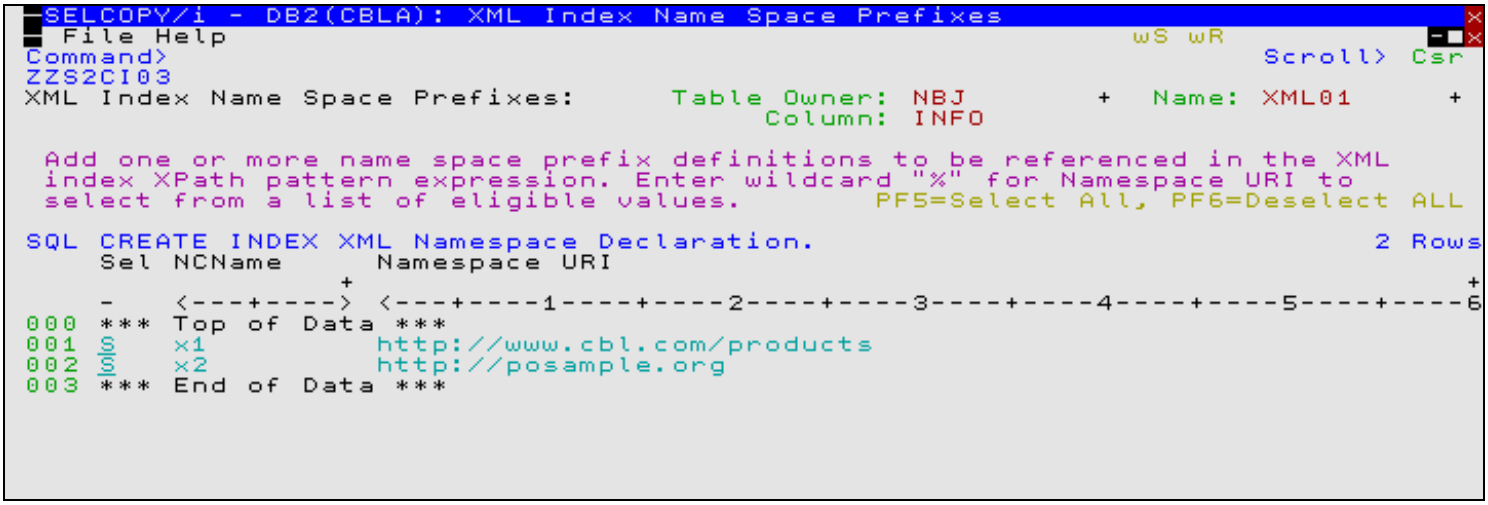


Figure 250. DB2: Create Index - XML Index Namespace Prefixes.

XML Index Namespace Prefixes - Panel Fields

Field names that follow are as appear in the zoomed view of a table row. Names in parentheses correspond to the equivalent column name in table view.

`Table Owner:`
A non-enterable field displaying the table owner id as supplied in the **Index & Table Name** panel view.

`Name:`
A non-enterable field displaying the table name as supplied in the **Index & Table Name** panel view.

`Column:`
A non-enterable field displaying the indexed XML column name.

`Sel> (Sel)`
Enter "/" (slash) or any non-blank character in this field to include (select) the Namespace prefix definition. A blank in this field will exclude (deselect) the definition.

Deselecting a column in table view has the same effect as executing the table primary command EXCLUDE (or line command "X") to exclude the table row.

`Name Space Prefix> (NCName)`
Specifies the XML non-colonized name (NCName) to be assigned as the XML Namespace prefix.

`Name Space URI> (Namespace URI)`
Specifies the name (URI) of the NameSpace to which the XML name will be assigned.

Wildcard character "%" (percent) or "*" (asterisk), representing zero or more characters, or wildcard character "_" (underscore), representing a single character, may be entered in this field. This opens an XML namespace selection panel which allows selection of a namespace URI that has previous been defined to the current DB2 server.

Generate SQL

The Generate SQL panel view is the last panel view in the sequence and is common to all SELCOPY/i DB2 Create Objects panel view sequences. For details, please refer to Generate SQL under the description of the DB2 Create Table sequence of panels.

This view allows the user to select how the generated SQL CREATE INDEX statement is to be implemented.

Primary Commands

The following primary commands are supported by selected views in the DB2 Create Index sequence of panel views. If issued from a panel view in which the command is not valid, the message "ZZSP102E Primary command not valid in the current context" is displayed.

CMX

```
>>--+- CMX ----------------------+-----------------------------------><
    +- EDITCMX -------------------+
```

Applicable only from the Generate SQL view, CMX generates the SQL statement and copies it to an in-storage output file in a format suitable for subsequent execution using the ACTION (default >F16>) facility.

CMX is assigned to <F17> by default.

JCL

```
>>--+- JCL ----------------------+-----------------------------------><
    +- EDITJCL -------------------+
```

Applicable only from the Generate SQL view, JCL generates the SQL statement and copies it to an in-storage output file with JCL statements that execute the DSNTIAD facility. This job may be submitted to batch using the SELCOPY/i text editor SUBMIT primary command.

JCL is assigned to <F18> by default.

KEYCOLUMNS

```
>>--+- KEYcolumns ----------------+-------------------------------------><
   -+- INDEXKey -----------------+
```

Supported for non-XML indexes only. If an Index Type is selected that supports use of and index key expression, the KEYCOLUMNS opens the Index Key Columns and Expressions sub-panel. Otherwise, it opens the Index Key Columns sub-panel.

KEYCOLUMNS is assigned to <F6> by default.

RUN

```
>>--+- RUN -----------------------+-------------------------------------><
    +- EXECSYNtax ----------------+
```

Applicable only from the Generate SQL view, RUN verifies input fields in all applicable panel views and then executes the generated SQL statement. This is the default action on pressing <Enter> from the last panel view in the sequence (i.e. the **Generate SQL** view.)

SELECT

```
>>---- SELect ------------------------------------------------------><
```

SELECT will open a sub-panel that is applicable to the current panel view. Each of the panel views listed below identify the sub-panel opened on executing SELECT. In all other panel views, SELECT is invalid.

- **Partitioned Index Space**
  Opens one of the Index Partition/Partitioning Attributes sub-panels to specify individual partition information.
- **XML Index**
  Opens the XML Index Namespace Prefixes sub-panel to define individual XML namespace prefixes.

SELECT is assigned to <F5> by default.

Create View

The Create View sequence of panel views (ZZS2CVI0) generate an SQL CREATE VIEW statement to create a new DB2 view in the current DB2 server.

The DB2 Create View panel views are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select View option 9. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.9)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made in a the panel views. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL CREATE VIEW syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

View Definition

Enter the name of the DB2 view to be created in the current DB2 sub-system. The current DB2 subsystem is displayed in the panel window title bar.

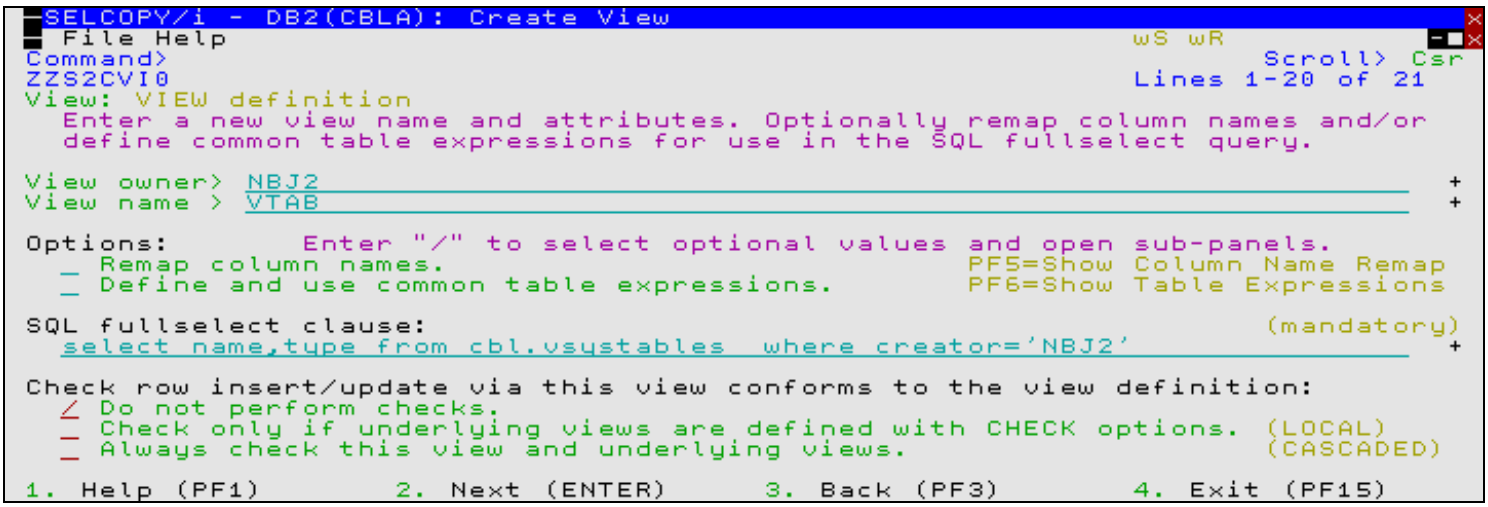


Figure 252. DB2: Create View.

Menu Bar Items

The following menu bar items are displayed in the Create View panel views.

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

View Definition - Panel Fields

View Owner>
: The owner (schema) of the new DB2 view.
The DB2 view owner value has a maximum length of 128 characters.

View Name>
: The name of the new DB2 view. The DB2 view name must not match the name of an existing DB2 table, view alias or synonym.
The DB2 view name has a maximum length of 128 characters.

This parameter field corresponds to SQL CREATE VIEW parameter *view-name*.

Remap column names
Enter "/" to indicate that the names of all columns selected by the SQL fullselect clause are to be remapped as different column names. If not specified, the columns names of the DB2 view inherit those of the results table generated by the fullselect.

If this option is selected and replacement column names have not yet been selected, then the Result Table Column Name Remap sub-panel is automatically opened.

Once column name replacements have been specified, thereafter, this sub-panel may be opened from this panel view using primary command, **COLMAP** (assigned to <F5> by default). If COLMAP is actioned before column remap definitions have been entered then the "Remap Column Names" option is automatically selected.

Define and use common table expressions
Enter "/" to indicate that one or more common table expressions are to be defined for reference by a FROM clause within the SQL fullselect clause.

If this option is selected and a common table expression has not yet been defined and selected, then the Common Table Expression Definitions sub-panel is automatically opened.

Once a common table expression has been specified, thereafter, this sub-panel may be opened from this panel view using primary command, **TABEXP** (assigned to <F6> by default). If TABEXP is actioned before a common table expression havs been entered then the "Define and use common table expressions" option is automatically selected.

SQL fullselect clause:
Specifies an SQL fullselect that defines the DB2 view. At any time, the DB2 view consists of the columns and rows that would result if the fullselect were executed.
This parameter field is mandatory and corresponds to SQL CREATE VIEW parameter *fullselect*.

Check row insert/update to a view
Enter "/" to select the CHECK action taken by DB2 when rows are updated or inserted using the DB2 view definition.

- ◊ **Do not perform checks**
  The definition of the DB2 view is **not** used to perform checks on inserted or updated rows. When row data is inserted or updated through the DB2 view, no check is made that the data conforms to search conditions specified by the DB2 view or any DB2 view on which it depends.

- ◊ **Cascaded**
  Updated and inserted rows must satisfy the search conditions of the DB2 view and **all** underlying DB2 views regardless of whether those underlying DB2 views were defined with the CHECK option.
  This option corresponds to SQL CREATE VIEW parameter WITH CASCADED CHECK OPTION.

- ◊ **Local**
  Updated and inserted rows must satisfy the search conditions of the DB2 view and all underlying DB2 views that have been defined with the CHECK option. No checking is performed for underlying DB2 views for which the CHECK option has been defined.
  This option corresponds to SQL CREATE VIEW parameter WITH LOCAL CHECK OPTIONS.

Result Table Column Name Remap

The Result Table Column Name Remap sub-panel is displayed from the View Definition panel view to define the column names used for columns selected by the DB2 view. It may also be displayed from the Common Table Expression Definitions sub-panel if specific column names are to be returned by a common table expression referenced by the DB2 view definition.

In both cases, the sub-panel may be displayed via one of the following methods:

1. For the View Definitions panel view or for the Common Table Expressions sub-panel displayed in zoomed view, the sub-panel will be displayed automatically when **Remap column names** is selected and column name remap entries have not yet been entered and selected.

2. For the View Definitions panel view or for the Common Table Expressions sub-panel displayed in zoomed view, enter primary command **COLMAP** (default for <F5>).

3. For Common Table Expressions sub-panel displayed in table view, select an entry in the **Rename Columns** field for the required expression definition.

This sub-panel contains an embedded table of column names to be assigned, in order of specification, to columns selected for the result table generated by the SQL fullselect assigned to the DB2 view definition. For reference only, the FullSelect statement is displayed in a non-enterable field.

These column names are used in place of the names of columns in the generated result table. Column name remap is mandatory if any of the result table columns are unnamed or have a duplicate name. Furthermore, the number of column names specified in this table must equal the number of columns in the result table.

Standard table edit primary and line commands must be used to INSERT, DELETE, COPY, MOVE or EXCLUDE table rows as appropriate and to scroll the table display UP, DOWN, LEFT and RIGHT.

A column name has a maximum length of 30 characters.

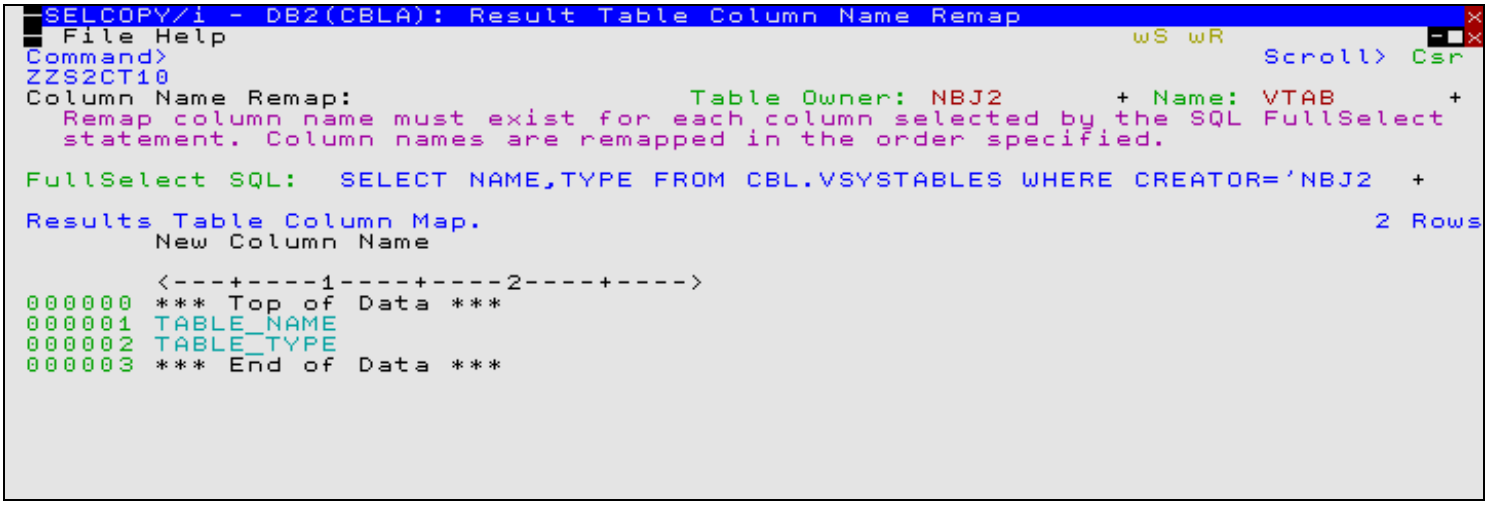


Figure 253. DB2: Create View - Result Table Column Name Remap.

Common Table Expression Definitions

The Common Table Expression Definitions sub-panel is displayed from the View Definition panel view via one of the following methods:

1. Automatically when **Define and use common table expressions** is selected and common table expressions have not yet been entered and selected.
2. Enter primary command **TABEXP** (default for <F6>).

This sub-panel contains an embedded table where each selected row defines a common table expression which may be referenced in the SQL fullselect query clause associated with the DB2 view.

Standard table edit primary and line commands must be used to INSERT, DELETE, COPY, MOVE or EXCLUDE table rows as appropriate, to scroll the table display UP, DOWN, LEFT and RIGHT and also to ZOOM the display of an individual table row.

In addition to the standard primary table commands, this sub-panel also supports primary command **SELECTAll** to toggle selection and deselection of all columns and expressions. (By default, SELECTALL is assigned to <F5> and SELECTALL DESELECT is assigned to <F6>.)

Validation is performed for all selected entries on exiting the Common Table Expression Definitions sub-panel. If an error is detected, then the row entry in error is automatically zoomed. To bypass validation, use CANCEL to exit the panel.

If the width of the Name or Fullselect Query fields in table view is not sufficient to enter the required input value(s), then the table row should be zoomed (default <F17>) and, if necessary, the field(s) expanded (default <F14>) to accomodate the input value.

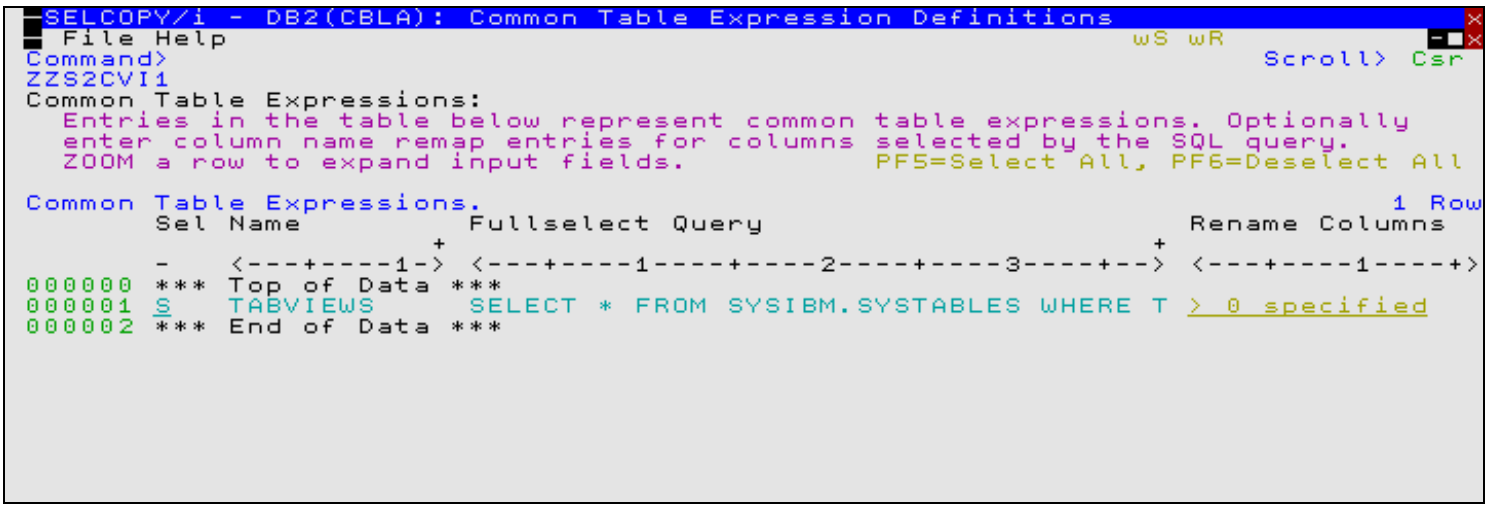


Figure 254. DB2: Create View - Common Table Expression Definitions.

Common Table Expression Definitions - Panel Fields

Field names that follow are as appear in the zoomed view of a table row. Names in parentheses correspond to the equivalent column name in table view.

Sel> (Sel)
Enter "/" (slash) or any non-blank character in this field to include (select) the common table expression definition for inclusion in the CREATE VIEW syntax. A blank in this field will exclude (deselect) the partition attributes.

Deselecting a common table expression in table view has the same effect as executing the table primary command EXCLUDE (or line command "X") to exclude the table row.

Expression Name> (Name)
The name of the common table expression. This name must not match the name of the DB2 view being created. The common table expression name has a maximum length of 128 characters.

Remap column names. (Rename Columns)
In table view, this field displays the number of column name remaps specified for columns selected by the common table fullselect clause. Position the cursor on an entry in this column for the required common table expression definition and press <Enter> or, if configured, double-click the left mouse button to display the Result Table Column Name Remap sub-panel.

In single row view, this field is an option selected on entering "/". If this option is selected and replacement column names have not yet been selected, then the **Result Table Column Name Remap** sub-panel is automatically opened.

Once column name replacements have been specified, thereafter, the sub-panel may be opened from this zoomed sub-panel view using primary command, **COLMAP** (assigned to <F5> by default). If COLMAP is actioned before column remap definitions have been entered then the "Remap Column Names" option is automatically selected.

SQL fullselect clause: (Fullselect Query)
Specifies an SQL fullselect that defines the common table expression. At any time, the common table expression consists of the columns and rows that would result if the fullselect were executed.

Generate SQL

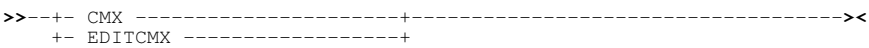
The Generate SQL panel view is the last panel view in the sequence and is common to all SELCOPY/i DB2 Create Objects panel view sequences. For details, please refer to Generate SQL under the description of the DB2 Create Table sequence of panels.

This view allows the user to select how the generated SQL CREATE VIEW statement is to be implemented.

Primary Commands

The following primary commands are supported by selected views in the DB2 Create View sequence of panel views. If issued from a panel view in which the command is not valid, the message "ZZSP102E Primary command not valid in the current context" is displayed.

CMX



Applicable only from the Generate SQL view, CMX generates the SQL statement and copies it to an in-storage output file in a format suitable for subsequent execution using the ACTION (default >F16>) facility.
CMX is assigned to <F17> by default.

COLMAP

Applicable to both the View Definition and Common Table Expression Definitions panel views, COLMAP opens the Result Table Column Name Remap sub-panel, used to define a column name AS new column name clause for each column in the DB2 result table.
COLMAP is assigned to <F5> by default.

JCL

```
>>--+- JCL ----------------------+-------------------------------------><
    +- EDITJCL ------------------+
```

Applicable only from the Generate SQL view, JCL generates the SQL statement and copies it to an in-storage output file with JCL statements that execute the DSNTIAD facility. This job may be submitted to batch using the SELCOPY/i text editor SUBMIT primary command.
JCL is assigned to <F18> by default.

RUN

```
>>--+- RUN ----------------------+-------------------------------------><
    +- EXECSYNtax ---------------+
```

Applicable only from the Generate SQL view, RUN verifies input fields in all applicable panel views and then executes the generated SQL statement. This is the default action on pressing <Enter> from the last panel view in the sequence (i.e. the **Generate SQL** view.)

TABEXP

```
>>---- TABEXP ------------------------------------------------------><
```

Applicable only to the View Definition panel view, TABEXP opens the Common Table Expression Definitions sub-panel, used to define one or more common table expressions for reference by a FROM clause within the DB2 View SQL fullselect clause.
TABEXP is assigned to <F6> by default.

Create Alias

The Create Alias sequence of panel views (ZZS2CAL0) generate an SQL CREATE ALIAS statement to create a new DB2 alias in the current DB2 server for an existing DB2 table or view.

The DB2 Create Alias panel views are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select Alias option 10. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.10)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made in a the panel views. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL CREATE ALIAS syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

Alias Values

Enter the name of the alias to be created in the current DB2 sub-system and the name of the DB2 table or view to be aliased. The DB2 table or view may be one that exists in another DB2 server as indicated by a specidfied location. The current DB2 subsystem is displayed in the panel window title bar.

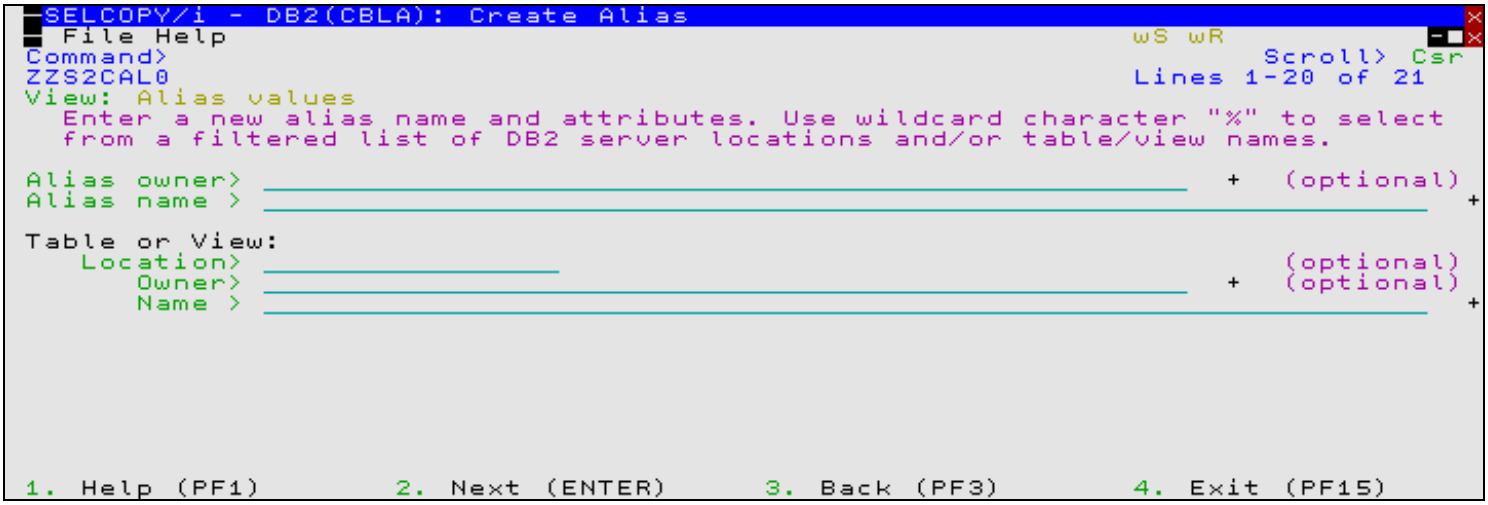


Figure 255. DB2: Create Alias.

Menu Bar Items

The following menu bar items are displayed in the Create Alias panel views.

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Alias Values - Panel Fields

Alias Owner>
: The owner (schema) of the new DB2 alias.
The DB2 alias owner value has a maximum length of 128 characters.

Alias Name>
: The name of the new DB2 alias. The DB2 alias must not match the name of an existing DB2 alias, table, view or synonym.
The DB2 alias name has a maximum length of 128 characters.

This parameter field corresponds to SQL CREATE ALIAS parameter *alias-name*.

Table or View:
Fields that together identify the DB2 table or view on which the alias is defined.

These field values may include wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the DB2 table or view location/owner/name specification.

If one or more of these wildcards is specified a selection panel will be displayed as indicated in the individual field descriptions below. In the selection panel, use prefix command "S", or press the <Enter> key on the required entry to select it.

Location>
The DBMS location at which the table or view is defined.
The table or view location name has a maximum length of 16 characters.

If a wildcard character is specified, the Select Server Location panel is displayed containing a list of defined DB2 server locations which satisfy the table location filter.

Owner>
The owner (schema) of the table or view.
The table or view owner name has a maximum length of 128 characters.

If a wildcard character is specified, the Select Table panel is displayed containing a list of eligible DB2 table or view names which satisfy the table owner/name filter.

Name>
The SQL identifier name of the table or view.
The table or view name has a maximum length of 128 characters.

If a wildcard character is specified, the Select Table panel is displayed containing a list of eligible DB2 table or view names which satisfy the table owner/name filter.

This parameter field corresponds to SQL CREATE ALIAS parameter FOR *table-name* or FOR *view-name*.

Generate SQL

The Generate SQL panel view is the last panel view in the sequence and is common to all SELCOPY/i DB2 Create Objects panel view sequences. For details, please refer to Generate SQL under the description of the DB2 Create Table sequence of panels.

This view allows the user to select how the generated SQL CREATE ALIAS statement is to be implemented.

Create Synonym

The Create Synonym sequence of panel views (ZZS2CSY0) generate an SQL CREATE SYNONYM statement to create a new DB2 synonym in the current DB2 server for an existing DB2 table, view or alias.

The DB2 Create Synonym panel views are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select Synonym option 11. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.11)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made in a the panel views. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL CREATE SYNONYM syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

Synonym Values

Enter the name of the synonym to be created in the current DB2 sub-system and the name of the DB2 table, view or alias that it represents. The current DB2 subsystem is displayed in the panel window title bar.

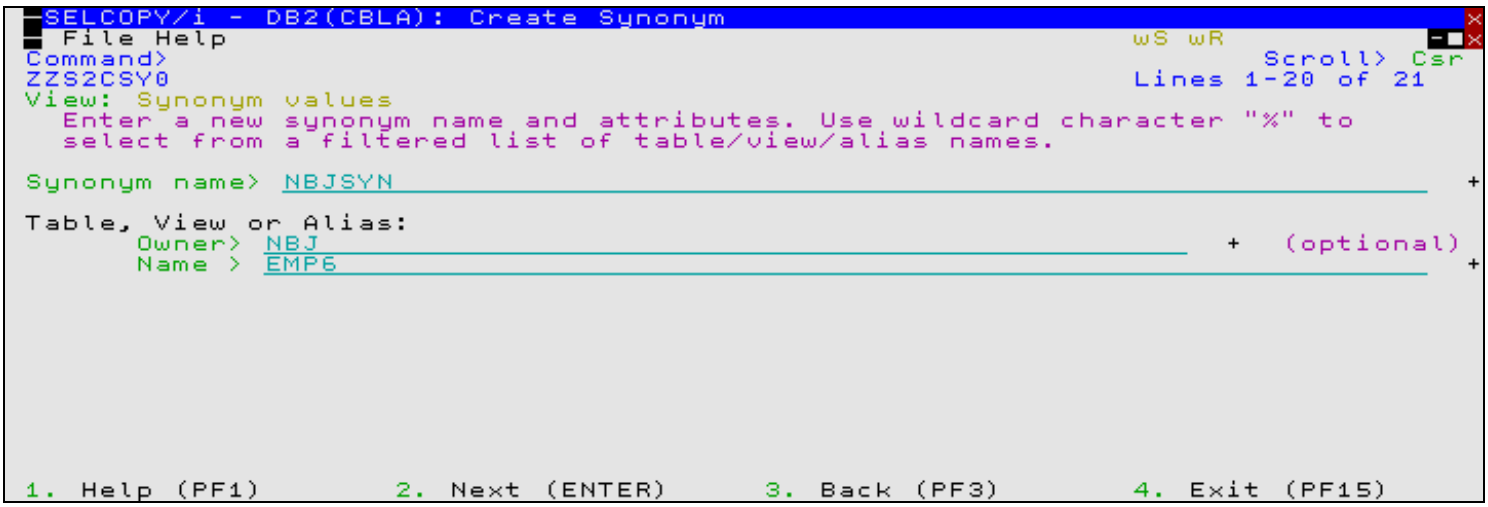


Figure 256. DB2: Create Synonym.

Menu Bar Items

The following menu bar items are displayed in the Create Synonym panel views.

File
The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

Synonym Values - Panel Fields

Synonym Name>
The name of the new DB2 synonym. The DB2 synonym must not match the name of an existing DB2 synonym, table, view or alias.
The DB2 synonym name has a maximum length of 128 characters.

This parameter field corresponds to SQL CREATE SYNONYM parameter *synonym*.

Table, View or Alias:
Fields that together identify the DB2 table, view or alias on which the synonym is defined.

These field values may include wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the owner or name specification.

If a wildcard character is specified, the Select Table panel is displayed containing a list of eligible DB2 table, view and alias names which satisfy the owner/name filter. In the selection panel, use prefix command "S", or press the <Enter> key on the required entry to select it.

Owner>
The owner (schema) of the table, view or alias.
The table, view or alias owner name has a maximum length of 128 characters.

Name>
The SQL identifier name of the table, view or alias.
The table, view or alias name has a maximum length of 128 characters.

This parameter field corresponds to SQL CREATE SYNONYM parameter FOR *authorization-name.table-name* or FOR *authorization-name.view-name*.

Generate SQL

The Generate SQL panel view is the last panel view in the sequence and is common to all SELCOPY/i DB2 Create Objects panel view sequences. For details, please refer to Generate SQL under the description of the DB2 Create Table sequence of panels.

This view allows the user to select how the generated SQL CREATE SYNONYM statement is to be implemented.

Create Distinct Type

The Create Distinct Type sequence of panel views (ZZS2CTY0) generate an SQL CREATE TYPE statement to create a new DB2 distinct type at the current DB2 server.

The DB2 Create Distinct Type panel views are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select Distinct Type option 12. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.12)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made in a the panel views. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL CREATE TYPE syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

Distinct Type Values

Enter the name of the distinct type and its source data type attributes to be created in the current DB2 sub-system. The source type attributes may be modelled on an existing distinct type definition. The current DB2 subsystem is displayed in the panel window title bar.

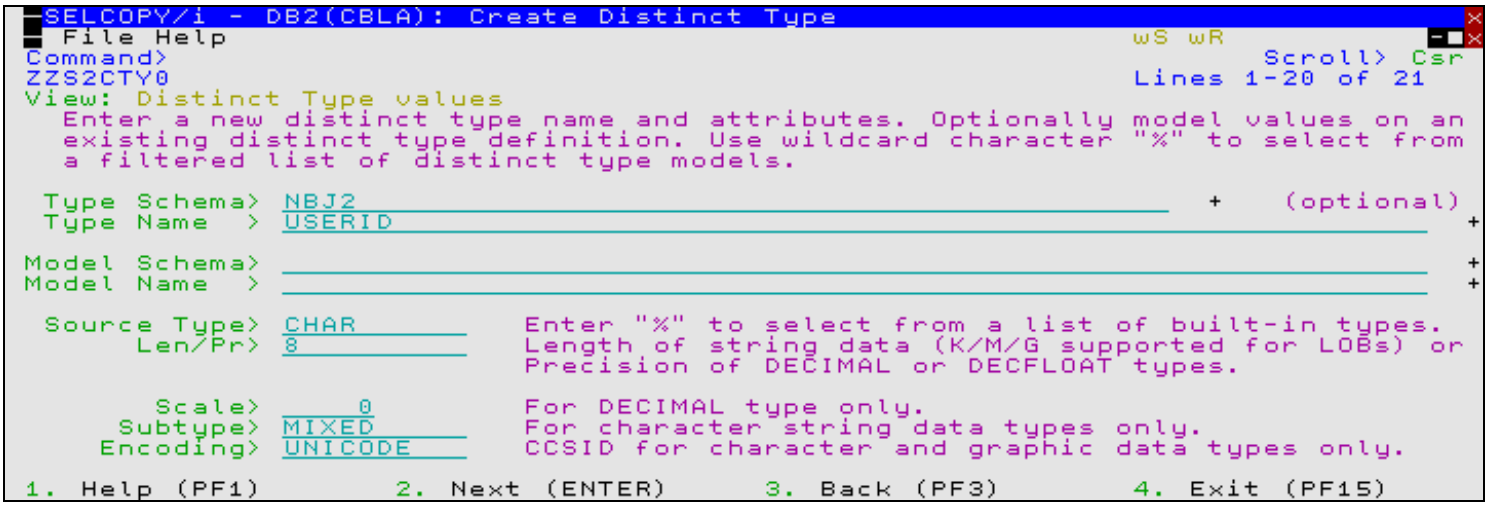


Figure 257. DB2: Create Distinct Type.

Menu Bar Items

The following menu bar items are displayed in the Create Distinct Type panel views.

File
The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
Display help for this panel view.

Distinct Type Values - Panel Fields

Type Schema>
The schema of the new DB2 distinct type.
The distinct type schema has a maximum length of 128 characters.

Type Name>
The name of the new DB2 distinct type. The distinct type name must not match the name of an existing distinct type, built-in type, BOOLEAN or a system reserved keyword.
The distinct type name has a maximum length of 128 characters.

This parameter field corresponds to SQL CREATE TYPE parameter *distinct-type-name*.

Model Schema>
The schema of an existing DB2 distinct type from which attributes of the new distinct type will be modelled.
See Model Name for use of wildcard characters in this field value.

Model Name>
The name of an existing DB2 distinct type from which attributes of the new distinct type will be modelled.

The model distinct type schema and name field values may include wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the model schema/name specification. If this is the case, a Select Distinct Type panel is displayed containing a list of eligible DB2 distinct types that satisfy the model schema/name filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

Having entered a model type, the new distinct type attribute fields will be automatically updated.

Source Type>
The DB2 source data type.

Enter blanks or any invalid value to select from a list of valid data types.

Len/Pr>
For character, graphic character and binary source data types, this field defines the length of the data within the column. For LOB data types, the length value may be suffixed with K, M or G representing a value wich is a multiple of 1024, 1048576 and 1073741824 respectively.

For DECIMAL and DECFLOAT source data types, this field defines the precision (number of decimal digits) represented by the decimal value. For DECFLOAT, the precision value will be rounded up to 16 or 34. If a value greater than 34 is entered error ZZSP080E is returned.

For all source data types, values entered in this field are ignored.

Scale>
For DECIMAL source data type, this field defines the scale (number of fractional digits) represented by the decimal value. This value must be less than or equal to the precision value. For all other source data types, this field is ignored.

Subtype>
For character source data types CHAR, VARCHAR and CLOB, this field specifies the character data subtype (MIXED, SBCS or BIT). For all other source data types, this field is ignored.

For source data types CLOB, selection of subtype BIT is invalid and will return error ZZSP101E.

Enter a blank or invalid value in this field to display a complete list of selectable subtype entries for this field.

Encoding>
For character graphic source data types and character source data types of subtype SBCS or MIXED, this field specifies the associated encoding scheme of the new distinct type definition (UNICODE, EBCDIC or ASCII). For all other source data types, this field is ignored.

Enter a blank or invalid value in this field to display a complete list of selectable encoding entries for this field.

Generate SQL

The Generate SQL panel view is the last panel view in the sequence and is common to all SELCOPY/i DB2 Create Objects panel view sequences. For details, please refer to Generate SQL under the description of the DB2 Create Table sequence of panels.

This view allows the user to select how the generated SQL CREATE TYPE statement is to be implemented.

Create Trigger

The Create Trigger sequence of panel views (ZZS2CTY0) generate an SQL CREATE TRIGGER statement to create a new DB2 trigger (and trigger package) in the current DB2 server.

The DB2 Create Trigger panel views are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select Trigger option 13. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.13)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made in a the panel views. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL CREATE TRIGGER syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

Trigger Name & Type

Enter the name of the trigger to be created in the current DB2 sub-system and select options that govern the time at which the triggered event is actioned and the table operation that activates the trigger. The current DB2 subsystem is displayed in the panel window title bar.

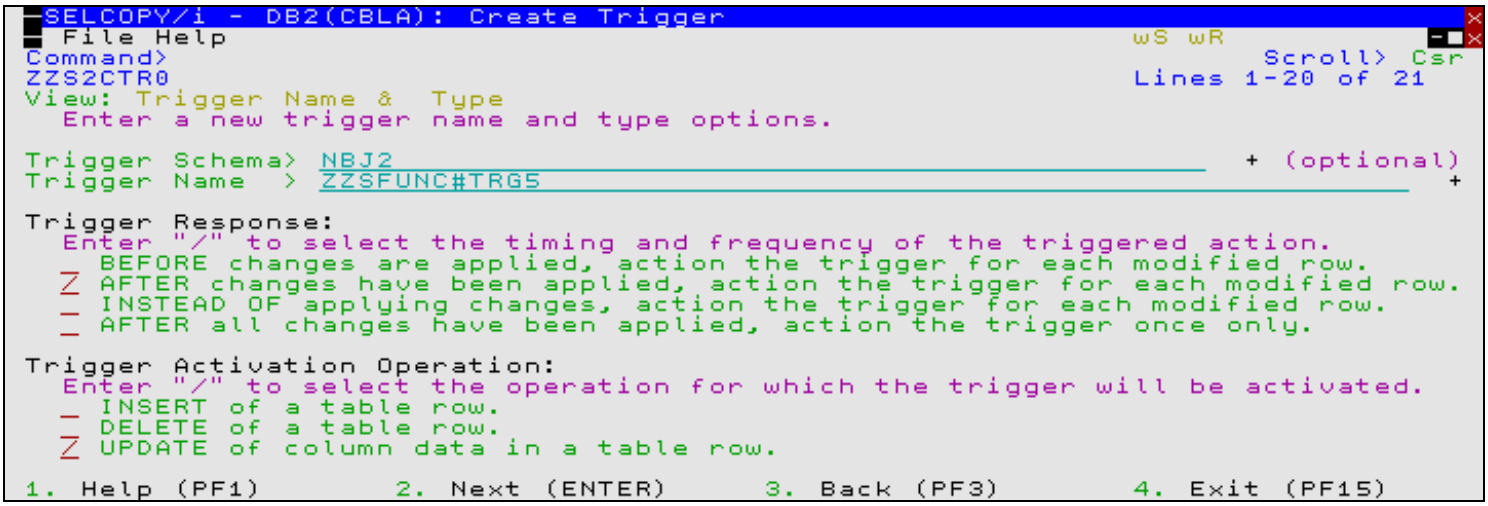


Figure 258. DB2: Create Trigger.

Menu Bar Items

The following menu bar items are displayed in the Create Trigger panel views.

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Trigger Name & Type - Panel Fields

Trigger Schema>
: The schema of the new DB2 trigger.
The trigger schema has a maximum length of 128 characters.

Trigger Name>
: The name of the new DB2 trigger. The trigger schema and name must not match that of an existing DB2 trigger or package.
The trigger name has a maximum length of 128 characters.

Trigger Response:
Enter "/" to select the time at which the triggered action is to be performed.

◊ **BEFORE changes are applied, action the trigger for each modified row.**
For every DB2 table row modified by an execution of the SQL insert, delete or update operation on which the trigger is defined, DB2 executes the triggered action **before** applying the modification to each row.
This option may not be selected if the subject of the trigger is a DB2 view and corresponds to SQL CREATE TRIGGER parameters NO CASCADE BEFORE... FOR EACH ROW.

◊ **AFTER changes have been applied, action the trigger for each modified row.**
For every DB2 table row modified by an execution of the SQL insert, delete or update operation on which the trigger is defined, DB2 executes the triggered action **after** applying the modification to each row.
This option may not be selected if the subject of the trigger is a DB2 view and corresponds to SQL CREATE TRIGGER parameters AFTER... FOR EACH ROW.

◊ **INSTEAD OF applying changes, action the trigger for each modified row.**
For every row of a DB2 view that would be modified by an execution of the SQL insert, delete or update operation on which the trigger is defined, DB2 executes the triggered action **instead of** applying the modification to each row.
This option may not be selected if the subject of the trigger is a DB2 table and corresponds to SQL CREATE TRIGGER parameters INSTEAD OF... FOR EACH ROW.

◊ **AFTER all changes have been applied, action the trigger once only.**
Regardless of whether DB2 table rows are modified by an execution of the SQL insert, delete or update operation on which the trigger is defined, DB2 executes the triggered action once only **after** execution of the SQL operation.
This option may not be selected if the subject of the trigger is a DB2 view and corresponds to SQL CREATE TRIGGER parameters AFTER... FOR EACH STATEMENT.

Trigger Activation Operation:
Enter "/" to select the SQL operation that will action the trigger.

◊ **Insert**
Action is triggered when an SQL INSERT operation is performed on the subject DB2 table or view.
This option corresponds to SQL CREATE TRIGGER parameter INSERT.

◊ **Delete**
Action is triggered when an SQL DELETE operation is performed on the subject DB2 table or view.
This option corresponds to SQL CREATE TRIGGER parameter DELETE.

◊ **Update**
Action is triggered when an SQL UPDATE operation is performed on the subject DB2 table or view.
This option corresponds to SQL CREATE TRIGGER parameter UPDATE.

Trigger Action

The Trigger Action panel view is the next view in the sequence, displayed following the Trigger Name & Type panel view.

This panel view's title and input fields reflect the options selected for Trigger Response and Trigger Activation Operation in the previous panel view. The panel view title may be one of the following:

- BEFORE INSERT of each row Trigger
- BEFORE DELETE of each row Trigger
- BEFORE UPDATE of each row Trigger
- AFTER INSERT of each row Trigger
- AFTER DELETE of each row Trigger
- AFTER UPDATE of each row Trigger
- AFTER INSERT once only Trigger
- AFTER DELETE once only Trigger
- AFTER UPDATE once only Trigger
- INSTEAD OF INSERT of each row Trigger
- INSTEAD OF DELETE of each row Trigger
- INSTEAD OF UPDATE of each row Trigger

Enter the name of the DB2 table or view on which the trigger is defined. A search condition on which execution of the triggered SQL statement(s) will depend and also correlation names and table names for the transition variables and tables may optionally be specified. Transition elements may be referenced within the search condition and/or triggered SQL statement(s).

If, on progressing to the next panel view, no triggered SQL statement has already been defined and selected, the Triggered SQL Statements sub-panel is automatically opened.

Once a triggered SQL statement has been defined and selected, thereafter, the **Triggered SQL Statements** sub-panel may be opened from this panel view using primary command, **TRIggeraction** (assigned to <F6> by default).

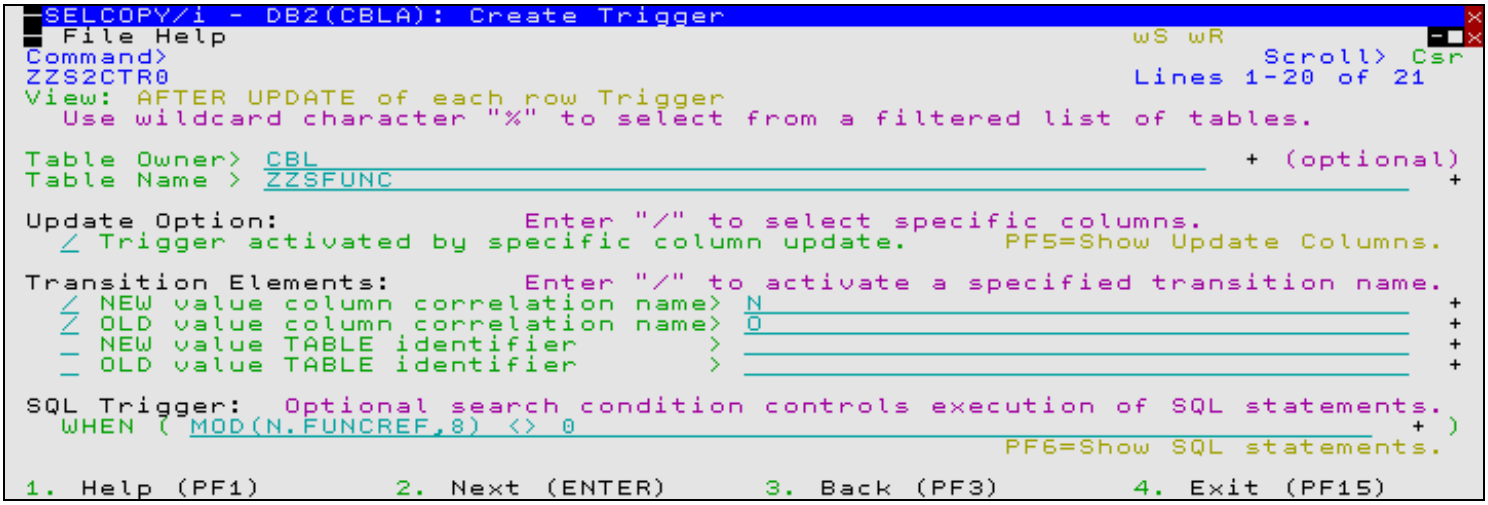


Figure 259. DB2: Create Trigger - AFTER UPDATE of each row Trigger.

Trigger Action - Panel Fields

`Table/View Owner>`
The owner id of the DB2 table or view on which the trigger will be defined.
The table/view owner id has a maximum length of 128 characters.

`Trigger Name>`
The name of the DB2 table or view on which the trigger will be defined.
The table/view name has a maximum length of 128 characters.

`Update Option:`
Applicable only to a trigger for which the triggering action is an SQL UPDATE operation. Enter "/" to indicate that the triggering action is to be limited to update of specific columns in the DB2 table or view.

If this option is selected and trigger update columns have not yet been defined and selected, then the Trigger Update Columns sub-panel is automatically opened.

Once a trigger update column has been defined and selected, thereafter, **Trigger Update Columns** sub-panel may be opened from this panel view using primary command, **SELect** (assigned to <F5> by default). If SELECT is actioned before trigger update columns have been entered, then the "Trigger activated by specific column update" option is automatically selected.

`Transition Elements:`
Enter "/" to select the transition elements (variables and/or tables) for which reference names will be defined. Any defined transition element name may be referenced in the SQL trigger search condition and/or trigger statement(s).

The transition elements are as follow:

`NEW value column correlation name>`
Applicable only to a trigger for which the triggering action is an SQL INSERT or UPDATE operation and for which the trigger is executed once for each modified row.

This option indicates that a correlation name is to be defined for transition variables representing the values in the row as modified by the triggering SQL operation and by any SET statement in a before trigger that has already been executed.

The input field that follows specifies the correlation name to be defined. This corresponds to SQL CREATE TRIGGER parameters NEW AS *correlation-name*.

`OLD value column correlation name>`
Applicable only to a trigger for which the triggering action is an SQL DELETE or UPDATE operation and for which the trigger is executed once for each modified row.

This option indicates that a correlation name is to be defined for transition variables representing the values in the row prior to being modified by the triggering SQL operation.

The input field that follows specifies the correlation name to be defined. This corresponds to SQL CREATE TRIGGER parameters OLD AS *correlation-name*.

`NEW value TABLE identifier>`
Applicable only to a trigger for which the triggering action is an SQL INSERT or UPDATE operation and for which the trigger is **not** executed **before** the SQL operation.

This option indicates that a table name identifier is to be defined for transition tables representing values in the complete set of affected rows as modified by the triggering SQL operation.

The input field that follows specifies the table identifier to be defined. This corresponds to SQL CREATE TRIGGER parameters NEW_TABLE AS *table-identifier*.

OLD value TABLE identifier>
Applicable only to a trigger for which the triggering action is an SQL DELETE or UPDATE operation and for which the trigger is **not** executed **before** the SQL operation.

This option indicates that a table name identifier is to be defined for transition tables representing values in the complete set of affected rows prior to being modified by the triggering SQL operation.

The input field that follows specifies the table identifier to be defined. This corresponds to SQL CREATE TRIGGER parameters OLD_TABLE AS *table-identifier*.

SQL Trigger: WHEN
Applicable only to a trigger that does **not** have a Trigger Response of INSTEAD OF.

This input field optionally specifies a DB2 search condition that must be true in order for the triggered SQL statement to be executed. If omitted, the SQL statement will be executed unconditionally.

The Triggered SQL Statements sub-panel, in which the SQL statements are defined, is opened automatically on progressing to the next panel view. Alternatively, the sub-panel may be opened from the Trigger Action panel view using primary command, **TRIggeraction** (assigned to <F6> by default).

This corresponds to the SQL CREATE TRIGGER parameter WHEN (*search-condition*).

Trigger Update Columns

The Trigger Update Columns sub-panel is displayed from the **Trigger Action** panel view on selecting Update Option: Trigger activated by specific column update or entering primary command, **SELect** (assigned to <F5> by default).

This sub-panel contains an embedded table of eligible column names that belong to the selected DB2 table. Table rows may not be deleted, copied or moved but the table display may be scrolled UP, DOWN, LEFT and RIGHT.

This sub-panel also supports primary commands **SELECTAll** and **REFRESH**, to toggle selection and deselection of all columns and reset display of column definitions. (By default, SELECTALL is assigned to <F5>, SELECTALL DESELECT is assigned to <F6> and REFRESH is assigned to <F16>.)

Each selected table row identifies a DB2 table column to be included in the UPDATE trigger definition. Only UPDATE operations performed on one or more of these columns will action the trigger.

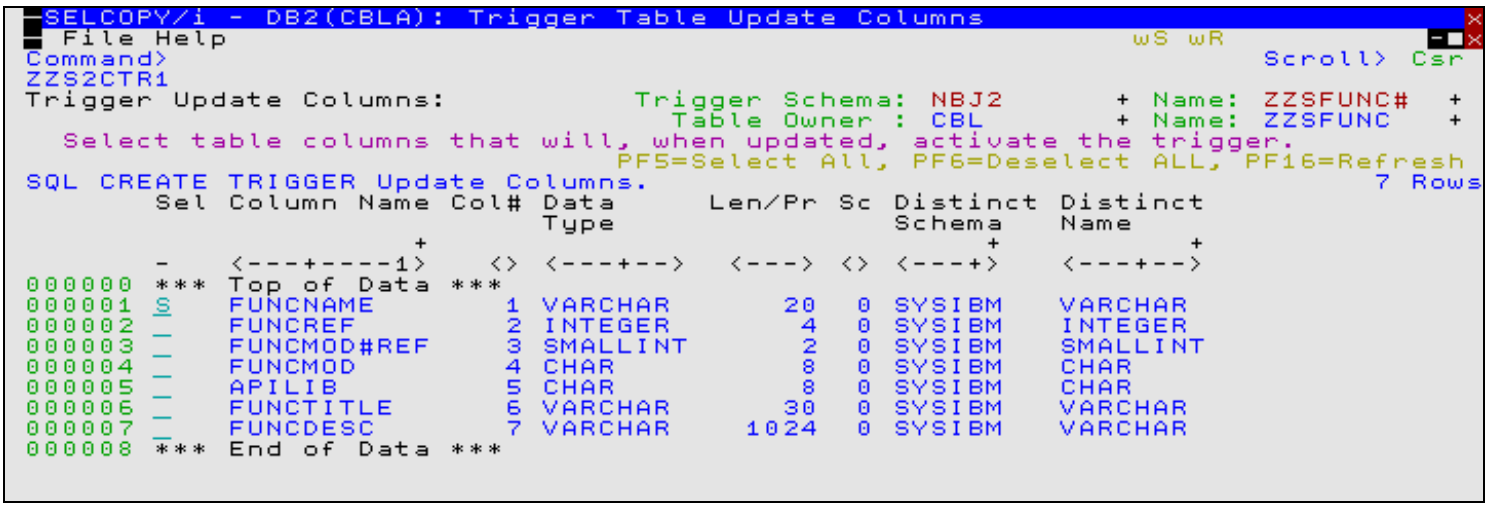


Figure 260. DB2: Create Trigger - Trigger Update Columns.

Trigger Update Columns - Panel Fields

Table Owner:
A non-enterable field displaying the owner id of the DB2 table on which the trigger is to be defined.

Table Name:
A non-enterable field displaying the name of the DB2 table on which the trigger is to be defined.

Trigger Schema:
A non-enterable field displaying the schema of the trigger.

Trigger Name:
A non-enterable field displaying the name of the trigger.

Sel
Enter "/" (slash) or any non-blank character in this field to include (select) the column definition in the trigger update column list. A blank in this field will deselect the column.

Column Name
A non-enterable field displaying a column name belonging to the selected DB2 table.

Col#
A non-enterable field displaying the column sequence number within the selected DB2 table.

Data Type
A non-enterable field displaying the column data type.

Len/Pr
A non-enterable field displaying the length or precision assigned to the column data type.

Sc
A non-enterable field displaying the scale value assigned to the column data type.

Distinct Schema
A non-enterable field which displays the source type schema of the column data type. This is relevant where the column is assigned a defined distinct type.

Distinct Name
A non-enterable field which displays the source type name of the column data type. This is relevant where the column is assigned a defined distinct type.

Triggered SQL Statements

The Triggered SQL Statements sub-panel is displayed on progressing from the Trigger Action panel view via one of the following methods:

1. Automatically when when a triggered SQL action has not yet been defined and selected.
2. Enter primary command **TRIGgeraction** (default for <F6>).

This sub-panel contains an embedded table of DB2 SQL statements to be executed when the trigger is actioned. The order of the table rows determines the sequence order in which the SQL statements will be executed.

Standard table edit primary and line commands must be used to INSERT, DELETE, COPY, MOVE or EXCLUDE table rows as appropriate, to scroll the table display UP, DOWN, LEFT and RIGHT and also to ZOOM the display of an individual table row.

In addition to the standard primary table commands, this sub-panel also supports primary command **SELECTAll** to toggle selection and deselection of all columns and expressions. (By default, SELECTALL is assigned to <F5> and SELECTALL DESELECT is assigned to <F6>.)

If the width of the Triggered SQL statment field in the table view is not sufficient to enter the required input value, then the table row should be zoomed (default <F17>) and, if necessary, the field expanded (default <F14>) to accomodate the input value. Note that the zoomed table row view also displays a useful table of the types of triggered SQL statement that may be executed for the various trigger response options (BEFORE, AFTER or INSTEAD OF).

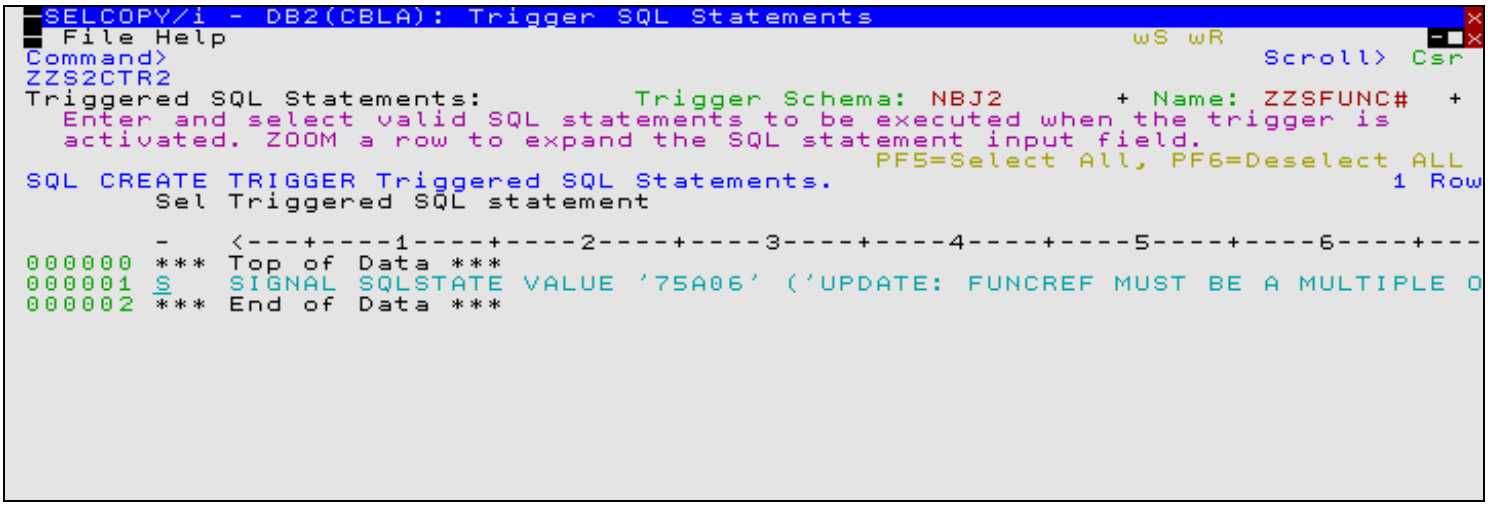


Figure 261. DB2: Create Trigger - Triggered SQL Statements (table view).

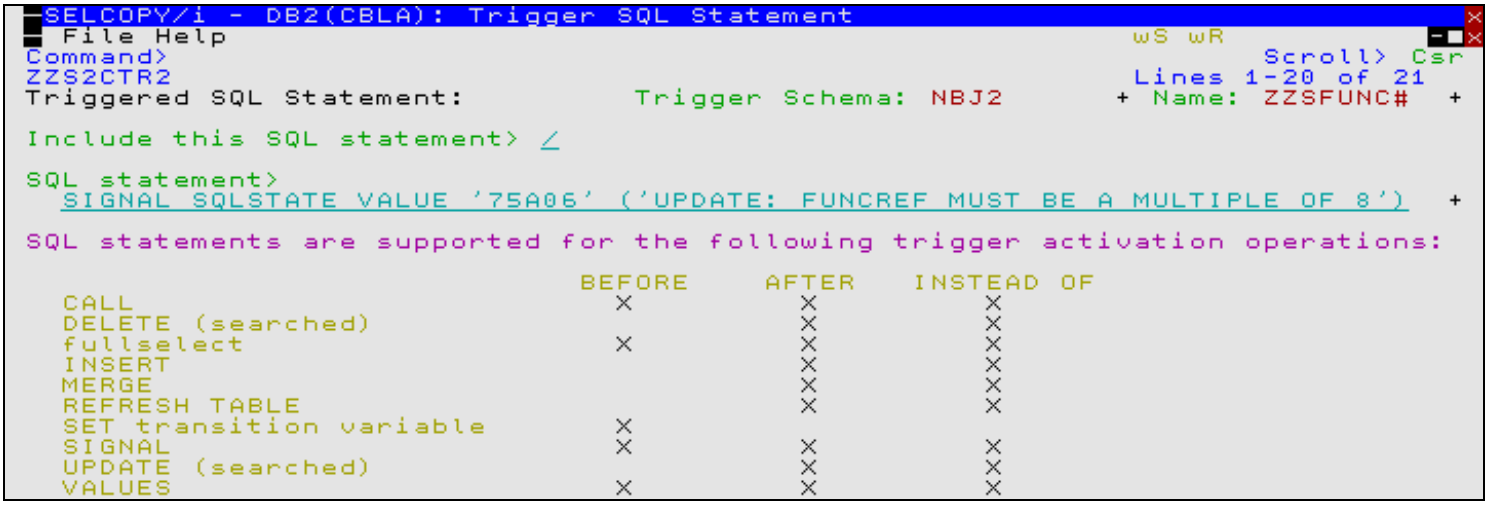


Figure 262. DB2: Create Trigger - Triggered SQL Statements (zoomed view).

Triggered SQL Statements - Panel Fields

Field names that follow are as appear in the zoomed view of a table row. Names in parentheses correspond to the equivalent column name in table view.

`Trigger Schema:`
A non-enterable field displaying the schema of the trigger.

`Trigger Name:`
A non-enterable field displaying the name of the trigger.

`Include this SQL statement> (Sel)`
Enter "/" (slash) or any non-blank character in this field to include (select) the column definition in the trigger update column list. A blank in this field will deselect the column.

Deselecting a column definition in table view has the same effect as executing the table primary command EXCLUDE (or line command "X") to exclude the table row.

`SQL statement> (Triggered SQL statement)`
Specifies a triggered SQL statement. Note that the SQL statement should not be terminated with a ';' (semi-colon) as one will be generated automatically if necessary.

In zoomed view, a table of valid triggered SQL statements is displayed. For further details on valid SQL statements, see CREATE TRIGGER documentation in the *"IBM DB2 SQL Reference"*.

This corresponds to the SQL CREATE TRIGGER parameter *triggered-SQL-statement* or BEGIN ATOMIC *triggered-SQL-statement*; ... END.

Generate SQL

The Generate SQL panel view is the last panel view in the sequence and is common to all SELCOPY/i DB2 Create Objects panel view sequences. For details, please refer to Generate SQL under the description of the DB2 Create Table sequence of panels.

This view allows the user to select how the generated SQL CREATE TRIGGER statement is to be implemented.

Primary Commands

The following primary commands are supported by selected views in the DB2 Create Trigger sequence of panel views. If issued from a panel view in which the command is not valid, the message "ZZSP102E Primary command not valid in the current context" is displayed.

CMX

```
>>--+- CMX ----------------------+-----------------------------------><
    +- EDITCMX -------------------+
```

Applicable only from the Generate SQL view, CMX generates the SQL statement and copies it to an in-storage output file in a format suitable for subsequent execution using the ACTION (default >F16>) facility.
CMX is assigned to <F17> by default.

JCL

```
>>--+- JCL ----------------------+-------------------------------------><
    +- EDITJCL ------------------+
```

Applicable only from the Generate SQL view, JCL generates the SQL statement and copies it to an in-storage output file with JCL statements that execute the DSNTIAD facility. This job may be submitted to batch using the SELCOPY/i text editor SUBMIT primary command.

JCL is assigned to <F18> by default.

RUN

```
>>--+- RUN ----------------------+-------------------------------------><
    +- EXECSYNtax ---------------+
```

Applicable only from the Generate SQL view, RUN verifies input fields in all applicable panel views and then executes the generated SQL statement. This is the default action on pressing <Enter> from the last panel view in the sequence (i.e. the **Generate SQL** view.)

SELECT

```
>>---- SELect ------------------------------------------------------><
```

SELECT will open a sub-panel that is applicable to the current panel view. Each of the panel views listed below identify the sub-panel opened on executing SELECT. In all other panel views, SELECT is invalid.

- **Trigger Action**
  For a trigger action based on an UPDATE to specific DB2 table columns, SELECT opens the Trigger Update Columns sub-panel, used to specify the FOR UPDATE OF trigger column names.

SELECT is assigned to <F5> by default.

TRIGGERACTION

```
>>---- TRIGgeraction ----------------------------------------------><
```

From any of the Trigger Action panel views, TRIGGERACTION will open the Triggered SQL Statements sub-panel, used to define one or more SQL statements that are to be executed when the trigger is actioned.

TRIGGERACTION is assigned to <F6> by default.

Create Sequence

The Create Sequence sequence of panel views (ZZS2CSQ0) generate an SQL CREATE SEQUENCE statement to create a new DB2 sequence at the current DB2 server.

The DB2 Create Sequence panel views are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select Sequence option 14. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.14)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made in a the panel views. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL CREATE SEQUENCE syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

Sequence Values

Enter the name of the sequence to be created in the current DB2 sub-system, its numeric data type and other DB2 sequence attribute values. The data type and attribute values may be modelled on an existing DB2 sequence definition. The current DB2 subsystem is displayed in the panel window title bar.

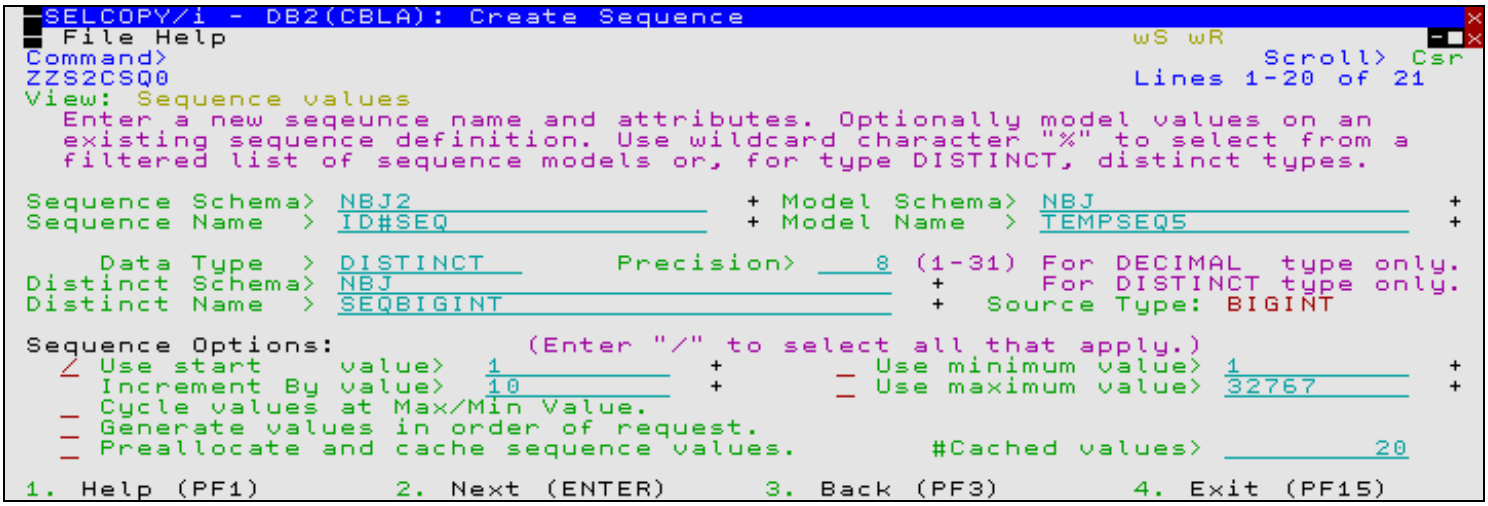


Figure 263. DB2: Create Sequence.

Menu Bar Items

The following menu bar items are displayed in the Create Sequence panel views.

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Sequence Values - Panel Fields

Sequence Schema>
: The schema of the new DB2 sequence.
The sequence name has a maximum length of 128 characters.

Sequence Name>
: The name of the new DB2 sequence. The sequence name must not match the name of an existing sequence at the current DB2 server.
The sequence name has a maximum length of 128 characters.

This parameter field corresponds to SQL CREATE SEQUENCE parameter *sequence-name*.

Model Schema>
The schema of an existing DB2 sequence from which attributes of the new sequence will be modelled.

See Model Name for use of wildcard characters in this field value.

Model Name>
The name of an existing DB2 sequence from which attributes of the new sequence will be modelled.

The model sequence schema and name field values may include wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the model schema/name specification. If this is the case, a Select Sequence panel is displayed containing a list of eligible DB2 sequences that satisfy the model schema/name filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

Having entered a model type, the new sequence data type and sequence option attribute fields will be automatically updated.

Data Type>
The DB2 data type of the numeric sequence values.

Enter blanks or any invalid value to select from a list of valid data types.

Precision>
For DECIMAL data type only, this field defines the precision (number of decimal digits) for values in the sequence. Note that only sequences of integer values are supported by DB2 and so DECIMAL scale must be 0 (zero).

For all other data types, values entered in this field are ignored.

Distinct Schema
Applicable only if the selected data type is DISTINCT, this field specifies the schema of the distinct type.

See Distinct Name for use of wildcard characters in this field value.

Distinct Name
Applicable only if the selected data type is DISTINCT, this field specifies the name of the distinct type.

The distinct type schema and name field values may include wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the distinct type schema/name specification. If this is the case, a Select Distinct Type panel is displayed containing a list of eligible DB2 distinct types that satisfy the distinct schema/name filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

Having entered a distinct type, the Precision and Source Type fields will be automatically updated.

Source Type:
Applicable only if the selected data type is DISTINCT, this is a non-enterable field displaying the DB2 source data type of the selected distinct type.

Sequence Options:
Enter "/" to select options or activate DB2 sequence attribute input fields.

Use start value>
This option indicates that a start value is to be defined for the DB2 sequence. If this option is selected, the input field that follows specifies the first positive or negative value in the sequence.

If not selected, the start value is the MINVALUE for ascending sequences or the MAXVALUE for descending sequences.

This corresponds to SQL CREATE SEQUENCE parameters START WITH *numeric-constant*.

Increment By value>
This input field specifies the increment value to added to the current value to obtain the next value in the sequence. A positive value or 0 (zero) indicates an ascending sequence, a negative value indicates a descending sequence.

This corresponds to SQL CREATE SEQUENCE parameters INCREMENT BY *numeric-constant*.

Use minimum value>
This option indicates that a minimum value is to be defined for the DB2 sequence. If this option is selected, the input field that follows specifies the minimum value.

If not selected, no minimum value is set and the default minimum value for a descending sequence is the minimum value for the sequence data type.

This corresponds to SQL CREATE SEQUENCE parameters NO MINVALUE and MINVALUE *numeric-constant*.

Use maximum value>
This option indicates that a maximum value is to be defined for the DB2 sequence. If this option is selected, the input field that follows specifies the maximum value.

If not selected, no maximum value is set and the default maximum value for an ascending sequence is the maximum value for the sequence data type.

This corresponds to SQL CREATE SEQUENCE parameters NO MAXVALUE and MAXVALUE *numeric-constant*.

Cycle values at Min/Max value
This option indicates that when an ascending sequence reaches the maximum value the sequence will cycle to the minimum value or when a descending sequence reaches the minimum value the sequence will cycle to the maximum value.

If not selected, no further sequence values will be generated when the minimum/maximum threshold limit is reached.

This corresponds to SQL CREATE SEQUENCE parameters NO CYCLE and CYCLE.

Generate values in order of request
This option indicates that sequence numbers are generated in order of request.

This corresponds to SQL CREATE SEQUENCE parameters NO ORDER and ORDER.

Preallocate and cache sequence values and **#Cached values>**
This option indicates that a number of sequence numbers may be preallocated and held in a cache for faster processing. If this option is selected, the #Cached values input field specifies the maximum number of values of the sequence that DB2 can preallocate and keep in memory. The minimum number of cached values is 2.

This corresponds to SQL CREATE SEQUENCE parameters NO CACHE and CACHE *numeric-constant*.

Generate SQL

The Generate SQL panel view is the last panel view in the sequence and is common to all SELCOPY/i DB2 Create Objects panel view sequences. For details, please refer to Generate SQL under the description of the DB2 Create Table sequence of panels.

This view allows the user to select how the generated SQL CREATE SEQUENCE statement is to be implemented.

Create Role

The Create Role sequence of panel views (ZZS2CRO0) generate an SQL CREATE ROLE statement to create a new DB2 role at the current DB2 server.

The DB2 Create Role panel views are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select Role option 15. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.15)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a role based on choices made in a the panel views. Field values and options may be selected in the focus panel before progressing to the next panel view in the role (NEXT). Progressing forwards from the last panel view in the role will generate the SQL CREATE ROLE syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view role (BACK) before proceding forwards again.

Role Values

Enter the name of the role to be created in the current DB2 sub-system. The current DB2 subsystem is displayed in the panel window title bar.

Menu Bar Items

The following menu bar items are displayed in the Create Role panel views.

`File`
: The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

`Help`
: Display help for this panel view.

Role Values - Panel Fields

`Role Name>`
: The name of the new DB2 role. The role name must not match the name of an existing role at the current DB2 server. The role name has a maximum length of 128 characters.

  This parameter field corresponds to SQL CREATE ROLE parameter *role-name*.

Generate SQL

The Generate SQL panel view is the last panel view in the role and is common to all SELCOPY/i DB2 Create Objects panel view roles. For details, please refer to Generate SQL under the description of the DB2 Create Table role of panels.

This view allows the user to select how the generated SQL CREATE ROLE statement is to be implemented.

Create Clone Table

The DB2 **Create Clone Table** sequence of panel views (ZZS2CCT0) generate an SQL ALTER TABLE ADD CLONE statement to create a clone table for the selected table within the current DB2 subsystem.

The DB2 Create Clone Table panel views and their sub-panels are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select Base Table Clone option 16. in the SELCOPY/i DB2 Create Objects option menu. (DB2 5.16)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made in a the panel views. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL ALTER TABLE ADD CLONE syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

Table Names

Enter the name of the clone table and the DB2 base table in the current DB2 server from which it will be created. The current DB2 subsystem is displayed in the panel window title bar.

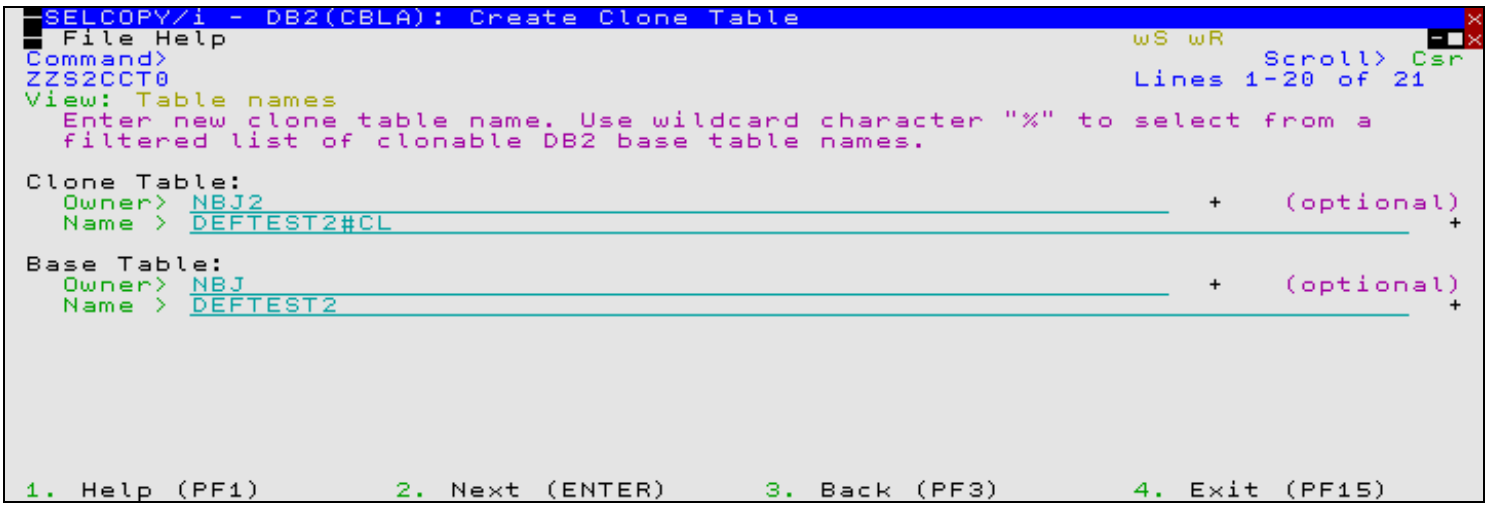


Figure 264. DB2: Create Clone Table.

Menu Bar Items

The following menu bar items are displayed in the Create Clone Table panel views.

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Table Names - Panel Fields

Clone Table Owner>
: Optionally specifies the owner id (schema) of the clone table to be created. If this field is left blank, DB2 will assign an owner id equal to the value of the user's current SQLID special register.

  A clone table owner id has a maximum length of 128 characters.

Clone Table Name>
: Mandatory field which specifies the name of the clone table to be created.

  A clone table name has a maximum length of 128 characters.

Base Table Owner>
Specifies the owner id (schema) of the base table to be cloned.
This field will be populated on specification of a DB2 table. See Base Table Name for details on table selection.

A base table owner id has a maximum length of 128 characters.

Base Table Name>
Mandatory field which specifies the name of the base table to be cloned.

Wildcard character "%" (percent) or "*" (asterisk), representing zero or more characters, or wildcard character "_" (underscore), representing a single character, in one of the Base Table Owner and Base Table Name fields will open a Select Table panel. This panel will contain a list of base tables that satisfy the criteria required for table cloning and also satisfy the filter derived from the 2 input fields. Use prefix command "S", or press the <Enter> key on the required entry to select it and so populate the 2 Base Table input fields.

A base table name has a maximum length of 128 characters.

Generate SQL

The Generate SQL panel view is the last panel view in the sequence and is common to all SELCOPY/i DB2 Create Objects panel view sequences. For details, please refer to Generate SQL under the description of the DB2 Create Table sequence of panels.

This view allows the user to select how the generated SQL ALTER TABLE statement is to be implemented.

Alter User Table Space

The DB2 **Alter Table Space** sequence of panel views (ZZS2ATS0) are almost identical to those for DB2 Create User Table Space The panel sequence will ultimately generate an SQL ALTER TABLESPACE statement to alter attributes of a table space which exists within a database of the current DB2 subsystem.

The DB2 Alter Table Space panel views and their sub-panels are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Enter line-command "A" against any entry in a Tablespaces list window.

By default, field entries are populated with arguments and options based on attributes that are currently assigned to the specified table space.

The panel views are displayed in a sequence based on choices made in a the panel views. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL ALTER TABLESPACE syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

Table Space Name

Enter the name of an existing table space and, optionally, the name of its database in the current DB2 server in which the table space exists. The current DB2 subsystem is displayed in the panel window title bar.

Menu Bar Items

The following menu bar items are common to all Alter Table Space panel views and sub-panels.

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem. Note that, unlike CANCEL, CLOSE will save field values that have been entered in the panel views so far. These values may be redisplayed the next time the panel is opened.

Help
: Display help for this panel view.

Table Space Name & Type - Panel Fields

Table Space Name>
: Mandatory field which specifies the name of the table space to be altered.

  The table space name may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the table space name. If this is the case, a Select Table Space panel is displayed containing a list of table spaces which satisfy the name filter.
  Use prefix command "S", or press the <Enter> key on the required entry to select it. Having selected a table space name, the values assigned to the panel fields will be updated to reflect the current (unaltered) values.

In DataBase>
: Optional field which specifies the name of the user database in which the table space will be altered.

  The database name may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the database name. If this is the case, a Select Database panel is displayed containing a list of eligible database names which satisfy the database name filter.
  Use prefix command "S", or press the <Enter> key on the required entry to select it. Having selected a database, if no modelling has occurred, then table space buffer pool name, DB2 storage group and CCSID encoding scheme defaults will be updated to values defined for the database.

  The default database is DSNDB04.

TSType:
: A non-enterable field which identifies the type of table space selected. Possible table space type descriptions are as follow:

  ◊ **Segmented**
  The selected table space is an exclusively segmented (non-partitioned) table space.

  ◊ **Partitioned.**
  The selected table space is an exclusively partitioned (non-segmented) table space.

◊ **Universal Partition-by-Growth.**
The selected table space is a universal (partioned and segmented) partitioned-by-growth table space. Partitions added as the table space grows.

◊ **Universal Partition-by-Range.**
The selected table space is a universal (partitioned and segmented) partitioned-by-range table space. All partitions defined and data clustered based on partitioning key column values.

◊ **LOB Table Space**
The selected table space is a LOB table space. LOB table spaces must exist in the same database as the tablespace in which the LOB column is defined.

Table Space Type Attributes

The table space attributes panel view is the next view in the sequence, displayed following the Table Space Name & Type panel view. The contents of the panel view include only those fields that are applicable to the selected table space type.

The panel view header displays one of the following types:

- `Segmented Table Space`
- `Partitioned Table Space`
- `Universal Part-by-Growth`
- `Universal Part-by-Range`
- `LOB Table Space`

Table Space Type Attributes - Panel Fields

Note that of the panel fields described below, only those that are applicable to the selected table space type will be displayed.

`TableSpace:`
A non-enterable field displaying the new table space name as supplied in the **Table Name & Type** panel view.

`In DataBase:`
A non-enterable field displaying the database in which the new table space will be altered, as supplied in the **Table Name & Location** panel view.

`Data set Size:`
Not applicable to exclusively Segmented table space type.

An unalterable field, provided for informational purposes only. This value specifies the maximum size in gigabytes of each table space partition or LOB table space data set. For sizes greater 4G, the data sets must be associated with an SMS DATACLAS that has been defined with extended format and extended addressability.

Enter blanks or any invalid value to select from a list of valid data set sizes.

`Table Space Data sets:`
Enter "/" to select the method by which table space data set will be managed and enter values appropriate to the selected method. Options are as follow:

◊ **DB2 managed**
Specifies that DB2 will define and manage the data sets for the table space. Each data set will be defined on a volume of the identified storage group with primary and secondary allocations sizes defined by the Primary and Secondary Alloc fields respectively.

◊ **User managed**
Specifies that table space data sets are to be managed by the user. Table space data sets are linear VSAM data sets cataloged in an ICF catalog identified by the catalog name/alias defined by the Catalog Alias field.
Note that this option is not available for Universal Partition-by-Growth table space type.

`Storage Group>`
Mandatory field for DB2 managed data sets which specifies the name of a storage group defined in the current DB2 server.

The storage group name may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the storage group name. If this is the case, a Select Storage Group panel is displayed containing a list of eligible storage groups which satisfy the name filter. Use prefix command "S", or press the <Enter> key on the required entry to select it.

`Primary Alloc>`
Mandatory field for DB2 managed data sets which specifies the minimum primary allocation size in KB for the table space data set.

A value of -1 indicates to DB2 that it should use a value based on system defaults. A value other than -1 may be adjusted by DB2 to satisfy minimum requirements for the associated buffer pool page size.

Secondary Alloc>
Mandatory field for DB2 managed data sets which specifies the minimum secondary allocation size in KB for the table space data set.

A value of -1 indicates to DB2 that it should use a value based on system defaults.

Action on DROP>
Enter "/" to select whether or not the table space data set or partition data set will be erased when the table space is deleted on execution of a utility or dropped using an SQL statement.

Catalog Alias>
Mandatory field for User managed data sets which specifies the name or alias of the ICF catalog in which the table space data sets are to be cataloged.

The catalog alias may include a wildcard character "%" (percent) or "*" (asterisk) representing zero or more characters, or wildcard character "_" (underscore) representing a single character of the storage group name. If this is the case, a Select Catalog Alias panel is displayed containing a list of eligible aliases that satisfy the model catalog alias filter.
Use prefix command "S", or press the <Enter> key on the required entry to select it.

Buffer Pool>
Specifies the 4K, 8K, 16K or 32K buffer pool name to be used for the table space and so determines the table space page size.

This field is usually initialised to be a value modelled on the model table space or the database default value. If a default value could not be determined from either of these sources, then "Default" is selected. "Default" will result in no BUFFERPOOL value being specified in the resultant ALTER TABLESPACE statement.

Enter blanks or any invalid value to select from a list of valid buffer pool names.

Segment Size:
Not applicable to exclusively Partitioned table space type.

An unalterable field, provided for informational purposes only. This value specifies an integer value (4-64) which defines the size of table space segments. The size corresponds to a number of pages to be assigned to each segment.

#Partitions:
Not applicable to exclusively Segmented or Universal Partition-by-Growth table space types.

An unalterable field, provided for informational purposes only. This value specifies an integer value (1-4096) which defines the number of range partitioned table space partitions.

The maximum number of partitions that can be specified depends on the selected buffer pool (page size) and data set size.

Max Partitions>
Applicable only to Universal Partition-by-Growth table space type.

Specifies an integer value (1-4096) which defines the maximum number of table space partitions that may be allocated as the table grows.

The maximum number of partitions that can be specified depends on the selected buffer pool (page size) and data set size. Specifying a number of partitions that exceeds the maximum will return an error. See IBM publication "*DB2 SQL Reference*", "*ALTER TABLESPACE*" for partition number limits.

Partition data set attributes
Not applicable to exclusively Segmented or Universal Partition-by-Growth table space types.

Select this option if attributes are to be altered for individual partition data sets.

Having selected this option, the Table Space Partition Attributes sub-panel will be displayed. This will occur until at least one partition's attributes has been configured and that partition selected from the sub-panel.

Primary command SELECT (assigned to <F5> by default) will also display this sub-panel and select this option field.

Do not use clustering index (Member Cluster)
Applicable only to exclusively Partitioned table space type.

Select this option if data inserted by an insert operation is **not** to be clustered by the implicit or explicit clustering index. DB2 will, instead, locate the data in the table space based on available space (MEMBER CLUSTER).

Table Space Partition Attributes

The Table Space Partition Attributes sub-panel is displayed on executing the SELECT primary command or by selecting option **Partition data set attributes** in the Table Space Type Attributes panel view.

This sub-panel contains an embedded table of DB2 table space partitions and their data set attributes. The table is for update only so that rows may not be manipulated using the standard table editing techniques.

Each table row identifies a partition of the table space. Zoom of an individual row displays a formatted view of the row data.

Table Space Partition Attributes - Panel Fields

TableSpace:
A non-enterable field displaying the new table space name as supplied in the **Table Name & Type** panel view.

In DataBase:
A non-enterable field displaying the database in which the new table space will be altered, as supplied in the **Table Name & Location** panel view.

Partition Number: (Ptn)
A non-enterable field displaying the table space partition number.

Sel> (Sel)
Enter "/" (slash) or any non-blank character in this field to include (select) the attributes for the partition number in the alter table space syntax. A blank in this field will exclude (deselect) the partition attributes.

Deselecting a column definition in table view has the same effect as executing the table primary command EXCLUDE (or line command "X") to exclude the table row.

Table Space data sets: (Using)
A value of STOGROUP or VCAT representing options "DB2 Managed" or "User Managed" respectively.

Options are as described for the Table Space Data sets: field of the Table Space Type Attributes panel view.

Storage Group> (StoGroup/Catalog)
Storage Group value as described for the Storage Group field of the Table Space Type Attributes panel view.

Primary Alloc> (Pri (KB))
Primary allocation value as described for the Primary Alloc field of the Table Space Type Attributes panel view.

Secondary Alloc> (Sec (KB))
Secondary allocation value as described for the Secondary Alloc field of the Table Space Type Attributes panel view.

Action on DROP> (Erase)
Data set erase option as described for the Action on DROP field of the Table Space Type Attributes panel view.

Catalog Alias> (StoGroup/Catalog)
Catalog alias name as described for the Catalog Alias field of the Table Space Type Attributes panel view.

Free page frequency> (Free Page)
Number of pages loaded after which a page of free space will be left. This is as described for the Free page frequency field of the Table Space Options (2/2) panel view.

Percent free/page> (Free Pcnt)
Percentage of space left free per page as described for the Percent free/page field of the Table Space Options (2/2) panel view.

Track Changes> (Track)
Option to track changes to data as described for the Track Modified Pages option field of the Table Space Options (2/2) panel view.

Compress Rows> (Comp)
Option to compress table space rows as described for the Compress Rows option field of the Table Space Options (2/2) panel view.

Group Buffer Pool Caching: (GBP Cache)
For data sharing only, identifies the type of group buffer pool caching as described for the Group Buffer Pool Caching option field of the Table Space Options (1/2) panel view.

Table Space Options (1/2)

The Table Space Options (1/2) panel view is the next view in the sequence, displayed following the Table Space Type Attributes panel view. The contents of the panel view reflect the selected table space type.

Table Options (1/2) - Panel Fields

TableSpace:
A non-enterable field displaying the new table space name as supplied in the **Table Name & Type** panel view.

In DataBase:
A non-enterable field displaying the database in which the new table space will be altered, as supplied in the **Table Name & Location** panel view.

Logged:
Select this option to record in the log all changes to the table and index data in the table space.

Close Priority:
Enter "/" to select the priority in which the table data sets belonging to the table space are closed when the limit of open table space data sets is reached.

Close first (CLOSE YES) indicates that the data sets will be eligible to be closed before data sets belonging to a table space altered with CLOSE NO.

Lock Size:
Enter "/" to select the size of locks used in the table space and so the threshold at which lock escalation occurs. Options are as follow:

- ◊ **Any size selected by DB2**
- ◊ **Tablespace**
- ◊ **Table** - Applicable only to exclusively Segmented table space type.
- ◊ **Page** - Not applicable to LOB table spaces.
- ◊ **Row** - Not applicable to LOB table spaces.
- ◊ **LOB** - Applicable only to LOB table spaces.

Lock Maximum:
Enter "/" to select whether the maximum number of locks before escalation is to be determined by the user or by the DB2 system.

If user defined, the threshold value (0-2147483647) may be specified in the accompanying field. A value of 0 (zero) indicates that locks are not to be counted and so no escalation occurs.

Group Buffer Pool Caching:
Applicable only in a data sharing environment, enter "/" to select what pages of the table space or partition are written to the group buffer pool. In a non-data sharing environment, this option will be ignored. Options are as follow:

- ◊ **Changed data pages**
  Unless defined in a group buffer pool that is defined to be used only for cross-invalidation, cache only those table space pages containing table or index data that has been modified.

- ◊ **ALL data pages read**
  Cache all table space pages as they are read from DASD.

- ◊ **Changed system pages**
  Applicable only to LOB table spaces, cache only system pages within the LOB table space that have changed. A system page is a space map page or any other page that does not contain actual data values.

- ◊ **No caching**
  Do not cache table space pages in the group buffer pool.

Table Space Options (2/2)

The Table Space Options (2/2) panel view is the next view in the sequence, displayed following the Table Space Options (1/2) panel view.

Table Options (2/2) - Panel Fields

TableSpace:
A non-enterable field displaying the new table space name as supplied in the **Table Name & Type** panel view.

In DataBase:
A non-enterable field displaying the database in which the new table space will be altered, as supplied in the **Table Name & Location** panel view.

Maximum Rows>
Specifies the maximum number of rows (1-255) that the DB2 system will consider placing on each data page for for insert, LOAD and REORG operations.

Compress Rows>
Enter "/" to select whether or not compression will be performed on rows of the table space or partition. If yes is selected, rows will not be compressed until the LOAD or REORG utility is run on the table in the table space or partition.

Free page frequency>
For a LOAD or REORG operation, specifies the number of pages (0-255) that will be loaded before a leaving a page of free space. The count of pages loaded restarts following the free page. A value of 0 (zero) indicates that no free pages are to be left.

Percent free/page>
For a LOAD or REORG operation, specifies the percentage (0-99) of each page to be left as free space.

CCSID Encoding Scheme:
Enter "/" to select the default encoding scheme (UNICODE, EBCDIC or ASCII) for table stored in the table space. If table space modelling has not occurred, then this field will be initialised to the encoding scheme assigned to the database in which the table space is to be altered, otherwise the system default defined in install panel DSNTIPF.

Track Modified Pages:
Select this option if DB2 is to track modified pages in the space map pages of the table space or partition. Tracking modified pages improves the performance of incremental image copy.

Generate SQL

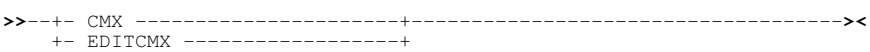
The Generate SQL panel view is the last panel view in the sequence and is common to all SELCOPY/i DB2 Alter Objects panel view sequences. For details, please refer to Generate SQL under the description of the DB2 Alter Table sequence of panels.

This view allows the user to select how the generated SQL ALTER TABLESPACE statement is to be implemented.

Primary Commands

The following primary commands are supported by selected views in the DB2 Alter Table Space sequence of panel views. If issued from a panel view in which the command is not valid, the message "ZZSP102E Primary command not valid in the current context" is displayed.

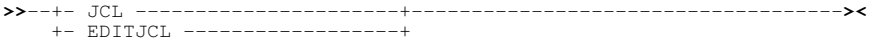
CMX



Applicable only from the Generate SQL view, CMX generates the SQL statement and copies it to an in-storage output file in a format suitable for subsequent execution using the ACTION (default >F16>) facility.

CMX is assigned to <F17> by default.

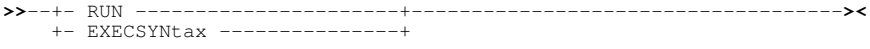
JCL



Applicable only from the Generate SQL view, JCL generates the SQL statement and copies it to an in-storage output file with JCL statements that execute the DSNTIAD facility. This job may be submitted to batch using the SELCOPY/i text editor SUBMIT primary command.

JCL is assigned to <F18> by default.

RUN



Applicable only from the Generate SQL view, RUN verifies input fields in all applicable panel views and then executes the generated SQL statement. This is the default action on pressing <Enter> from the last panel view in the sequence (i.e. the **Generate SQL** view.)

SELECT

SELECT will open a sub-panel that is applicable to the current panel view. Each of the panel views listed below identify the sub-panel opened on executing SELECT. In all other panel views, SELECT is invalid.

- **Table Space Type Attributes**
  For **Partitioned** and **Universal Partioned-by-Range** table spaces only, SELECT opens the Table Space Partition Attributes sub-panel, used to specify individual partition information.

SELECT is assigned to <F5> by default.

Drop DB2 Objects

Drop DB2 Objects Panel

The Drop DB2 Object sequence of panel views (ZZS2D001) generate an SQL SQL DROP statement to drop the selected object type from the current DB2 server. Note that the current DB2 server (subsystem name) is displayed in the panel window title bar.

These sequence of panel views are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select Drop, option 6, from the SELCOPYi DB2 primary option menu. (DB2 6)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made in a the panel views. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL DROP syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

Note that successful drop of some DB2 objects is dependent upon the version of DB2 used by the connected DB2 system to which this panel applies. An SQL error message will occur if an SQL DROP parameter field is used which is unsupported by the DB2 version. Please refer to the relevant edition of the *"DB2 for z/OS SQL Reference"*.

The first panel view allows the user to select the type of object to be dropped by entering the relevant option number or by positioning the cursor on the required option and pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Menu Bar Items

`File`
: The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Help`
: Open the general help for the Drop DB2 Objects option menu panel.

Options

1 Storage group	9 Type	17 Clone Table
2 Database	10 Function	
3 Table space	11 Stored procedure	
4 Table	12 Trigger	
5 View	13 Sequence	
6 Alias	14 Role	
7 Index	15 Trusted context	
8 Synonym	16 Package	

Panel Input/Output Fields

`Confirm foreground execution of DB2 object drop.`
: Enter "/" in this option field to ensure that, prior to performing foreground execution of an SQL DROP operation from the Generate SQL panel view, SELCOPYi will first prompt the user for confirmation.

Drop DB2 Storage Group

A DB2 storage group object is to be dropped from the current DB2 server.

The specified storage group must not be one that is in use by a table space or index space.

Field Entries

`Storage Group Name>`
: Identifies the name of the DB2 storage group to be dropped from the current server. Maximum length of a storage group name is 128 characters.

  Enter wild card "*" or "%" in this field to select from a list of storage groups eligible for drop.

Drop DB2 Database

A DB2 database object is to be dropped from the current DB2 server.

Dropping a database will also drop all of its table spaces, tables, index spaces and indexes.

Field Entries

Database Name>
Identifies the name of the DB2 database to be dropped from the current server. Maximum length of a database name is 8 characters.

Enter wild card "*" or "%" in this field to select from a list of databases eligible for drop.

Drop DB2 Table Space

A DB2 table space object is to be dropped from the current DB2 server.

Dropping a table space will also drop all of its tables. Note that the name must not identify a catalog table space or a table space that has been implicitlty defined for LOB or XML columns.

Field Entries

Database>
Identifies the name of the DB2 database at the current server which contains the table space to be dropped. This database name must not be that of the DB2 catalog database (DSNDB06). Maximum length of a database name is 8 characters.

Enter wild card "*" or "%" in this field to select from a list of databases which may contain table spaces eligible for drop.

Table Space Name>
Identifies the name of the DB2 table space to be dropped from the current server. Maximum length of a table space name is 8 characters.

Enter wild card "*" or "%" in this field to select from a list of table spaces eligible for drop.

Drop DB2 Table

A DB2 table object is to be dropped from the current DB2 server.

Dropping a table will also drop all aliases, synonyms, views indexes and privileges on that table; all referential constraints in which the table is a parent or dependent and, if implicitly created, the table space containing the table.

Field Entries

Table Owner>
Identifies the owner (schema) of the DB2 table to be dropped from the current server. Maximum length of a table schema is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of tables eligible for drop.

Table Name>
Identifies the name of the DB2 table to be dropped from the current server. Maximum length of a table name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of tables eligible for drop.

Drop DB2 View

A DB2 view object is to be dropped from the current DB2 server.

Dropping a view will also drop all synonyms, other views or materialised query tables defined on the view and privileges on the view.

Field Entries

View Owner>
Identifies the owner (schema) of the DB2 view to be dropped from the current server. Maximum length of a view schema is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of views eligible for drop.

View Name>
Identifies the name of the DB2 view to be dropped from the current server. Maximum length of a view name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of views eligible for drop.

Drop DB2 Alias

A DB2 alias object is to be dropped from the current DB2 server.

Field Entries

Alias Owner>
Identifies the owner (schema) of the DB2 alias to be dropped from the current server. Maximum length of a alias schema is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of aliass eligible for drop.

Alias Name>
Identifies the name of the DB2 alias to be dropped from the current server. Maximum length of a alias name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of aliass eligible for drop.

Drop DB2 Index

A DB2 index object is to be dropped from the current DB2 server.

Dropping an index will also drop the index space containing the index.

Field Entries

Index Owner>
Identifies the owner (schema) of the DB2 index to be dropped from the current server. Maximum length of a index schema is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of indexes eligible for drop.

Index Name>
Identifies the name of the DB2 index to be dropped from the current server. Maximum length of a index name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of indexes eligible for drop.

Drop DB2 Synonym

A DB2 synonym object is to be dropped from the current DB2 server.

Field Entries

Synonym Name>
Identifies the name of the DB2 synonym to be dropped from the current server. Maximum length of a synonym name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of synonyms eligible for drop.

Drop DB2 Distinct Type

A DB2 distinct type object is to be dropped from the current DB2 server.

A distinct type may not be dropped if it used in the definition of other DB2 objects.

Field Entries

Type Schema>
Identifies the schema of the DB2 distinct type to be dropped from the current server. Maximum length of a distinct type schema is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of distinct types eligible for drop.

Type Name>
Identifies the name of the DB2 distinct type to be dropped from the current server. Maximum length of a distinct type name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of distinct types eligible for drop.

Drop DB2 Function

A DB2 user-defined function object is to be dropped from the current DB2 server.

A function may not be dropped if it used in the definition of other DB2 objects.

If the specified schema and name do not uniquely identify a user-defined function, a list of matching functions is displayed from which the reuired function may be selected. The specific function name will then be used to identify the function to be dropped.

Field Entries

Function Schema>
Identifies the schema of the DB2 function to be dropped from the current server. Maximum length of a function schema is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of functions eligible for drop.

Function Name>
Identifies the name of the DB2 function to be dropped from the current server. Maximum length of a function name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of functions eligible for drop.

Specific Name>
A non-eneterable field which identifies the specific function name of the function selected for drop.

Drop DB2 Stored Procedure

A DB2 stored procedure object is to be dropped from the current DB2 server.

All versions of the native SQL procedure are dropped; all privileges on the procedure are also dropped. In addition, any plans or packages that are dependent on the procedure are marked as being invalid.

Field Entries

Procedure Schema>
Identifies the schema of the DB2 stored procedure to be dropped from the current server. Maximum length of a stored procedure schema is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of stored procedures eligible for drop.

Procedure Name>
Identifies the name of the DB2 stored procedure to be dropped from the current server. Maximum length of a stored procedure name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of stored procedures eligible for drop.

Drop DB2 Trigger

A DB2 trigger object is to be dropped from the current DB2 server.

Whenever a trigger is directly or indirectly dropped, all privileges on the trigger are also dropped and the associated trigger package is freed. The name of that trigger package is the same as the trigger name and the collection ID is the schema name.

If a trigger has current, previous, and original copies, the DROP statement will drop all copies.

Field Entries

Trigger Schema>
Identifies the schema of the DB2 trigger to be dropped from the current server. Maximum length of a trigger schema is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of triggers eligible for drop.

Trigger Name>
Identifies the name of the DB2 trigger to be dropped from the current server. Maximum length of a trigger name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of triggers eligible for drop.

Drop DB2 Sequence

A DB2 user-defined sequence object is to be dropped from the current DB2 server.

A sequence may not be dropped if it used in the definition of a DB2 trigger or is used by an in-line SQL function.

Whenever a sequence is dropped, all privileges on the sequence are also dropped, and the plans and packages that refer to the sequence are invalidated.

Field Entries

Sequence Schema>
Identifies the schema of the DB2 sequence to be dropped from the current server. Maximum length of a sequence schema is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of sequences eligible for drop.

Sequence Name>
Identifies the name of the DB2 sequence to be dropped from the current server. Maximum length of a sequence name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of sequences eligible for drop.

Drop DB2 Role

A DB2 role object is to be dropped from the current DB2 server.

A role may not be dropped if it associated with a trusted context or is the owner of other DB2 objects.

Field Entries

Role Name>
Identifies the name of the DB2 role to be dropped from the current server. Maximum length of a role name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of roles eligible for drop.

Drop DB2 Trusted Context

A DB2 trusted context object is to be dropped from the current DB2 server.

When a trusted context is dropped, all associations to attributes (IP addresses, job names) and associations to users of the trusted context are dropped.

Field Entries

Trusted Context>
Identifies the name of the DB2 trusted context to be dropped from the current server. Maximum length of a trusted context name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of trusted contexts eligible for drop.

Drop DB2 Package

A DB2 package object is to be dropped from the current DB2 server.

The name must not identify a trigger package or a package that is associated with an SQL routine.

If a package has current, previous, and original copies, the DROP statement will drop all copies.

Field Entries

Collection-id>
Identifies the name of the package collection at the current server containing the DB2 package to be dropped. Maximum length of a package collection id is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of packages eligible for drop.

Package Name>
Identifies the name of the DB2 package to be dropped from the current server. Maximum length of a package name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of packages eligible for drop.

Version>
Identifies the version of the DB2 package to be dropped. Maximum length of a package name is 122 characters.

If no version is specified, the null version is implied.

Drop DB2 Clone Table

A DB2 clone table object is to be dropped from the current DB2 server.

In order to drop a clone table, an ALTER TABLE DROP CLONE statement will be generated referencing the table on which the specified table is a clone.

Field Entries

Clone Table Owner>
Identifies the owner (schema) of the DB2 clone table to be dropped from the current server. Maximum length of a clone table schema is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of clone tables eligible for drop.

Clone Table Name>
Identifies the name of the DB2 clone table to be dropped from the current server. Maximum length of a clone table name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of clone tables eligible for drop.

Generate SQL

The Generate SQL panel view is the last view in the sequence, displayed following any of the Drop object panel views.

This view allows the user to select how the generated SQL DROP statement is to be implemented.

Generate SQL - Panel Fields

Drop Object:
A non-enterable field displaying the type of object being dropped.

Action:
Enter "/" (slash) or any non-blank character to select the the action to be performed with the generated SQL statement on completion of the drop object sequence of panel views.

For all the actions below, except Execute immediately, the SQL will be displayed in an edit view. The appropriate command or facility may be issued by the user to subsequently execute the generated SQL statement.

Copy to a file
Copy the generated SQL statement to the output file specified by the SQL Output File fields below.

The SQL statement may subsequently be executed using the EXECSQL primary command.

Display an in-storage copy
Copy the generated SQL statement to an in-storage output file with a temporary DSN.

The SQL statement may subsequently be executed using the EXECSQL primary command.

Display as an executable line command
Copy the generated SQL statement to an in-storage output file in a format suitable for subsequent execution using the ACTION (default >F16>) facility.

Display within generated batch JCL
Copy the generated SQL statement to an in-storage output file and enclose it within JCL which executes the DSNTIAD load module.

The SQL statement may subsequently be executed using the SUBMIT primary command.

Execute immediately
Opens the Execute SQL Statements panel and immediately executes the generated SQL statement to drop the DB2 object. DB2 SQL messages are also displayed in this panel.

If the Confirm foreground execution of DB2 object drop option was selected in the Drop DB2 Object Menu panel view, a confirmation pop-up window will be displayed before actioning the drop.

SQL Output File:
Applicable only if the Copy to file action has been selected.

Input fields which together identify a single output file (sequential data set, HFS file path or PDS/PDSE library member) to which the generated SQL statement will be copied. This output file may be a new or existing data set, HFS file or library member.

DSN/Path>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

If a DSN is specified for a data set (of orgaisation PS or PO) that does not already exist, the Allocate NonVSAM data set dialog window will be opened to create the new output file.

Member>

If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcard characters "*" (asterisk) or "%" (percent), or is blanked out.

Model Dsn>

Applicable only if the Copy to file action has been selected and **DSN/Path>** specifies a new data set or PDS/PDSE library name.

This field specifies the DSN of an existing sequential or PDS/PDSE library that will be used to model a new data set in the Allocate NonVSAM dialog window.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Append>

Applicable only if the Copy to file action has been selected.

Enter "/" (slash) or any non-blank character in this field to append the generated SQL statement to existing text in the output file. If not, existing text will be replaced by the SQl statement.

List DB2 Objects

List DB2 Objects Menu Panel

The List DB2 Object Menu panel (ZZS2L000) is an interactive panel window, opened on selection of option 7. in the DB2 Primary options menu or option 12. 'DB2' from the List Menu.

This panel allows the user to select the type of object to be listed by entering the relevant option number or by positioning the cursor on the required option and pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Menu Bar Items

`File`
: The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Help`
: Open the general help for the List DB2 Objects option menu panel.

Options

1 Storage groups 2 Databases 3 Table spaces 4 Tables 5 Views 6 Aliases 7 Indexes 8 Synonyms 9 Types	10 Triggers 11 Global temporary tables 12 Sequences 13 Roles 14 Trusted contexts 15 Columns 16 Volumes 17 Table space parts

List Storage groups

The List Storage groups panel (ZZS2LSTG) may be used to list storage groups defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 1. Storage groups from the List DB2 Object options menu.
- Execute the command **LDSTG** with or without parameters from the command line of any window.
- Execute the prefix command "**SG**" against an entry in the List Database panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog table SYSIBM.SYSSTOGROUP. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Location>`
Used to specify the server location of the storage groups to be displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Owner>`
Used to specify a filter on storage group owner (creator) authorisation ID.
A storage group owner ID has a maximum length of 128 characters.

`Name>`
Used to specify a filter on storage group name.
A storage group name has a maximum length of 128 characters.

`Catalog>`
Used to specify a filter on ICF catalog name/alias associated with the storage group.
The ICF catalog name/alias has a maximum length of 8 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command DB.
D	Drop the storage group.
DB	List Databases in the storage group.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

List Databases

The List Databases panel (ZZS2LDB0) may be used to list databases defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 2. Databases from the List DB2 Object options menu.
- Execute the command **LDDB** with or without parameters from the command line of any window.
- Execute the prefix command "**DB**" against an entry in the List Storage Groups or List Table Spaces panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog table SYSIBM.SYSDATABASE. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Location>`
Used to specify the server location of the databases to be displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Owner>`
Used to specify a filter on database owner (creator) authorisation ID.
A database owner ID has a maximum length of 128 characters.

`Name>`
Used to specify a filter on database name.
A database name has a maximum length of 8 characters.

`Storage Group>`
Used to specify a filter on the default storage group associated with the database.
The storage group name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command T.
D	Drop the database.
SG	List Storage Group to which the entry belongs.
T	List Tables in the database entry.
TS	List Table Spaces in the database entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

List Table spaces

The List Table spaces panel (ZZS2LTS0) may be used to list table spaces defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 3. Table spaces from the List DB2 Object options menu.
- Execute the command **LDTSP** with or without parameters from the command line of any window.
- Execute the prefix command "**TS**" against an entry in the List Database panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog table SYSIBM.SYSTABLESPACE. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Location>`
Used to specify the server location of the table spaces to be displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Owner>`
Used to specify a filter on table space owner (creator) authorisation ID.
A table space owner ID has a maximum length of 128 characters.

`Name>`
Used to specify a filter on table space name.
A table space name has a maximum length of 8 characters.

`Database>`
Used to specify a filter on the Database name to which the table space belongs.
The database name has a maximum length of 8 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command T.
D	Drop the table space.
DB	List database to which the entry belongs.
T	List Tables in the table space entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

List Tables

The List Tables panel (ZZS2LTAB) may be used to list tables defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 4. Tables from the List DB2 Object options menu.
- Execute the command **LDTAB** with or without parameters from the command line of any window.
- Execute prefix command "**T**" against an entry in a List Databases, List Table Spaces or List Tablespace partitions panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog table SYSIBM.SYSTABLES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Location>`
Used to specify the server location of the tables to be displayed. If left blank, the local server for the connected DB2 sub-system is used. A server location has a maximum length of 16 characters.

`Owner>`
Used to specify a filter on table owner (schema). A table schema has a maximum length of 128 characters.

`Name>`
Used to specify a filter on table name. A table name has a maximum length of 128 characters.

`Database>`
Used to specify a filter on the database name to which the table belongs. The database name has a maximum length of 8 characters.

`Table Space>`
Used to specify a filter on the table space name in which the table is defined. The table space name has a maximum length of 8 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command E.
A	Create an alias for the table entry.
AL	List aliases for the table entry.
B	Open the DB2 Browse panel to browse contents of the table entry.
BI	Browse the contents of the table entry.
CL	List columns belonging to the table entry.
D	Drop the table.
E	Open the DB2 Edit panel to edit the contents of the table entry.
EI	Edit the contents of the table entry.
I	Display detailed information about the table entry.
RL	List tables related to the table entry via a referential constraint.
S	Create a synonym for the table entry.
SC	Create a SELCOPY batch job using the DB2 table as input.
SL	List synonyms for the table entry.
T	Create a trigger for the table entry.
TL	List triggers for the table entry.
V	Create a view using the table entry.
VL	List views using the table entry.
X	Create an index using the table entry.
XL	List indexes using the table entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

List Related Tables

The List Related Tables panel (ZZS2LREL) may be used to list DB2 tables that are involved in a referential constraint (parent/foreign key) relationship as the parent or dependent of the selected DB2 table.

The panel is an interactive panel window which contains an embedded table and may be started via the following:

- Execute the prefix command "**RL**" against a DB2 table entry in another List Related Tables panel or in a List Tables panel.
- From a DB2 table data edit view, execute the prefix command "**RE**" or primary command "**REDIT**" against a row for which related table rows will be displayed.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

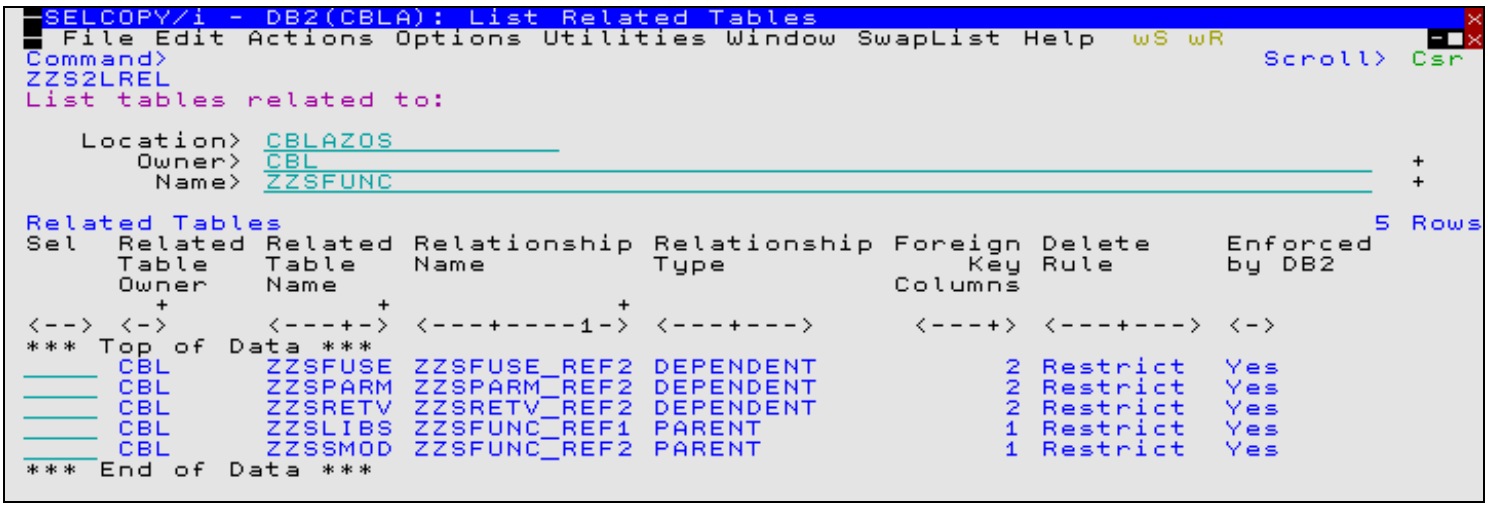


Figure 270. DB2 Related Tables.

Panel Fields

Location>
Used to specify the server location of the related tables displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

Owner>
Input field used to specify a filter on the subject DB2 table owner (schema).
A table schema has a maximum length of 128 characters.

Name>
Input field used to specify a filter on the subject DB2 table name.
A table name has a maximum length of 128 characters.

Sel
Input field used to specify one of the supported prefix line commnds for execution against the particular table entry.

Related Table Owner
Non-enterable field displaying the owner (schema) of the related DB2 table.

Related Table Name
Non-enterable field displaying the name of the related DB2 table.

Relationship Name
Non-enterable field displaying the name of the referential constraint that links this DB2 table to the subject DB2 table.

Relationship Type
Non-enterable field displaying the referential constraint relationship of the DB2 table with the subject DB2 table.

1. **PARENT** indicates that the DB2 table contains the parent key and the subject DB2 table the foreign key.
2. **DEPENDENT** indicates that the DB2 table contains the foreign key and the subject DB2 table the parent key.

Foreign Key Columns
Non-enterable field displaying the number of key columns from each table that are involved in the referential constraint relationship.

Delete Rule
Non-enterable field displaying the action performed when a row of the parent DB2 table is deleted.

1. **No Action** indicates that the DELETE operation will fail if a dependent DB2 table contains rows that have a matching foreign key value. The contents of the parent DB2 table remains unchanged. This delete rule is enforced **after** all other constraints, including referential constraints defined with delete rule CASCADE or SET NULL.

2. **Cascade** indicates that the parent row will be deleted and all rows in the dependent DB2 table that have matching foreign key values will also be deleted.

3. **Set null** indicates that the parent row will be deleted and all rows in the dependent DB2 table that have matching foreign key values will not be deleted but instead the foreign key column values will be set to NULL (provided the columns that constitute the foreign key are nullable). Other values in the dependent row are not affected.

4. **Restrict** indicates that the DELETE operation will fail if a dependent DB2 table contains rows that have a matching foreign key value. The contents of the parent DB2 table remains unchanged. This delete rule is enforced **before** all other constraints, including referential constraints defined with delete rule CASCADE or SET NULL.

`Enforced by DB2`
Non-enterable field indicating whether or not the referential constraint is enforced by DB2 (Yes or No).

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command E.
A	Create an alias for the table entry.
AL	List aliases for the table entry.
B	Open the DB2 Browse panel to browse contents of the table entry.
BI	Browse the contents of the table entry.
CL	List columns belonging to the table entry.
D	Drop the table.
E	Open the DB2 Edit panel to edit the contents of the table entry.
EI	Edit the contents of the table entry.
RE	Applicable only if a DB2 table edit/browse exists. Edit the related table entry selecting only rows which match the parent key/foreign key value in the focus row of the current DB2 SDE edit/browse view.
RL	List tables related to the table entry via a referential constraint.
S	Same as RE.
SC	Create a SELCOPY batch job using the DB2 table as input.
SL	List synonyms for the table entry.
SY	Create a synonym for the table entry.
T	Create a trigger for the table entry.
TL	List triggers for the table entry.
V	Create a view using the table entry.
VL	List views using the table entry.
XL	List indexes using the table entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

List Views

The List Views panel (ZZS2LVI0) may be used to list views defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 5. Views from the List DB2 Object options menu.
- Execute the command **LDVIEW** with or without parameters from the command line of any window.
- Execute the prefix command "**VL**" against an entry in the List Tables panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog table.

List columns are those defined in the DB2 catalog view SYSIBM.SYSVIEWS. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Location>`
Used to specify the server location of the DB2 views displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Owner>`
Used to specify a filter on view owner (schema).
A view schema has a maximum length of 128 characters.

`Name>`
Used to specify a filter on view name.
A view name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command E.
B	Browse the contents of the view entry.
CL	List columns belonging to the view entry.
D	Drop the view.
E	Open the DB2 Edit panel to edit the contents of the view entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

List Aliases

The List Aliases panel (ZZS2LALI) may be used to list aliases defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 6. Aliases from the List DB2 Object options menu.
- Execute the command **LDALI** with or without parameters from the command line of any window.
- Execute the prefix command "**AL**" against an entry in the List Tables panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSTABLES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Location>`
Used to specify the server location of the aliases displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Owner>`
Used to specify a filter on alias owner (schema).
An alias schema has a maximum length of 128 characters.

`Name>`
Used to specify a filter on alias name.
An alias name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command E.
B	Browse the contents of the table to which the alias entry applies.
CL	List columns belonging to the table to which the alias entry applies.
D	Drop the alias.
E	Open the DB2 Edit panel to edit the contents of the table to which the alias entry applies.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

List Indexes

The List Indexes panel (ZZS2LINX) may be used to list indexes defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window and may be started via the following:

- Select option 7. Indexes from the List DB2 Object options menu.
- Execute the command **LDINDEX** with or without parameters from the command line of any window.
- Execute list prefix command "XL" against an entry in the DB2 List Tables panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSINDEXES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

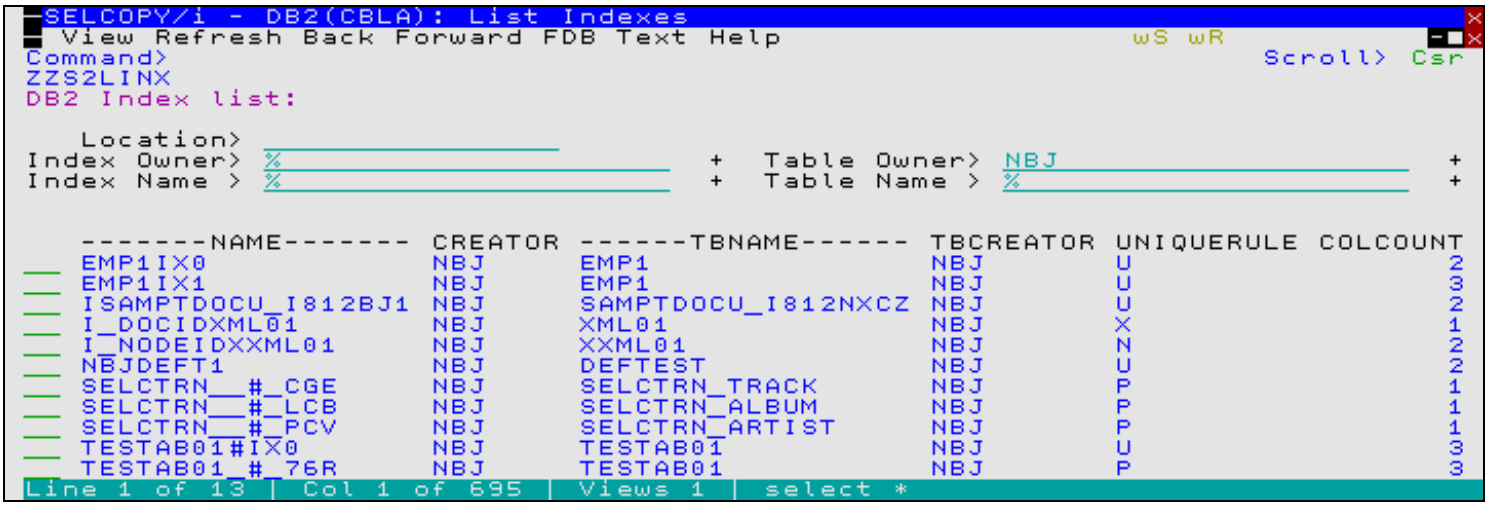


Figure 271. DB2 List Indexes.

Panel Input Fields

Location>
Used to specify the server location of the indexes displayed. If left blank, the local server for the connected DB2 sub-system is used. A server location has a maximum length of 16 characters.

Index Owner>
Used to specify a filter on index owner (schema). An index schema has a maximum length of 128 characters.

Index Name>
Used to specify a filter on index name. An index name has a maximum length of 128 characters.

Table Creator>
Used to specify a filter on table owner (schema) for the table on which the index is defined. A table schema has a maximum length of 128 characters.

Table Name>
Used to specify a filter on table name for the table on which the index is defined. A table name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command KL.
B	Browse the contents of the table to which the index entry applies.
KL	List Index Keys.
D	Drop the index.
E	Open the DB2 Edit panel to edit the contents of the table to which the index entry applies.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

List Index Keys

The List Index Keys panel (ZZS2LKEY) may be used to list key columns defined for an index at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Execute the line-command **KL** from the List Indexes panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSKEYS. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Index Location>`
Used to specify the server location of the index for which key columns are displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Column Name>`
Used to specify a filter on then index key column name.

`Index Name>`
Used to specify a filter on index name.
An index name has a maximum length of 128 characters.

`Index Owner>`
Used to specify a filter on index owner (schema).

Prefix Line Commands

The Index Keys list does not support any prefix commands.

List Synonyms

The List Synonyms panel (ZZS2LSYN) may be used to list synonyms defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 8. Synonyms from the List DB2 Object options menu.
- Execute the command **LDSYN** with or without parameters from the command line of any window.
- Execute the prefix command "**SL**" against an entry in the List Tables panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSSYNONYMS. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Location>`
Used to specify the server location of the synonyms displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Owner>`
Used to specify a filter on synonym owner (creator) authorisation ID.
A synonym owner ID has a maximum length of 128 characters.

`Name>`
Used to specify a filter on synonym name.
A synonym name has a maximum length of 128 characters.

`Object Owner>`
Used to specify a filter on the DB2 object (table or view) schema to which the synonym applies.
A table or view schema has a maximum length of 128 characters.

`Object Name>`
Used to specify a filter on the DB2 object (table or view) name to which the synonym applies.
A table or view name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command E.
B	Browse the contents of the table to which the synonym entry applies.
CL	List columns belonging to the table to which the synonym entry applies.
D	Drop the synonym.
E	Open the DB2 Edit panel to edit the contents of the table to which the synonym entry applies.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

List Distinct Types

The List Distinct Types panel (ZZS2LTYP) may be used to list distinct types defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 9. Types from the List DB2 Object options menu.
- Execute the command **LDTYP** with or without parameters from the command line of any window.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSDATATYPES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Location>`
Used to specify the server location of the distinct types displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Schema>`
Used to specify a filter on distinct type schema.
A distinct type schema has a maximum length of 128 characters.

`Owner>`
Used to specify a filter on distinct type owner ID.
A distinct type owner ID has a maximum length of 128 characters.

`Name>`
Used to specify a filter on distinct type name.
A distinct type name has a maximum length of 128 characters.

Prefix Line Commands

The Distinct Types list does not support any prefix commands.

List Triggers

The List Triggers panel (ZZS2LTRG) may be used to list triggers defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 10. Triggers from the List DB2 Object options menu.
- Execute the prefix command "**TL**" against an entry in the List Tables panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSTRIGGERS. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Location>`
Used to specify the server location of the triggers displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Schema>`
Used to specify a filter on trigger schema.
A trigger schema has a maximum length of 128 characters.

`Name>`
Used to specify a filter on trigger name.
A trigger name has a maximum length of 128 characters.

`Object Owner>`
Used to specify a filter on the DB2 object (table or view) schema to which the trigger applies.
A table or view creator ID has a maximum length of 128 characters.

`Object Name>`
Used to specify a filter on the DB2 object (table or view) name to which the trigger applies.
A table or view name has a maximum length of 128 characters.

Prefix Line Commands

The Triggers list does not support any prefix commands.

List Global Temporary Tables

The List Global Temporary Tables panel (ZZS2LGTT) may be used to list tables defined as being temporary at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 11. Tables from the List DB2 Object options menu.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog table SYSIBM.SYSTABLES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Location>`
Used to specify the server location of the tables displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Owner>`
Used to specify a filter on global temporary table owner (schema).
A table schema has a maximum length of 128 characters.

`Name>`
Used to specify a filter on the global temporary table name.
A table name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
CL	List columns in the global temporary table.
D	Drop the global temporary table.
B	Browse the tables contents.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

List Sequences

The List Sequences panel (ZZS2LSEQ) may be used to list sequences defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 12. Sequences from the List DB2 Object options menu.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSSEQUENCES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Location>`
Used to specify the server location of the sequences displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Schema>`
Used to specify a filter on sequence schema.
A sequence schema has a maximum length of 128 characters.

`Owner>`
Used to specify a filter on sequence owner.
A sequence owner has a maximum length of 128 characters.

`Name>`
Used to specify a filter on sequence name.
A sequence name has a maximum length of 128 characters.

Prefix Line Commands

The Sequences list does not support any prefix commands.

List Roles

The List Roles panel (ZZS2LROL) may be used to list roles defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 13. Roles from the List DB2 Object options menu.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSROLES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Location>`
Used to specify the server location of the roles displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Definer>`
Used to specify a filter on the authorisation ID (or role) that defined the role name.
A role authorisation ID or role name has a maximum length of 128 characters.

`Name>`
Used to specify a filter on role name.
A role name has a maximum length of 128 characters.

Prefix Line Commands

The Roles list does not support any prefix commands.

List Trusted Contexts

The List Trusted Context panel (ZZS2LTRC) may be used to list trusted context defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 14. Trusted Contexts from the List DB2 Object options menu.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSCONTEXT. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Location>`
Used to specify the server location of the trusted contexts displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Definer>`
Used to specify a filter on the authorisation ID (or role) that defined the trusted context.
An authorisation ID or role name has a maximum length of 128 characters.

`Name>`
Used to specify a filter on the trusted context name.
A trusted context name has a maximum length of 128 characters.

Prefix Line Commands

The Trusted Contexts list does not support any prefix commands.

List Columns

The List Columns panel (ZZS2LCOL) may be used to list columns defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 15. Columns from the List DB2 Object options menu.
- Execute the command **LDCOL** with or without parameters from the command line of any window.
- Execute the prefix command "**CL**" against an entry in the List Tables, List Views, List Aliases, List Indexes, List Synonyms or List Global Temporary Tables panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog table.

List columns are those defined in the DB2 catalog column SYSIBM.SYSCOLUMNS. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Table Location>`
Used to specify the server location of the columns displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Column Name>`
Used to specify a filter on column name.
A column name has a maximum length of 128 characters.

`Table Owner>`
Used to specify a filter on the schema (owner) of the table or view that contains the column.
A table or view schema has a maximum length of 128 characters.

`Table Name>`
Used to specify a filter on the name of the table or view that contains the column.
A table or view name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command E.
B	Browse the contents of the column entry.
BT	Browse the contents of the table to which the column entry belongs.
E	Open the DB2 Edit panel to edit the contents of the table to which the column entry belongs.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

List Volumes

The List Volumes panel (ZZS2LVOL) may be used to list volumes defined to storage groups at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 16. Volumes from the List DB2 Object options menu.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog view SYSIBM.SYSVOLUMES. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Location>`
Used to specify the server location of the storage group for which volumes are displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Owner>`
Used to specify a filter on storage group owner (creator) authorisation ID to which the volume is defined.
A storage group authorisation ID a maximum length of 128 characters.

`Name>`
Used to specify a filter on the storage group to which the volume is defined.
A storage group name has a maximum length of 128 characters.

Prefix Line Commands

The Volumes list does not support any prefix commands.

List Table Space Parts

The List Table space parts panel (ZZS2LTS1) may be used to list table space partitions defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 17. Table space parts from the List DB2 Object options menu.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog table SYSIBM.SYSTABLEPART. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

Panel Input Fields

`Location>`
Used to specify the server location of the table space partitions displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Table Space>`
Used to specify a filter on the table space name.
A table space name has a maximum length of 8 characters.

`Catalog>`
Used to specify a filter on ICF catalog name/alias used for table space partition free space allocation.
The ICF catalog name/alias has a maximum length of 8 characters.

`Database>`
Used to specify a filter on the Database name to which the table space belongs.
The database name has a maximum length of 8 characters.

`Stor Group>`
Used to specify a filter on the storage group used for table space allocation.
The storage group name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command T.
D	Drop the table space partition.
DB	List database to which the entry belongs.
T	List Tables in the table space partition entry.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

List Packages

The List Packages panel (ZZS2LPKG) may be used to list packages defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Select option 18. Packages from the List DB2 Object options menu.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog table SYSIBM.SYSPACKAGE. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

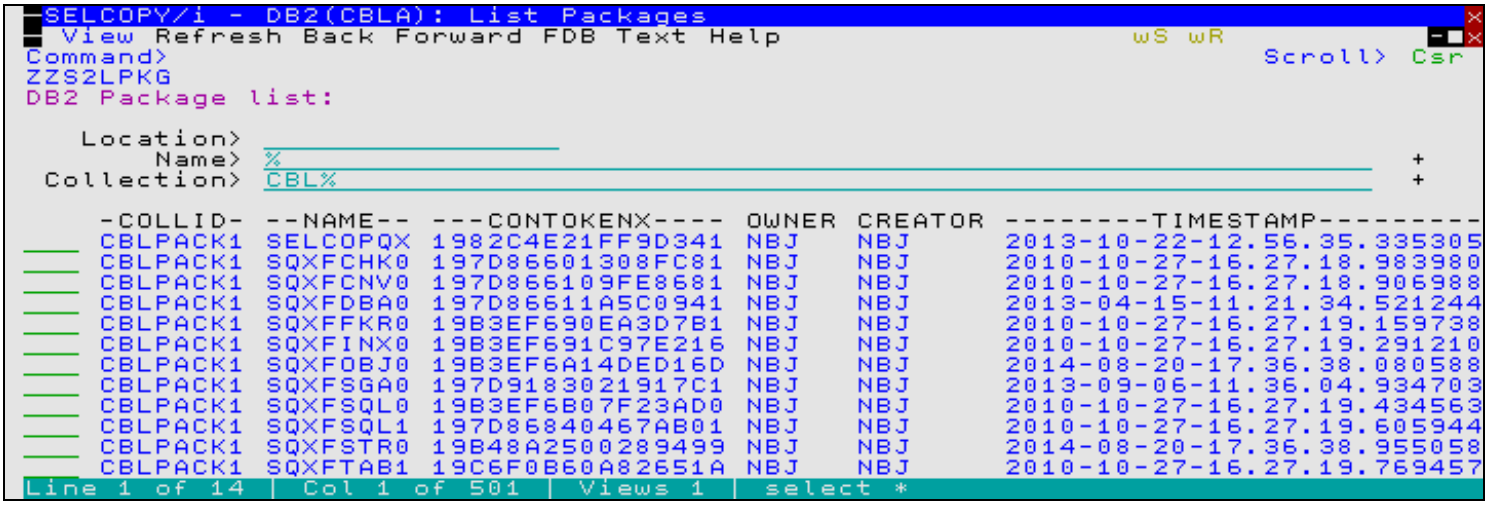


Figure 272. DB2 List Packages.

Panel Input Fields

Location>
Used to specify the server location of the packages displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

Name>
Used to specify a filter on the package name.

Collection>
Used to specify a filter on the package collection (COLLID) or, for trigger packages, the trigger schema name.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command DEP.
DEP	List Package Dependencies.
P	List Privileges.
/	Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command. Assigned to F16 by default.
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

List Package Dependencies

The List Package Dependencies panel (ZZS2LPKD) may be used to list dependencies for the selected package defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Type line-command 'DEP' from the List DB2 Packages window.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog table SYSIBM.SYSPACKDEP. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

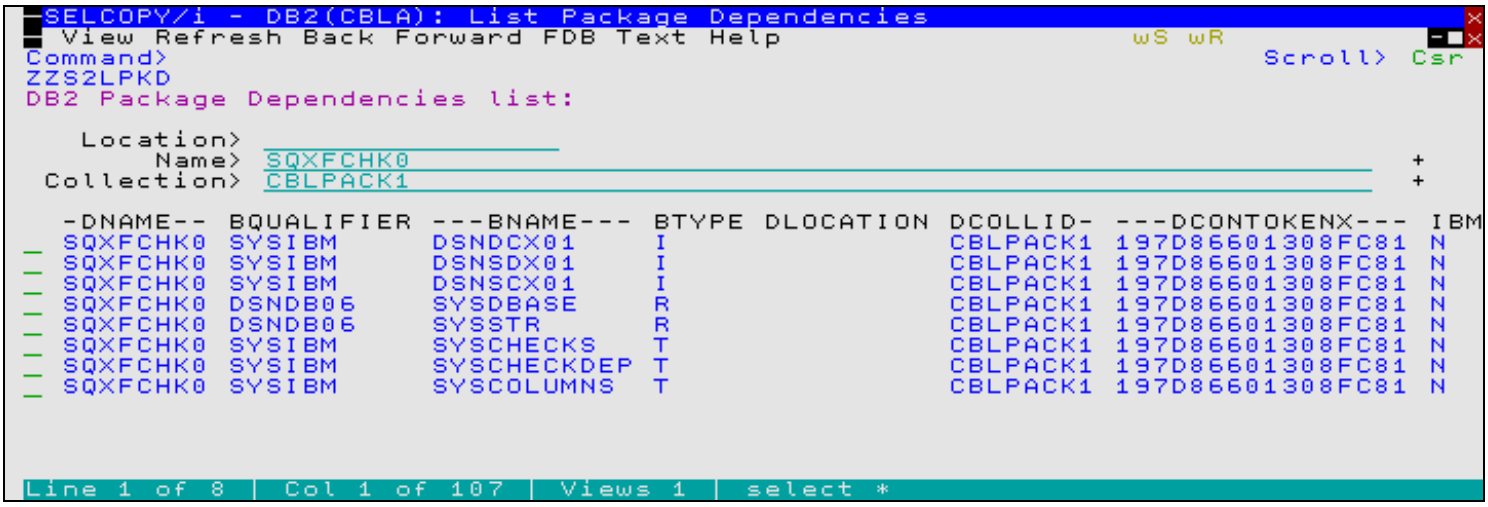


Figure 273. DB2 List Package Dependencies.

Panel Input Fields

`Location>`
Used to specify the server location of the package for which dependencies are displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Name>`
Used to specify a filter on the package name.

`Collection>`
Used to specify a filter on the package collection (COLLID) or, for trigger packages, the trigger schema name.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

List Package Privileges

The List Package Privileges panel (ZZS2LPKP) may be used to list privileges for the selected package defined at the local or remote DB2 server.

The panel is an interactive panel window which contains a child list type window (window class LISTFILE), and may be started via the following:

- Type line-command 'P' from the List DB2 Packages window.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to apply a filter before rows are fetched from the relevant catalog tables.

List columns are those defined in the DB2 catalog table SYSIBM.SYSPACKAUTH. See IBM publication "*DB2 SQL Reference*", "Appendix - DB2 Catalog Tables" for details of entries in this table.

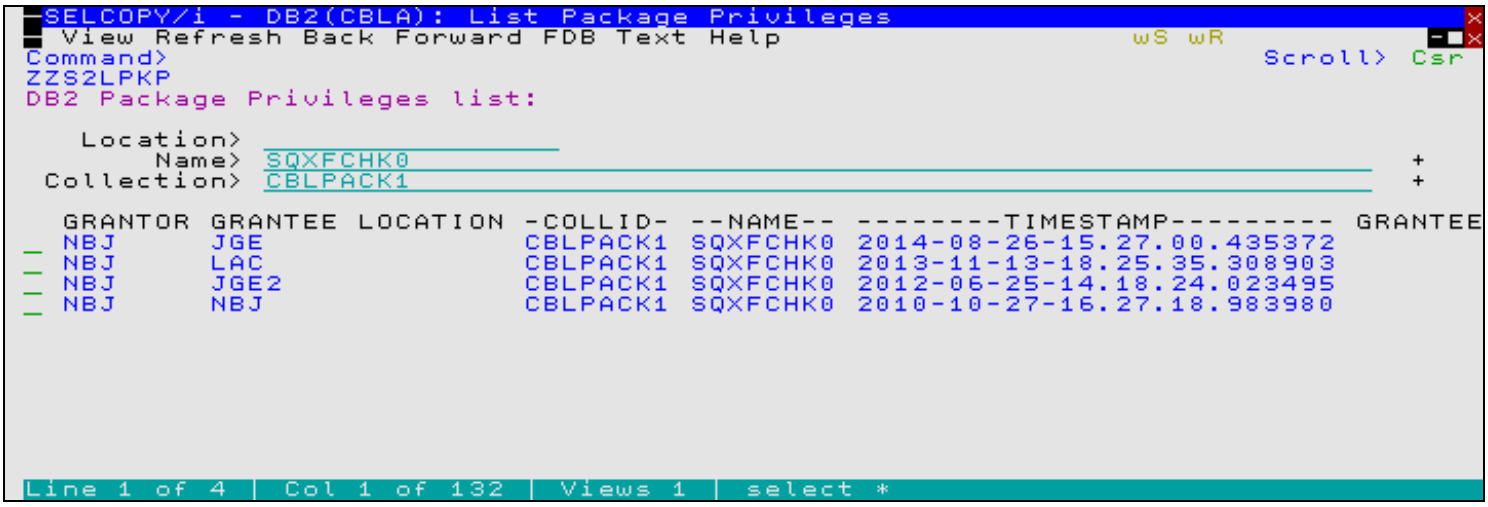


Figure 274. DB2 List Package Privileges.

Panel Input Fields

`Location>`
Used to specify the server location of the package for which privileges are displayed. If left blank, the local server for the connected DB2 sub-system is used.
A server location has a maximum length of 16 characters.

`Name>`
Used to specify a filter on the package name.

`Collection>`
Used to specify a filter on the package collection (COLLID) or, for trigger packages, the trigger schema name.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
>	Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns. Assigned to PF17 (Shift-F5) by default.

Select DB2 Objects

DB2 Object selection panels are interactive panel windows which contains an embedded list.

These types of panel are used extensively throughout SELCOPYi DB2 utility panels to select an input field value from a list of valid DB2 object entries.

The list of valid entries may be the result of an SQL query against the DB2 catalog tables or a SELCOPY/i in-storage table (i.e. a panel embedded table).

Select Storage Group

The Select Storage Group panel (ZZS2SSGx) may be used to select a storage group defined in the current DB2 server (sub-system.)

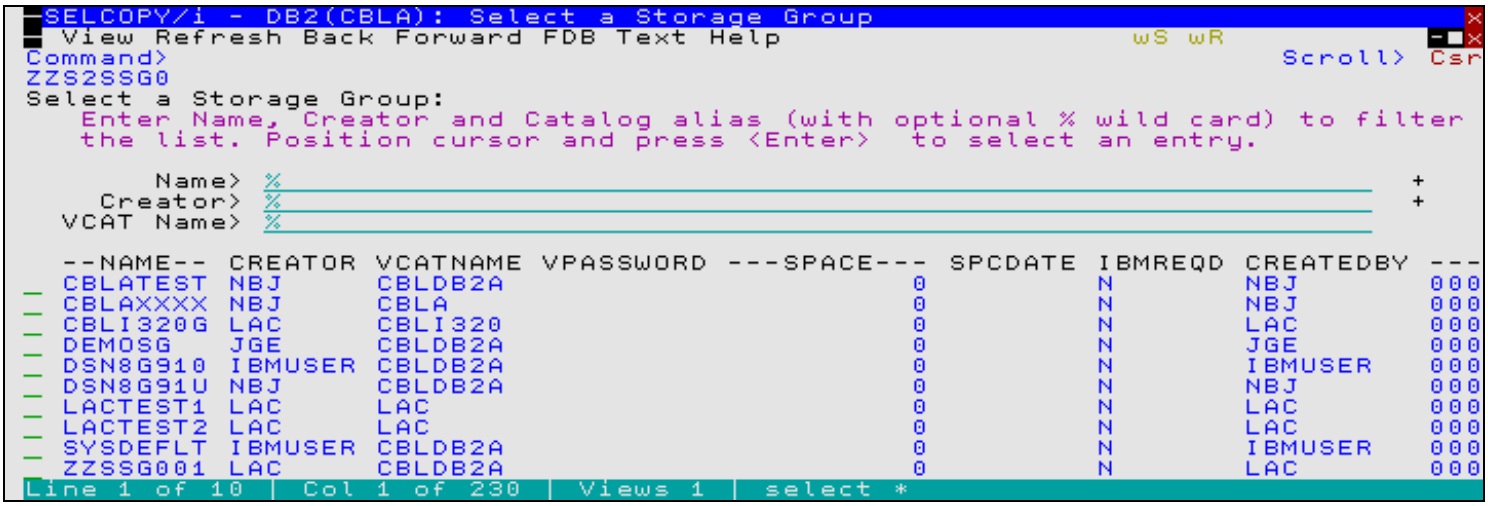


Figure 275. DB2: Select a Storage Group.

Variations of this panel exist as follow:

`Select Storage Group (ZZS2SSG0)`
: Used by create DB2 database, table space and index panel views to select a storage group for use in DB2 data set allocation. It is also used by the create DB2 storage group utility to select another DB2 storage group on which to model the new DB2 storage group definition panel input fields.

`Select Storage Group (ZZS2SSG2)`
: Used by drop DB2 storage group panel views to select a storage group that is eligible to be dropped.

  This selection panel omits storage group entries that are in use by existing table spaces or index spaces.

The contents of at least one of the selection panel input fields is set by an equivalent field in the parent panel which has invoked this selection panel.

The selection panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of storage group entries fetched from the DB2 catalog table SYSIBM.SYSSTOGROUP. See IBM publication "*DB2 SQL Reference*", "*Appendix - DB2 Catalog Tables*" for details on the column values displayed in this table.

Panel Input Fields

`Owner>`
: Used to specify a filter on storage group owner (creator) authorisation ID.
  A storage group creator ID has a maximum length of 128 characters.

`Name>`
: Used to specify a filter on the storage group name.
  A storage group name has a maximum length of 8 characters.

`Catalog>`
: Used to specify a filter on catalog name or alias (VCATNAME).
  A catalog name/alias entry has a maximum length of 8 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the storage group entry. Values associated with this entry will be inserted in the parent panel input fields.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <PF2> by default.

Select Database

The Select Database panel (ZZS2SDBx) may be used to select a database defined in the current DB2 server (sub-system.)

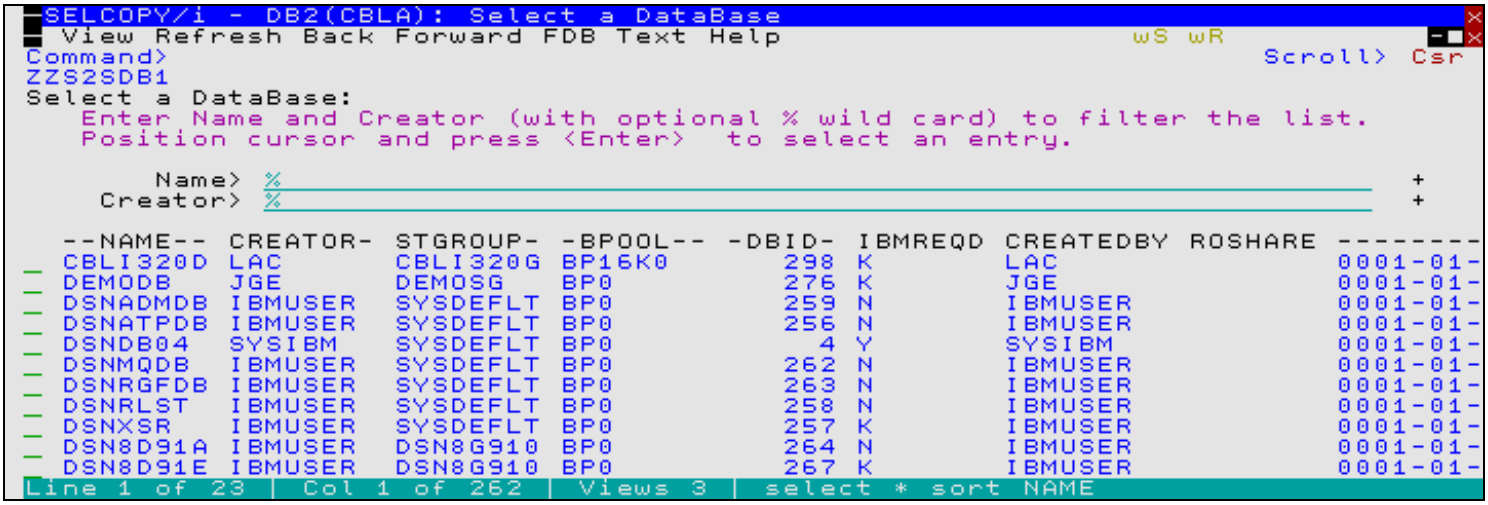


Figure 276. DB2: Select a Database.

The contents of at least one of the selection panel input fields is set by an equivalent field in the parent panel which has invoked this selection panel.

Databases that are invalid for use by the parent panel are omitted from the selection list. e.g. The DB2 catalog table tablespace (DSNDB06), implicitly defined tablespaces and work file tablespaces are omitted for create table panel tablespace name selection.

The selection panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of database entries fetched from the DB2 catalog table SYSIBM.SYSDATABASE. See IBM publication "*DB2 SQL Reference*", "*Appendix - DB2 Catalog Tables*" for details on the column values displayed in this table.

Panel Input Fields

Name>
Used to specify a filter on the database name.
A database name has a maximum length of 8 characters.

Creator>
Used to specify a filter on database creator (owner) authorisation ID.
A database creator ID has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the database entry. Values associated with this entry will be inserted in the parent panel input fields.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <PF17> by default.

Select Tablespace

The Select Tablespace panel (ZZS2STSx) may be used to select a tablespace defined in the current DB2 server (sub-system.)

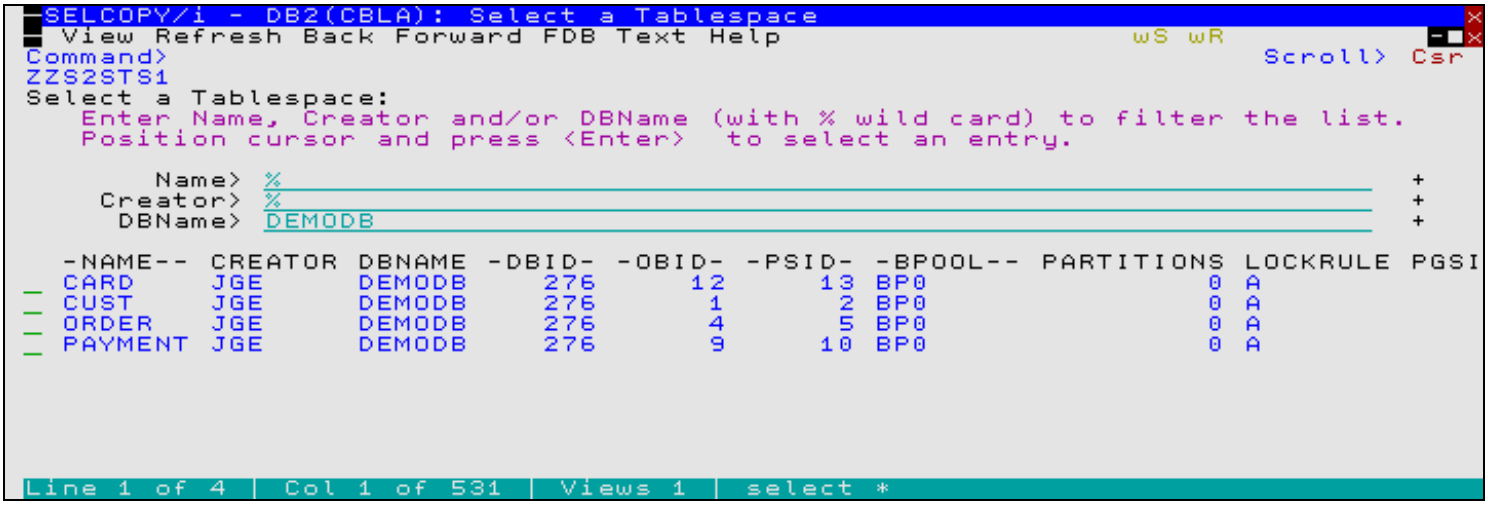


Figure 277. DB2: Select a Tablespace.

The contents of at least one of the selection panel input fields is set by an equivalent field in the parent panel which has invoked this selection panel.

Tablespaces that are invalid for use by the parent panel are omitted from the selection list. e.g. Work file, LOB, XML tablespaces and any partitioned or universally partitioned tablespaces that already contain a table definition are omitted for create table panel tablespace name selection.

The selection panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of tablespace entries fetched from the DB2 catalog table SYSIBM.SYSDATABASE. See IBM publication "*DB2 SQL Reference*", "*Appendix - DB2 Catalog Tables*" for details on the column values displayed in this table.

Panel Input Fields

Name>
Used to specify a filter on tablespace name.
A table space name has a maximum length of 8 characters.

Creator>
Used to specify a filter on tablespace creator (owner) authorisation ID.
A table space creator ID has a maximum length of 128 characters.

DBName>
Used to specify a filter on the database name to which the table space belongs.
The database name has a maximum length of 8 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the tablespace entry. Values associated with this entry will be inserted in the parent panel input fields.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <PF2> by default.

Select Table

The Select Table panels (ZZS2STBx) may be used to select a table, view or alias defined at the local or remote DB2 server.

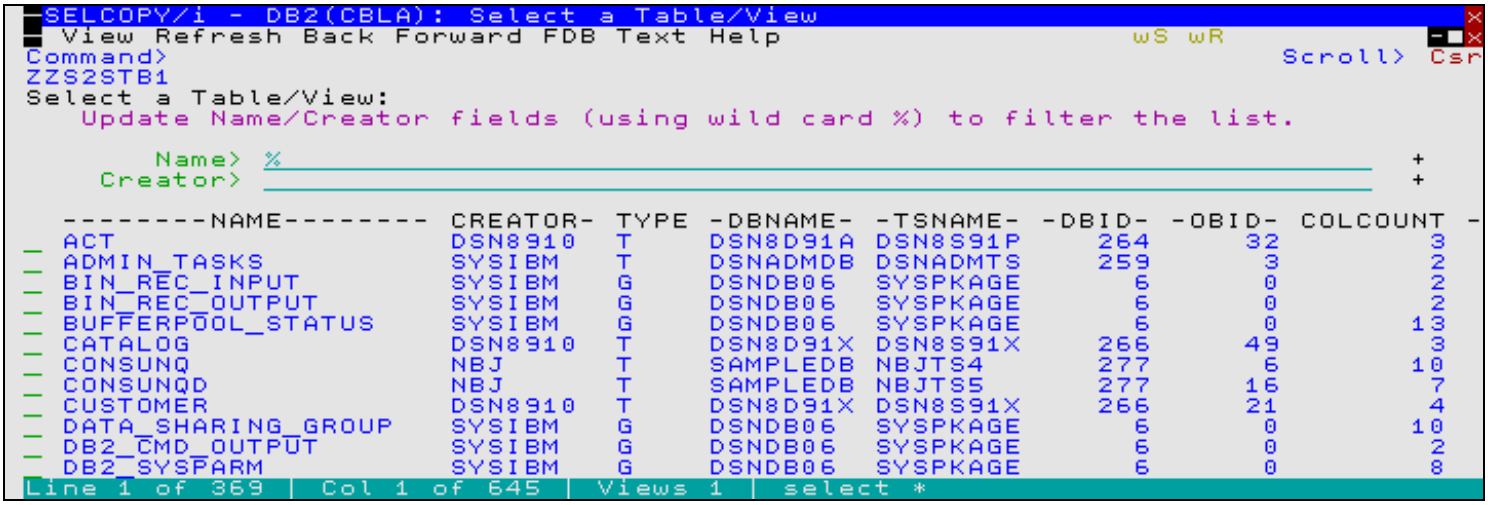


Figure 278. DB2: Select a Table/View.

Variations of this panel exist as follow:

Select Table/View (ZZS2STB0)
Used by edit, browse and compare DB2 table utilities to select an input table, and also by the create DB2 table utility to select a DB2 table or view on which to model new DB2 table definition panel input fields.

This table/view selection panel also filters entries based on location, database and/or tablespace name. In addition to DB2 table attributes, the list generated for this panel includes the number of partitions and number of tables defined in the tablespace to which the table belongs.

Select Table/View (ZZS2STB1)
Used by create DB2 table panel views to select a DB2 table or view on which to model a new DB2 table using CREATE TABLE LIKE syntax.

This table/view selection panel omits ALIAS entries, auxiliary and clone table entries.

Select Parent Table (ZZS2STB2)
Used by create DB2 table panel views to select a DB2 table to be used as a parent table in the definition of a referential constraint (parent/foreign key) relationship.

This table selection panel omits VIEW, ALIAS, DB2 catalog table and global temporary table entries.

Select Table/View (ZZS2STB3)
Used by create DB2 alias panel views to select a DB2 table or view to be assigned to new alias.

This table selection panel omits Auxiliary table entries and also implicit tables created for XML columns.

Select Table/View/Alias (ZZS2STB4)
Used by create DB2 synonym panel views to select a DB2 table, view or alias for which the synonym is created.

This table selection panel omits Auxiliary table entries and also implicit tables created for XML columns.

Select Table (ZZS2STB5)
Used by create DB2 trigger panel views to select a DB2 base table on which the trigger will operate.

This table selection panel omits VIEW, ALIAS, DB2 catalog, Auxiliary, Materialised Query, Clone and global temporary table entries.

Select View (ZZS2STB6)
Used by drop DB2 view and also by create DB2 trigger panel views to select a DB2 view on which the trigger will operate.

This table selection panel omits all ALIAS and table entries.

Select Table (ZZS2STB7)
Used by create DB2 index panel views to select a DB2 table on which the new index will be created.

This table selection panel omits all ALIAS, VIEW, Clone and global temporary table entries and also implicit tables created for XML columns.

This table selection panel also filters entries based on database and/or tablespace name. In addition to DB2 table attributes, the list generated for this panel includes number of partitions and number of tables defined in the tablespace to

which the table belongs.

Select Table (ZZS2STB8)
Used by create DB2 clone table panel views to select a DB2 table on which the clone table will be created.

This table selection panel displays only base tables that satisfy the criteria required for creating its clone table. See *"DB2 SQL Reference"* for details.

Select Table (ZZS2STB9)
Used by the drop DB2 table panel view to select a DB2 table eligible to be dropped.

This table selection panel displays only base tables, materialised query tables and global temporary tables.

Select Table (ZZS2STBA)
Used by the drop DB2 clone table panel view to select a DB2 clone table to be dropped.

This table selection panel displays only clone table entries with references to the base table on which they are cloned.

The contents of at least one of the selection panel input fields is set by an equivalent field in the parent panel which has invoked this selection panel.

The selection panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of table entries fetched from the DB2 catalog table SYSIBM.SYSTABLES. See IBM publication "*DB2 SQL Reference*", "*Appendix - DB2 Catalog Tables*" for details on the column values displayed in this table.

Panel Input Fields

Location>
Used to specify the server location of the table name.
A server location has a maximum length of 16 characters.

Owner>
Used to specify a filter on table schema (creator) ID.
A table schema has a maximum length of 128 characters.

Name>
Used to specify a filter on table name.
A table name has a maximum length of 128 characters.

DBName>
Used to specify a filter on the database name to which the table belongs.
The database name has a maximum length of 8 characters.

TSName>
Used to specify a filter on the table space name in which the table is defined.
The table space name has a maximum length of 8 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the table entry. Values associated with this entry will be inserted in the parent panel input fields.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <PF2> by default.

Select Alias

The Select Alias panel (ZZS2SAL0) may be used to select a DB2 alias which has been defined in the current DB2 server.

The contents of at least one of the selection panel input fields is set by an equivalent field in the parent panel which has invoked this selection panel.

The selection panel alias Owner and Name input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of DB2 aliases.

Panel Input Fields

Owner>
Used to specify a filter on alias owner (schema) id.
A schema has a maximum length of 128 characters.

Name>
Used to specify a filter on alias name.
A DB2 alias name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the alias entry. Values associated with this entry will be inserted in the parent panel input fields.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <F17> by default.

Select Index

The Select Index panel (ZZS2SIN0) may be used to select a DB2 index which has been defined in the current DB2 server.

The contents of at least one of the selection panel input fields is set by an equivalent field in the parent panel which has invoked this selection panel.

Indexes that are invalid for use by the parent panel are omitted from the selection list. e.g. Implicitly created indexes for tables containing an XML column are omitted for drop index name selection.

The selection panel index/target table Owner and Name input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of DB2 indexs.

Panel Input Fields

Index Owner>
Used to specify a filter on index owner (schema) id.
A schema has a maximum length of 128 characters.

Index Name>
Used to specify a filter on index name.
A DB2 index name has a maximum length of 128 characters.

Table Owner>
Used to specify a filter on the table or view owner (schema) id for which indexes are defined.
A schema has a maximum length of 128 characters.

Table Name>
Used to specify a filter on the table or view name for which indexes are defined.
A DB2 index name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the index entry. Values associated with this entry will be inserted in the parent panel input fields.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <F17> by default.

Select Synonym

The Select Synonym panel (ZZS2SSY0) may be used to select a DB2 synonym which has been defined in the current DB2 server.

The contents of at least one of the selection panel input fields is set by an equivalent field in the parent panel which has invoked this selection panel.

The selection panel synonym/target table Owner and Name input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of DB2 synonyms.

Panel Input Fields

Synonym Owner>
Used to specify a filter on synonym owner (schema) id.
A schema has a maximum length of 128 characters.

Synonym Name>
Used to specify a filter on synonym name.
A DB2 synonym name has a maximum length of 128 characters.

Table Owner>
Used to specify a filter on the table or view owner (schema) id for which synonyms are defined.
A schema has a maximum length of 128 characters.

Table Name>
Used to specify a filter on the table or view name for which synonyms are defined.
A DB2 synonym name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the synonym entry. Values associated with this entry will be inserted in the parent panel input fields.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <F17> by default.

Select Distinct Type

The Select Distinct Type panels (ZZS2SDTx) may be used to select a user defined distinct type form within the current DB2 server (sub-system.)

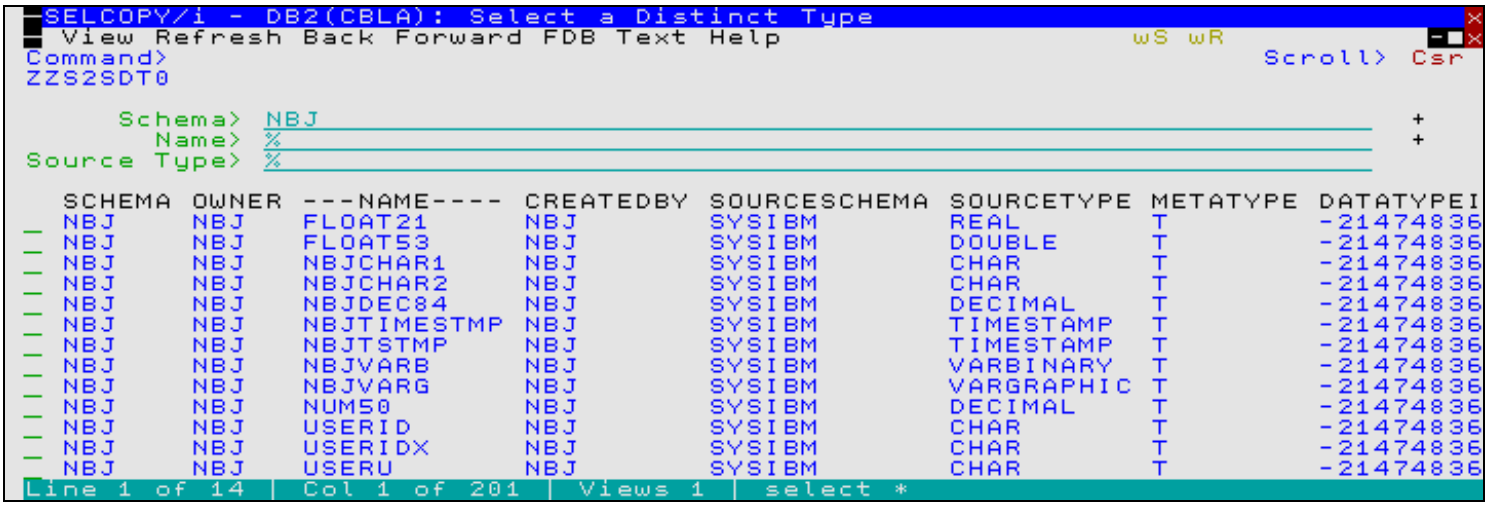


Figure 282. DB2: Select a Distinct Type.

Variations of this panel exist as follow:

`Select Distinct Type (ZZS2SDT0)`
Used by DB2 column definition sub-panel views to select an existing user defined distinct type for a column assigned a data type of DISTINCT.

This distinct type selection panel omits distinct type entries that have an invalid encoding scheme for the current DB2 table definition.

`Select Distinct Type for Identity Column (ZZS2SDT1)`
Used by DB2 column definition sub-panel views to select an existing user defined distinct type for an identity column assigned a data type of DISTINCT.

This distinct type selection panel omits distinct type entries that are not of one of the following built-in source types:

- ◊ SMALLINT
- ◊ INTEGER
- ◊ BIGINT
- ◊ DECIMAL with a zero scale

`Select Distinct Type for Security Label Column (ZZS2SDT2)`
Used by DB2 column definition sub-panel views to select an existing user defined distinct type for a security label column assigned a data type of DISTINCT.

This distinct type selection panel omits distinct type entries that do not have the following criteria:

- ◊ Built-in source type CHARACTER.
- ◊ Length 8.
- ◊ Sub-type SBCS.
- ◊ Encoding scheme that matches that of the current DB2 table definition.

`Select Distinct Type for Row Change Timestamp Column (ZZS2SDT3)`
Used by DB2 column definition sub-panel views to select an existing user defined distinct type for a row change timestamp column assigned a data type of DISTINCT.

This distinct type selection panel omits distinct type entries that are not of Built-in source type TIMESTAMP.

`Select Distinct Type for ROWID Column (ZZS2SDT4)`
Used by DB2 column definition sub-panel views to select an existing user defined distinct type for a ROWID column assigned a data type of DISTINCT.

This distinct type selection panel omits distinct type entries that are not of Built-in source type ROWID.

The contents of at least one of the selection panel input fields is set by an equivalent field in the parent panel which has invoked this selection panel.

The selection panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of entries fetched from the DB2 catalog table SYSIBM.SYSDATATYPES. See IBM publication "*DB2 SQL Reference*", "*Appendix - DB2 Catalog Tables*" for details on the column values displayed in this table.

Panel Input Fields

Owner>
Used to specify a filter on distinct type creator (owner) authorisation ID.
A distinct type creator ID has a maximum length of 128 characters.

Name>
Used to specify a filter on distinct type name.
A distinct type name has a maximum length of 128 characters.

Source Type>
Used to specify a filter on the distinct type built-in source type.
Enter blank or an invalid built-in type to display a scrollable, selectable list of of valid built-in types.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the table entry. Values associated with this entry will be inserted in the parent panel input fields.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <PF2> by default.

Select Function

The Select Function panel (ZZS2SFU0) may be used to select a DB2 function which has been defined in the current DB2 server.

The contents of at least one of the selection panel input fields is set by an equivalent field in the parent panel which has invoked this selection panel.

Functions that are invalid for use by the parent panel are omitted from the selection list. e.g. Cast functions are omitted for drop function name selection.

The selection panel function Schema and Name input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of DB2 functions.

Panel Input Fields

Schema>
Used to specify a filter on function schema.
A schema has a maximum length of 128 characters.

Name>
Used to specify a filter on function name.
A DB2 function name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the function entry. Values associated with this entry will be inserted in the parent panel input fields.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <F17> by default.

Select Stored Procedure

The Select Stored Procedure panel (ZZS2SPR0) may be used to select a DB2 stored procedure which has been defined in the current DB2 server.

The contents of at least one of the selection panel input fields is set by an equivalent field in the parent panel which has invoked this selection panel.

Stored procedures that are invalid for use by the parent panel are omitted from the selection list. e.g. Native SQL procedures are omitted for drop stored procedure name selection.

The selection panel stored procedure Schema and Name input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of DB2 stored procedures.

Panel Input Fields

Schema>
Used to specify a filter on stored procedure schema.
A schema has a maximum length of 128 characters.

Name>
Used to specify a filter on stored procedure name.
A DB2 stored procedure name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the stored procedure entry. Values associated with this entry will be inserted in the parent panel input fields.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <F17> by default.

Select Trigger

The Select Trigger panel (ZZS2STR0) may be used to select a DB2 trigger which has been defined in the current DB2 server.

The selection panel Trigger Schema and Name input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of DB2 triggers.

Panel Input Fields

Schema>
Used to specify a filter on trigger schema.
A schema has a maximum length of 128 characters.

Name>
Used to specify a filter on trigger name.
A DB2 trigger name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the trigger entry. Values associated with this entry will be inserted in the parent panel input fields.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <F17> by default.

Select Role

The Select Role panel (ZZS2SROx) may be used to select a role defined in the current DB2 server (sub-system).

The contents of at least one of the selection panel input fields is set by an equivalent field in the parent panel which has invoked this selection panel.

The selection panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of role entries fetched from the DB2 catalog table SYSIBM.SYSROLES. See IBM publication "*DB2 SQL Reference*", "*Appendix - DB2 Catalog Tables*" for details on the column values displayed in this table.

Panel Input Fields

Name>

Used to specify a filter on the role name.
A role name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the role entry. Values associated with this entry will be inserted in the parent panel input fields.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <PF17> by default.

Select Trusted Context

The Select Trusted Context panel (ZZS2STCx) may be used to select a trusted context defined in the current DB2 server (sub-system).

The contents of at least one of the selection panel input fields is set by an equivalent field in the parent panel which has invoked this selection panel.

The selection panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of trusted context entries fetched from the DB2 catalog table SYSIBM.SYSCONTEXT. See IBM publication "*DB2 SQL Reference*", "*Appendix - DB2 Catalog Tables*" for details on the column values displayed in this table.

Panel Input Fields

Definer>
Used to specify a filter on the authorisation id or role that defined the trusted context.
An authorisation id and role name have a maximum length of 128 characters.

Name>
Used to specify a filter on the trusted context name.
A trusted context name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the trusted context entry. Values associated with this entry will be inserted in the parent panel input fields.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <PF17> by default.

Select Package

The Select Package panel (ZZS2SPKx) may be used to select a package defined in the current DB2 server (sub-system) via BIND command.

The contents of at least one of the selection panel input fields is set by an equivalent field in the parent panel which has invoked this selection panel.

Packages that are invalid for use by the parent panel are omitted from the selection list. e.g. Packages created by CREATE TRIGGER and CREATE PROCEDURE are omitted for drop package selection.

The selection panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of package entries fetched from the DB2 catalog table SYSIBM.SYSPACKAGE. See IBM publication "*DB2 SQL Reference*", "*Appendix - DB2 Catalog Tables*" for details on the column values displayed in this table.

Panel Input Fields

Collection-id>
Used to specify a filter on package collection name.
A package collection name has a maximum length of 128 characters.

Name>
Used to specify a filter on the package name.
A package name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the package entry. Values associated with this entry will be inserted in the parent panel input fields.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <PF17> by default.

Select Unique Key Constraint

The Select Unique Key Constraint panel (ZZS2SCNx) may be used to select a primary or unique key constraint defined in the current DB2 server (sub-system.)

This panel is required for selection of valid parent table key columns for use in a referental constraint (parent/foreign key) relationship which requires that the parent key columns constitute a unique key on the parent table.

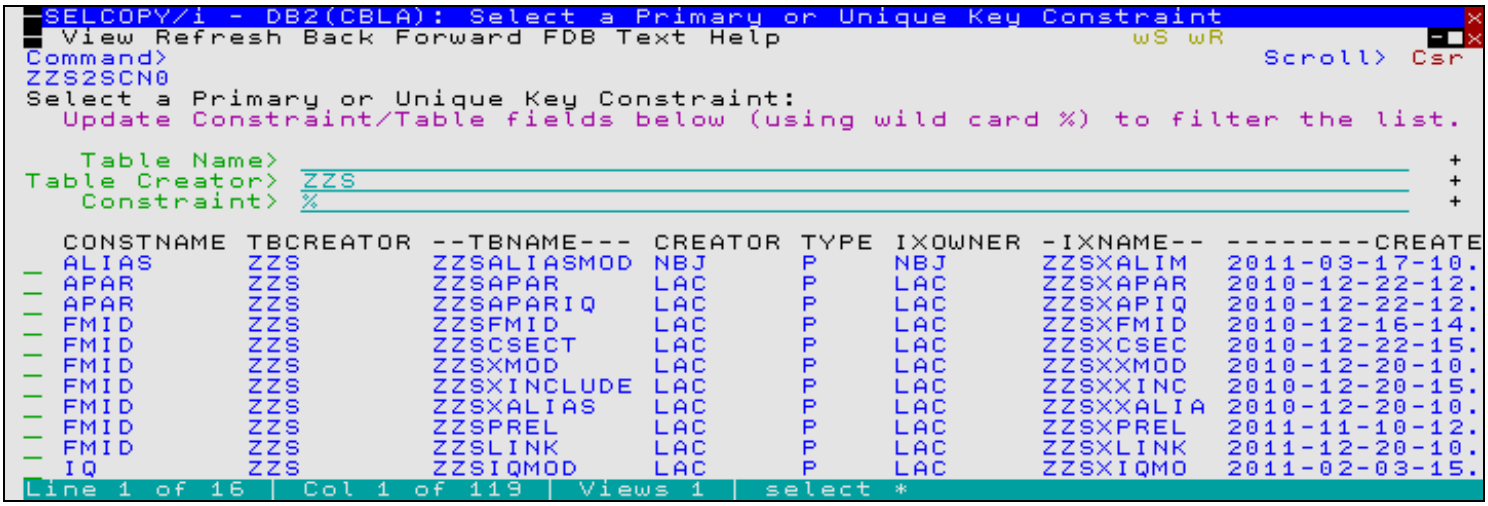


Figure 289. DB2: Select a Primary or Unique Key Constraint.

The contents of at least one of the selection panel input fields is set by an equivalent field in the parent panel which has invoked this selection panel.

Unique Key Constraints that are invalid for use as a referential constraint parent key are omitted from the selection list. These are unique constraints belonging to DB2 catalog, history and global temporary tables.

The selection panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of primary or unique key constraint entries fetched from the DB2 catalog table SYSIBM.SYSTABCONST. See IBM publication "*DB2 SQL Reference*", "*Appendix - DB2 Catalog Tables*" for details on the column values displayed in this table.

To view a list of the DB2 table columns assigned to any of the listed constraints, enter line command "C" to display the **Primary/Unique Key Constraint Columns** sub-panel (ZZS2LCC0). This sub-panel may be displayed for informational purposes only, prior to selection of a unique constraint.

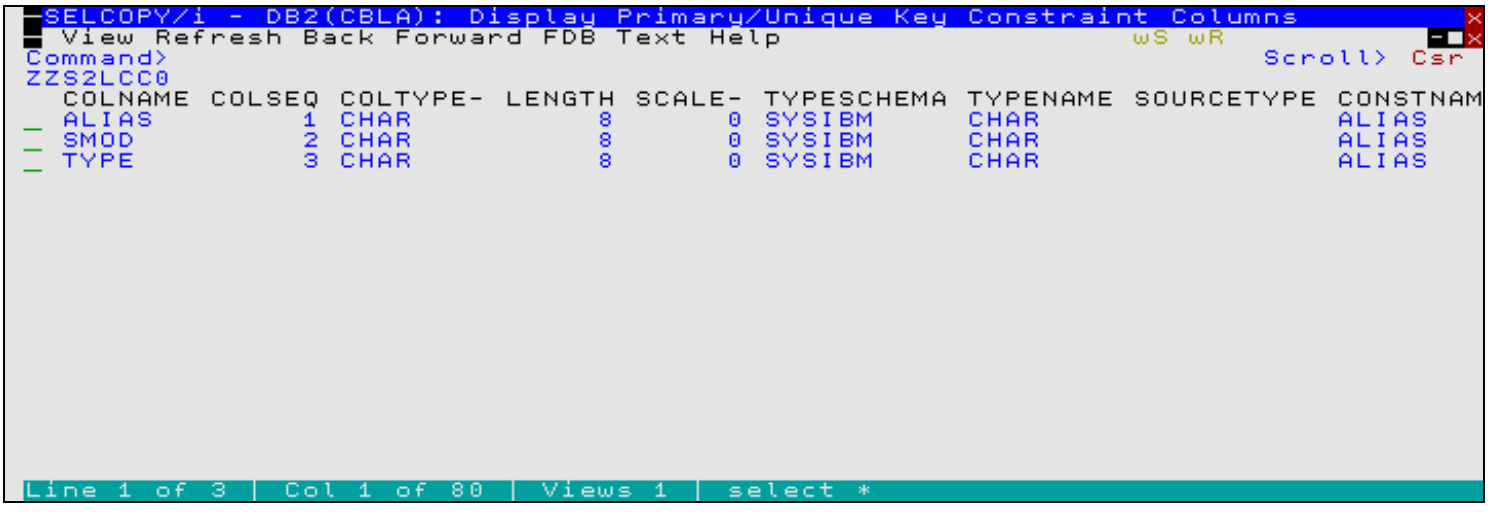


Figure 290. DB2: Display Primary/Unique Key Constraint Columns.

Panel Input Fields

Table Name>
Used to specify a filter on the name of the table to which the primary or unique key constraint is defined. A table name has a maximum length of 128 characters.

Table Creator>
Used to specify a filter on the creator ID of the table to which the primary or unique key constraint is defined.
A table creator ID has a maximum length of 128 characters.

Constraint>
Used to specify a filter on the name of the primary or unique key constraint.
A constraint name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
C	Display the **Primary/Unique Key Constraint Columns** sub-panel.
S	Select the primary or unique key constraint entry. Values associated with this entry will be inserted in the parent panel input fields.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <PF2> by default.

Select Column

The Select Column panel (ZZS2SCnn) may be used to select a column from the list of columns belonging to the current DB2 table definition.

Variations of this panel exist as follow:

Select Foreign Key Column (ZZS2SC00)
: Used by create DB2 table panel views to select a foreign key column in a referental constraint (parent/foreign key) relationship

  This column selection panel filters entries to display only columns that are compatible with the parent key column to which it applies.

Select XML Column (ZZS2SC02)
: Used by create DB2 index panel views to select a column of data type XML from the selected DB2 table.

Select Index Key Column (ZZS2SC03)
: Used by create DB2 index panel views to select a column from the selected DB2 table to be used as an index key column.

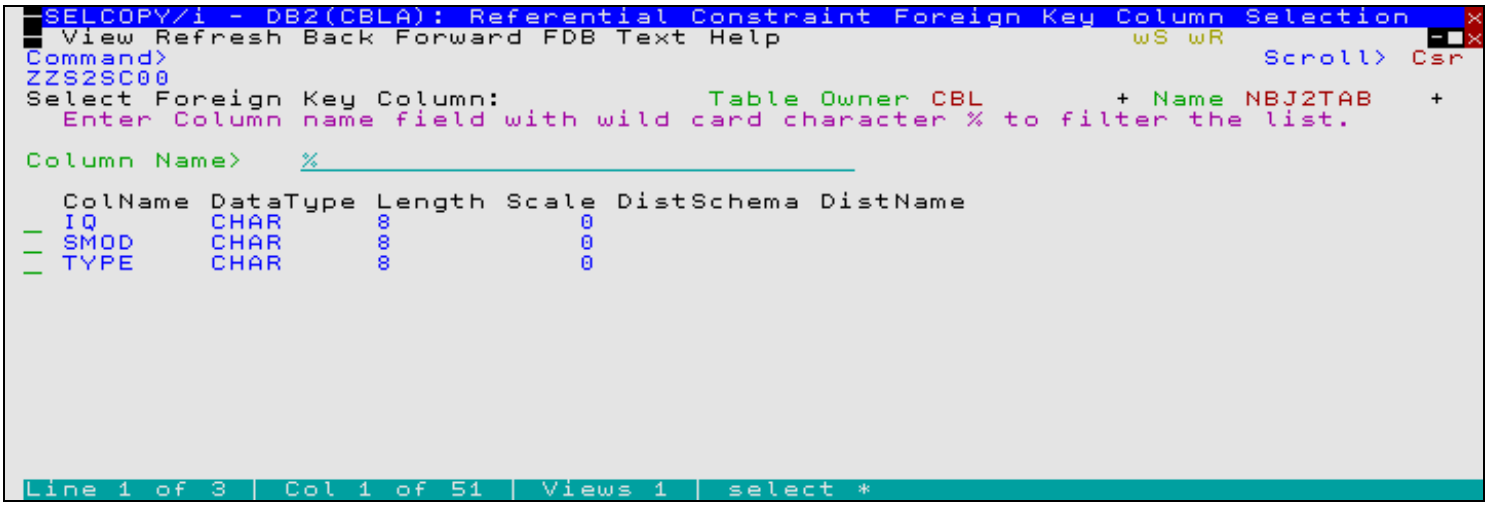


Figure 291. DB2: Select a Foreign Key Column.

The contents of the column name input field within the parent panel will be updated with the name of the selected DB2 table column.

The selection panel Column Name input field supports standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of column entries.

Panel Input Fields

Table Owner:
: A non-enterable field displaying the DB2 table owner id.

Name:
: A non-enterable field displaying the DB2 table name.

Column Name>
: Used to specify a filter on DB2 table column name.
  A DB2 table column name has a maximum length of 30 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the column entry.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <F17> by default.

Select XML NameSpace

The Select XML NameSpace panel (ZZS2SXNn) may be used to select an XML Namespace URI from a list of URIs which have already been defined in the current DB2 server.

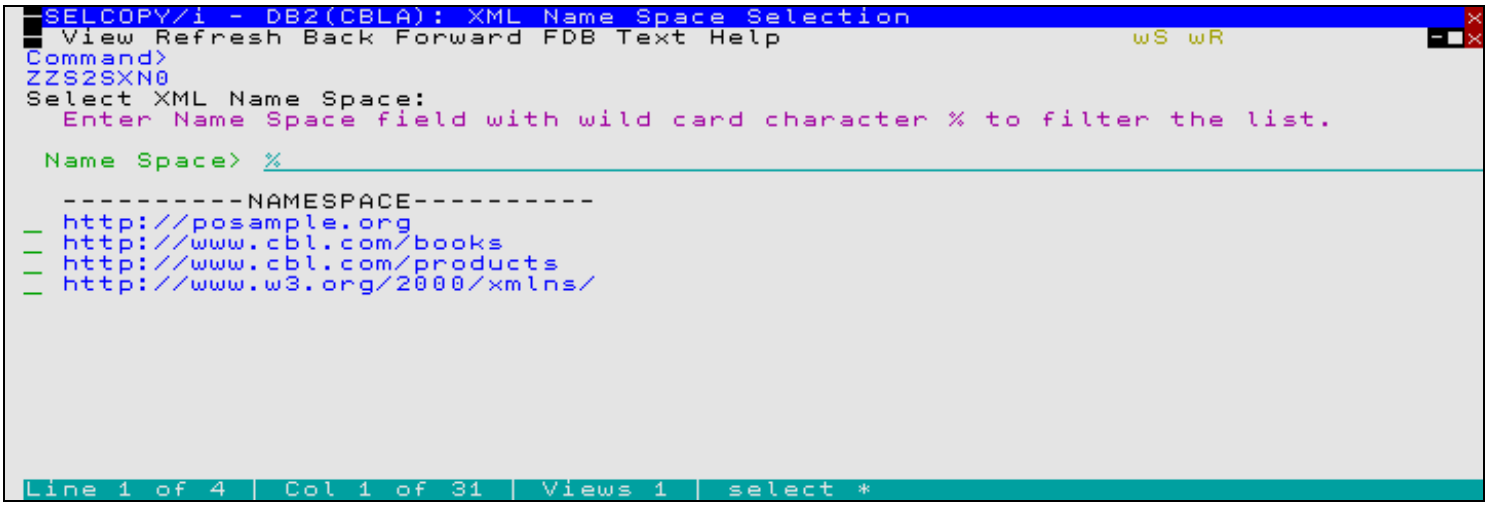


Figure 292. DB2: Select XML Name Space.

The selection panel Name Space input field supports standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of Name Space URIs.

Panel Input Fields

`Name Space>`
Used to specify a filter on XML Name space URI.
A DB2 XML string has a maximum length of 1000 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the URI entry.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <F17> by default.

Select Server Location

The Select Server Location panel (ZZS2SLO0) may be used to select a DB2 server location which has been defined in the current DB2 server.

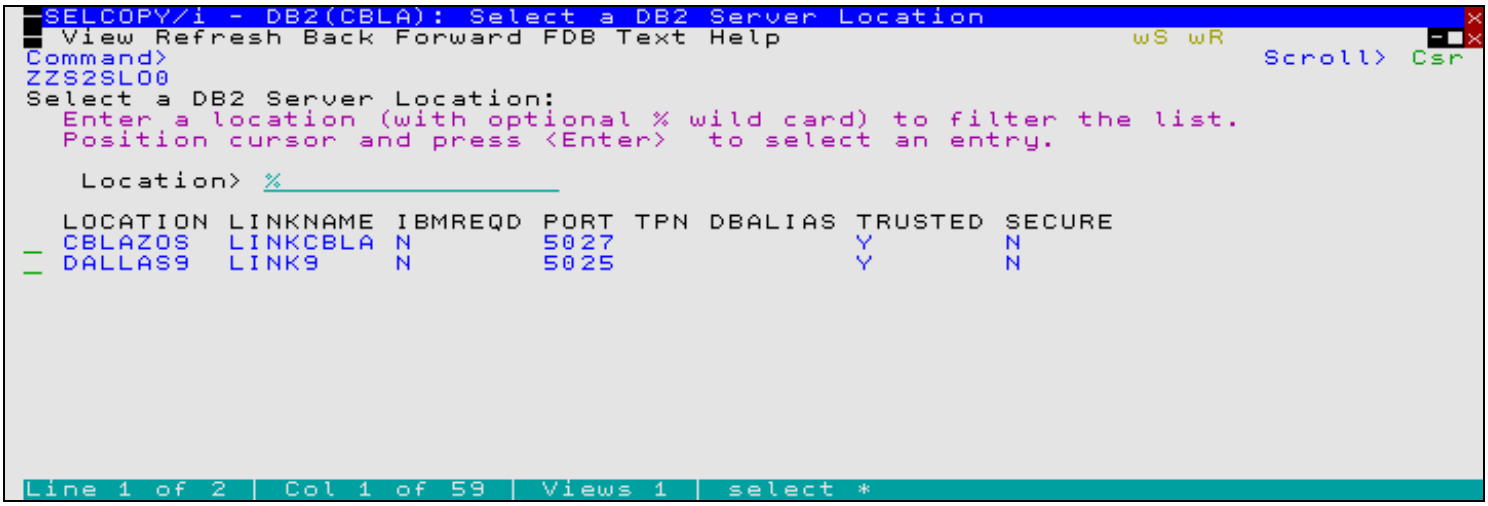


Figure 293. DB2: Select Server Location.

The selection panel Location input field supports standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of DB2 Locations.

Panel Input Fields

Location>
Used to specify a filter on location.
A DB2 location has a maximum length of 16 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the URI entry.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <F17> by default.

Select Catalog Alias

The Select Catalog Alias panel (ZZSGSHLQ) may be used to select an ICF catalog in which DB2 table and index spaces may be cataloged.

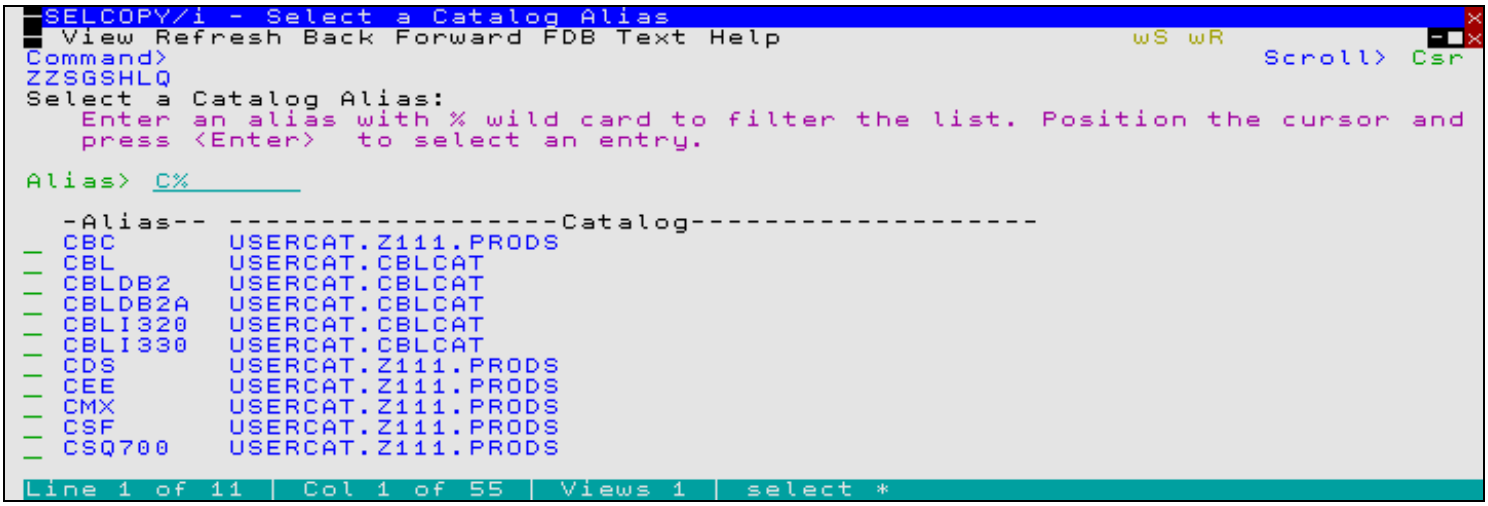


Figure 294. DB2: Select Catalog Alias.

The selection panel Alias input field supports standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of catalog aliases.

Panel Input Fields

Alias>
Used to specify a filter on catalog aliases.
A single qualifier alias has a maximum length of 8 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the URI entry.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <F17> by default.

Select Sequence

The Select Sequence panel (ZZS2SSQ0) may be used to select a DB2 sequence which has been defined in the current DB2 server.

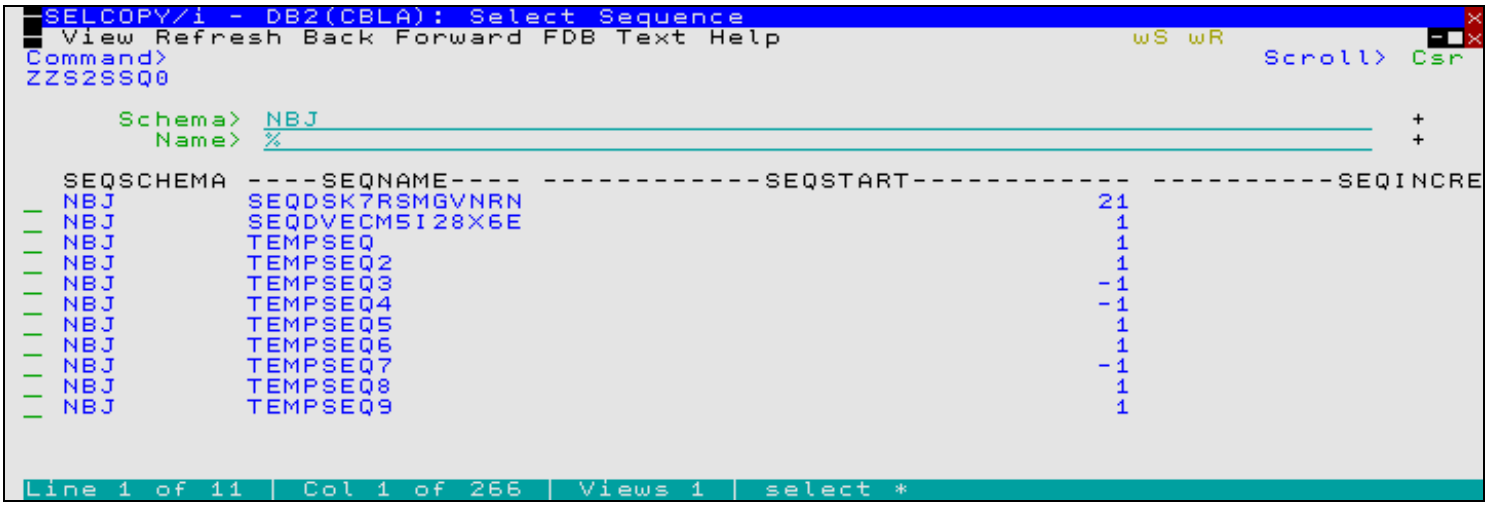


Figure 295. DB2: Select Sequence.

The selection panel Sequence Schema and Name input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be ammended to re-apply the filter and so refresh the display of DB2 sequences.

Panel Input Fields

Schema>
Used to specify a filter on sequence schema.
A schema has a maximum length of 128 characters.

Name>
Used to specify a filter on sequence name.
A DB2 sequence name has a maximum length of 128 characters.

Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command S.
S	Select the URI entry.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <F17> by default.

Audit Trail Functions

Audit Trail Functions Panel

The List DB2 Object panel (ZZS2AUD0) is an interactive panel window, opened on selection of option 8. in the DB2 Primary options menu.

This panel includes options relating to management and display of SELCOPY/i DB2 audit log data sets.

SELCOPY/i DB2 supports the generation of audit log data sets that record all SQL activity that has occurred during a DB2 connection. Unless logging is deactivated, a single audit log data set is automatically allocated for each DB2 subsystem connection. All SQL statements executed using this subsystem connection will be logged in the audit file.

Except for DB2 edit, which maintains a separate log file for each table edited, a DB2 connection and log file will be generated once for each execution of the DB2 primary option menu panel.

Users may choose to maintain a log data set via a flag in the DB2 primary option menu panel or, for table edit, a flag in the Edit Object panel or EDIT line command. Data set options, used by SELCOPY/i to allocate a new log data set, may be customised via the Audit Log Dataset Options panel. Other Audit panels provide facilities for listing and printing Audit data sets.

Menu Bar Items

`File`
The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Help`
Open the general help for the Audit Trail Functions option menu panel.

Options

1 Audit Log Dataset Options
2 Print Audit Report
3 List Audit Datasets

Audit Log Dataset Options

The Audit Log Allocation panel (ZZS2AUDS) is an interactive panel window, opened on selection of option 1. in the DB2 Audit Trail Functions options menu.

This panel allows the user to configure data set options that are subsequently used by SELCOPY/i when allocating new DB2 audit log data sets for the user. Note that SELCOPY/i DB2 logs are RECFM=VB physical sequential data sets.

Options in this panel should be customised so that log data sets comply with your system standards.

Panel Input Fields

`High Level Qualifier:`
Specifies the data set name high level qualifier prefix to be used. Mutually exclusive options are as follow:

- ◊ **Use Installation Default**
  Use the SELCOPYi installation default value as assigned to the INI variable, SYSTEM.UserDSNPrefix. If this variable is unset, the TSO prefix is used, otherwise the user's TSO (or SELCOPY/i VTAM) logon id.
- ◊ **Use TSO Prefix**
  Use the TSO PREFIX value as set by the user's profile.
- ◊ **Use User ID**
  Use the user's TSO (or SELCOPY/i VTAM) logon id.
- ◊ **Use Specified HLQ**
  Use the HLQ specified by field entry HLQ>

`HLQ>`
Applicable only if option Use Specified HLQ is selected. this field names the high level qualifier prefix to be used, up to a maximum length of 30 characters.

Device Type:
The device or type of device on which the log data set should be allocated.

Unit>
Specifies the UNIT device number, device type or esoteric group name.
Note that no UNIT parameter is required if the log data set is SMS managed. Specify a STORCLAS or let an automated class selection (ACS) routine select a storage class for the data set.

Allocation Unit:
Identifies the SPACE unit of allocation. Mutually exclusive options are as follow:

- ◊ **Cylinders**
  Requests that the space be allocated in DASD cylinders.
- ◊ **Tracks**
  Requests that the space be allocated in DASD tracks.
- ◊ **Blocks**
  Requests that the space be allocated in blocks.

Allocation Size:
Fields relating to the number of SPACE allocation units to allocate.

Primary>
Primary quantity of allocated units.

Secondary>
Secondary quantity of allocated units.

SMS Classes:
Fields relating to SMS data set management.

Data Class>
SMS Data Class to be used. Specify a Data Class if one is not automatically selected via an ACS routine.

Storage Class>
SMS Storage Class to be used. Specify a Storage Class if one is not automatically selected via an ACS routine.

Management Class>
SMS Management Class to be used. Specify a Management Class if one is not automatically selected via an ACS routine.

Print Audit Report

The Audit Log Report panel (ZZS2AUDP) is an interactive panel window, opened on selection of option 2. in the DB2 Audit Trail Functions options menu.

This panel allows the user to invoke the AUDPRINT command to display print output of a selected SELCOPYi DB2 audit log. Printing an audit log, first processes the formatted records of the audit log file to generate a printable report.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to display the report in a temporary text edit view.

Menu Bar Items

File
The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Run
Execute AUDPRINT in the SELCOPYi foreground to formats the audit log records and display the printed report in a temporary text edit view.
Hitting <Enter> will perform the same action.

Command
Opens a text edit view for a temporary data set containing the AUDPRINT command syntax generated for the selected DB2 audit log data set. The command text is in a format suitable for execution by positioning the cursor on the first line of the text and hitting <F16>. (i.e. using the ACTION facility.)

JCL
Opens a text edit view for a temporary data set containing JCL that runs program SELCOPY in batch to format the log records and write the generated report to SYSPRINT.

Help
Open the general help for the DB2 Audit Log Report panel.

Panel Input Fields

`Audit DSN:`
Specifies the DSN of the SELCOPYi DB2 audit log file to print.

The format of a log file DSN is *prefix*.ZZSX.*ssn*.Dyyyyddd.Thhmmsss.AUD and, by default, this field entry displays the DSN of the last log file created by SELCOPYi for the user.

List Audit Datasets

The List Audit Datasets window is simply a List Catalog Entries window with the the default DB2 audit log DSN mask passed to the **Entry>** field so that only the audit log entries are displayed.

Use the "AP" prefix command to open the Print Audit Report panel to display a printable report of any log file entry in this list.

Compare DB2 Tables Panels

Compare DB2 Base/Results Tables - New Table details and options

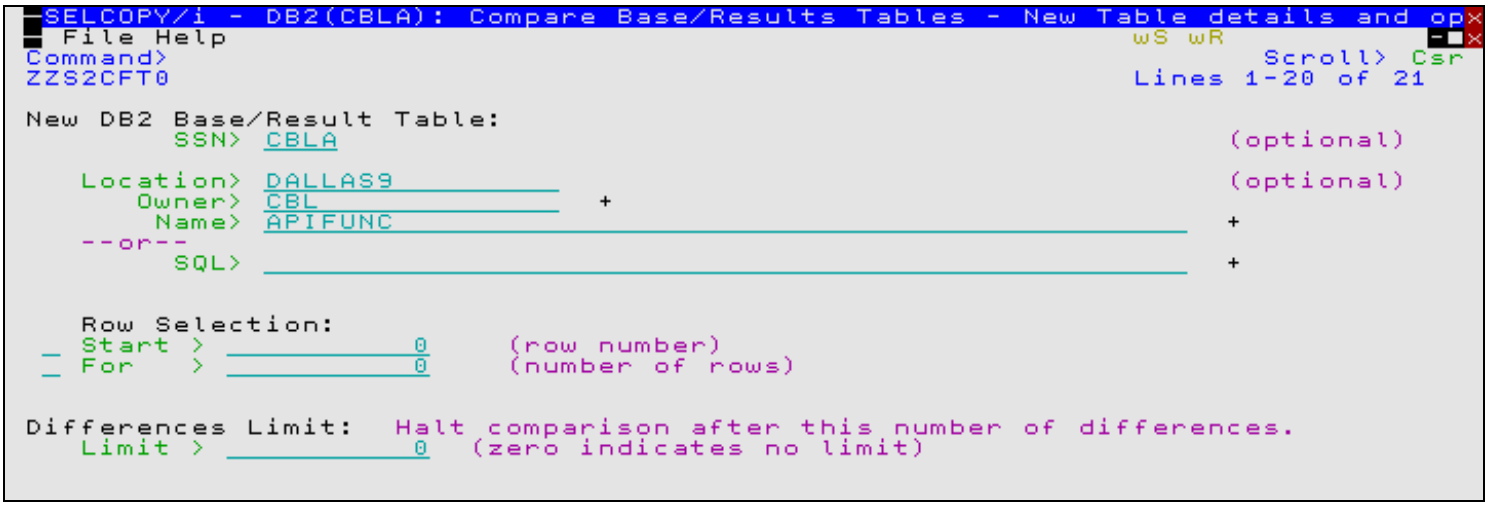


Figure 296. SELCOPY/i - Compare DB2 Base/Results Tables - New table details and options.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will will proceed to the next Compare DB2 Tables panel view.

Panel Input Fields

New DB2 Base/Result Table:

SSN>
The DB2 subsytem on which the NEW Base or Results table to be compared is located.

If left blank then the current subsystem (identified in the window title bar) will be used. If a "select" command is supplied in the **SQL>** field, then the **Location>**, **Owner>** and **Name>** fields will be ignored.

Location>
The server location of the NEW Base or Results table to be compared. If left blank, the local server for the connected DB2 sub-system is used.

Owner>
The owner (schema) of the NEW Base or Results table to be compared.
A table selection list will be provided if wildcards (*) are supplied.

Name>
The name of the NEW Base or Results table to be compared.
A table selection list will be provided if wildcards (*) are supplied.

SQL>
An SQL "select" commaned used to specify the NEW Results table to be compared e.g.

```
select TRACK_NUM,NAME from U123.SELCTRN_TRACK where NAME like '%(Live)%'
```

If supplied then the **Location>**, **Owner>** and **Name>** fields will be ignored.

Row Selection:

Start>
Defines the row number in the NEW table at which rows will start to be compared.

A row number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**.

This field corresponds to the COMPFILE parameter NSTARTREC.
Default is 1.

For>
Specifies the maximum number of rows to be compared from the NEW table. The compare operation stops if this threshold is encountered even if the equivalent threshold for OLD table rows has not been reached.

This field corresponds to the COMPFILE parameter NFOR.
Default is 0 (all rows).

Differences Limit:

Limit>
Use this option in order terminate the compare process as soon as the specified number of row mismatches has been encountered.

Specifying zero or blank indicates that no limit is placed, and therefore the whole of each results table (or row selection range) is processed.

This field corresponds to the COMPFILE parameter LIMIT.
Default is 0 (no limit).

Compare DB2 Base/Results Tables - Old Table details and options

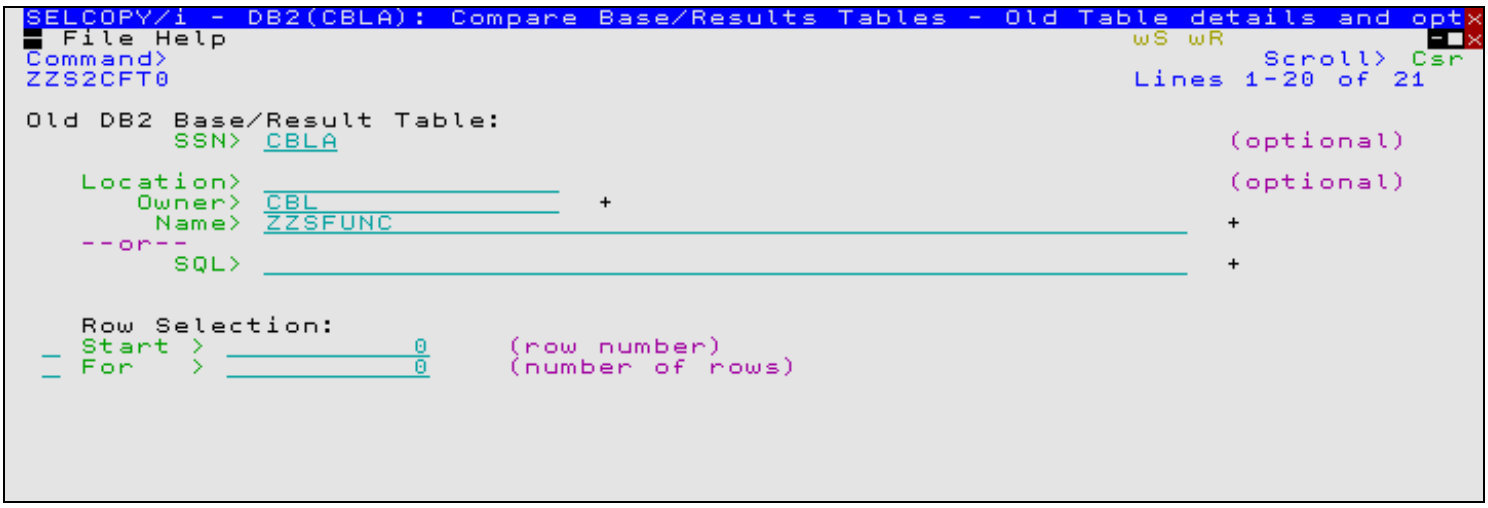


Figure 297. SELCOPY/i - Compare DB2 Base/Results Tables - Old table details and options.

The **Compare DB2 Base/Results Tables - Old table details and options** panel view is displayed following the Compare DB2 Base/Results Tables - New table details and options panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will will proceed to the next Compare DB2 Tables panel view.

Panel Input Fields

Old DB2 Base/Result Table:

SSN>
The DB2 subsytem on which the OLD Base or Results table to be compared is located.

If left blank then the current subsystem (identified in the window title bar) will be used. If a "select" command is supplied in the **SQL>** field, then the **Location>**, **Owner>** and **Name>** fields will be ignored.

Location>
The server location of the OLD Base or Results table to be compared. If left blank, the local server for the connected DB2 sub-system is used.

Owner>
The owner (schema) of the OLD Base or Results table to be compared.
A table selection list will be provided if wildcards (*) are supplied.

Name>
The name of the OLD Base or Results table to be compared.
A table selection list will be provided if wildcards (*) are supplied.

SQL>
An SQL "select" commaned used to specify the OLD Results table to be compared e.g.

```
select TRACK_NUM,NAME from U123.SELCTRN_TRACK where NAME like '%(Live)%'
```

If supplied then the **Location>**, **Owner>** and **Name>** fields will be ignored.

Row Selection:

Start>
Defines the row number in the OLD table at which rows will start to be compared.

A row number may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**.

This field corresponds to the COMPFILE parameter OSTARTREC.
Default is 1.

For>
Specifies the maximum number of rows to be compared from the OLD table. The compare operation stops if this threshold is encountered even if the equivalent threshold for NEW table rows has not been reached.

This field corresponds to the COMPFILE parameter OFOR.
Default is 0 (all rows).

Compare DB2 Base/Results Tables - Re-synchronisation options

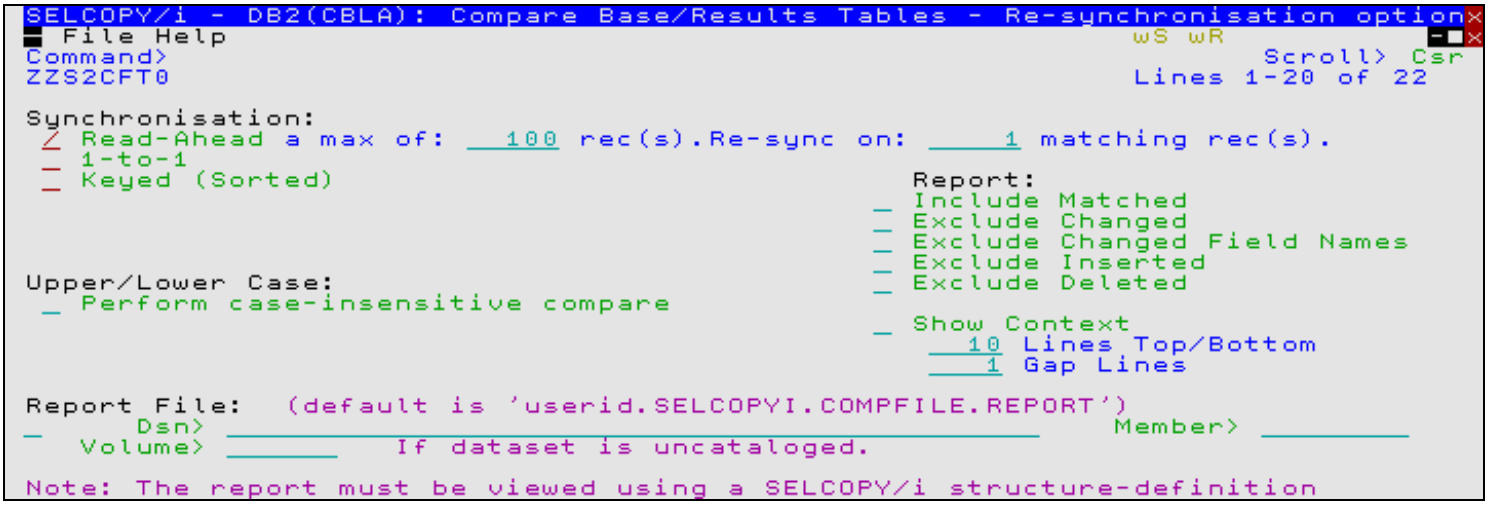


Figure 298. SELCOPY/i - Compare DB2 Base/Results Tables - Re-synchronisation options.

The **Compare Tables: Re-synchronisation options** panel view is displayed following the Compare DB2 Base/Results Tables - Old table details and options panel view.

Pressing the <Enter> key or, if configured, double-clicking the left mouse button will will do the following:

- If Read-Ahead or 1-to-1 synchronisation is selected, the next Compare DB2 Tables panel view will be displayed.
- If Keyed (Sorted) synchronisation is selected, then panel "Compare Base/Results Tables - Specify Key Columns:" is displayed.

Panel Input Fields

Read-Ahead ...
Select this option to use read-ahead row synchronisation.

Read-ahead synchronisation technique is suitable where the NEW and OLD tables are predominantly comprised of equal rows, although some may have been changed, inserted or deleted.

When a row mismatch is detected, the compare tables utility will attempt to resynchronsise the current, mismatching rows by reading a specified number of rows, first from the OLD table then from the NEW table, in order to find a match on a specified number of consecutive rows. If successful, a resynchronised row pair may be established and the compare operation continued from these rows.

Records that have been skipped as a result of the read-ahead synchronisation are flagged as having been inserted or deleted as appropriate.

For a detailed description, see *"Read-Ahead Synchronisation"*.

This option corresponds to COMPFILE parameters SYNC READAHEAD.

a maximum of *RALimit* rec(s).
The maximum number of rows to read-ahead in each table when attempting to establish a synchronised row pair.

For efficiency, this value should be only one more than the maximum number of expected consecutive non-matching row pairs.

This field corresponds to the number *n1* in the COMPFILE parameters SYNC READAHEAD(*n1 n2*). Default value is 100.

Re-sync on *RAMatch* matching rec(s).
The number of consecutive matching row pairs that are required in order to establish a synchronised row pair. If satisfied, the first matching row pair is identified as a synchronised row pair.

This field corresponds to the number *n2* in the COMPFILE parameters SYNC READAHEAD(*n1 n2*). Default value is 1.

1-to-1
Select this option to use 1-TO-1 row synchronisation.

For 1-TO-1 synchronisation the tables are assumed to contain corresponding rows, so no attempt is made to resynchronise.

For a detailed description, see *"1-TO-1 Synchronisation"*.

This option corresponds to COMPFILE parameters SYNC 1TO1.

Keyed (Sorted)
Select this option to use Sorted Key Synchronisation.

Keyed (Sorted) synchronisation type is suitable where the NEW and OLD tables are sorted based on one or more key column within each row.

If this option is selected then a sub-panel will be opened, prompting the user to specify the required key columns, either directly or via a defined table index name.

An OLD and NEW table row may then be identified as a synchronised row pair when there is an exact match in all key columns of the row.

Where data mismatches occur in other parts of the rows comprising the synchronised row pair, then the row is flagged as having been **changed**.

Records that are not established as being one of a synchronised row pair are reported as having been **inserted** or **deleted** as appropriate.

In general, synchronisation occurs by reading rows from the table with the lower key data until a row with matching or higher key data is read. Intervening rows are then treated as having been inserted or deleted.

For a detailed description, see *"Key Synchronisation"*.

This option corresponds to COMPFILE parameters SYNC KEY.

Include Matched
Select this option to include matching rows in the output report table.

This option corresponds to COMPFILE parameter INCMATCHED.

Exclude Changed
Select this option to exclude changed rows from the output report table.

This option corresponds to COMPFILE parameter EXCHANGED.

Exclude Changed Column Values
Select this option to indicate that column values for any changed rows are not to be included in the output report.

Note that specifying this option may result in a significant performance improvement since the process of comparing column-by-column is terminated at the first mismatch in each row.

This option corresponds to COMPFILE parameter EXFIELDCHANGED.

Exclude Inserted
Select this option to exclude inserted rows from the output report table.

This option corresponds to COMPFILE parameter EXINSERTED.

Exclude Inserted
Select this option to exclude deleted rows from the output report table.

This option corresponds to COMPFILE parameter EXDELETED.

Include Column Values
Select this option to indicate that column values for any matched, inserted and/or deleted rows are eligible to be included in the output report.

For example, column values will be displayed for inserted rows provided that the "Exclude Inserted" option is **not selected.**

This option corresponds to COMPFILE parameter INCFIELDS.

Show Context
Select this option to include a specified number of (possibly matching) rows immediately before and after each detected difference in order to provide context without including all matching rows, which for large tables is likely to be prohibitive.

Note that this option will be ignored if "Include Matched" is already selected.

nn Lines Top/Bottom
The number of context lines to be displayed before and after each difference.

A value of zero may be specified if gap lines are required without context.

This option corresponds to COMPFILE parameter "CONTEXT nn".

nn Gap Lines
The number of "Gap" lines to display in order to separate each difference context block.

This option corresponds to COMPFILE parameter "GAP nn".

Perform case-insensitive compare
Select this option to perform a case insensitive compare. Character (AN) fields will be translated to upper case before comparison.

This option corresponds to COMPFILE parameter CASEINSENSITIVE (synonym CASEIGNORE).

Report File:
Dsn>
Member>
Volume>
If the Report File option field is selected, then these fields identify the name of the file to which the compare utility report records will be written. Dataset names must be fully qualified, quotes being unnecessary but permitted.

The report is a structured data file designed to be browsed (not printed) using a Data Edit structure definition object (SDO), which will also be generated by the compare files utility.

The associated SDO fileid is constructed simply by adding **.SDO** to the report fileid. Therefore, the DSN of the report table is restricted to 40 bytes in length.
Report output to an HFS dataset is not currently supported.

If the report file and/or the SDO file do not already exist, then they will automatically be allocated by the compare utility, relying on SMS ACS to select a suitable storage group of eligable DASD volumes. The report file is allocated using DCB geometry RECFM=VB, LRECL=32756, BLKSIZE=0 and a space allocation of TRACKS(150,75). The SDO is allocated using DCB geometry RECFM=VB, LRECL=16380, BLKSIZE=0 and a space allocation of TRACKS(2,2).

If this option is not specified, *fileid* defaults to "*user*.SELCOPYI.COMPFILE.REPORT" with SDO fileid "*user*.SELCOPYI.COMPFILE.REPORT.SDO".

A selectable list of files will be presented if wildcards are entered, or dataset is a PDS/PDSE library and member is left blank.

These fields correspond to COMPFILE parameter REPORT.

Type> `Default | FMT | TEXT`
This field corresponds to the COMPFILE option "LIST=".

A selection list will be displayed if the option entered is unrecognised or left blank. Choose from one of the following options:

FMT
The report generated is a **structured data file** designed to be (automatically) browsed (not printed) from within a SELCOPYi online session, using a structure definition (SDO) which is also generated automatically during execution of the compare.

TEXT
The report generated is a more traditional formatted plain text document, designed to be printed if necessary.

Default
FMT if running online, **TEXT** if running from JCL in batch.

Compare DB2 Base/Results Tables - Specify Key Columns

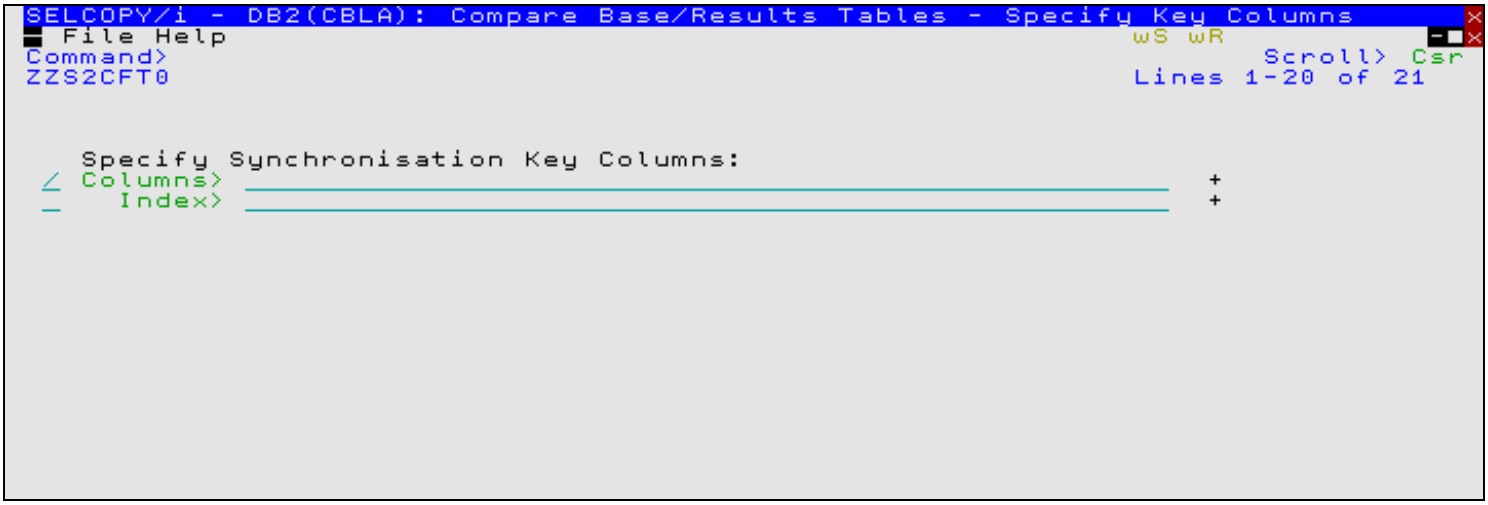


Figure 299. SELCOPY/i - Compare DB2 Base/Results Tables - Specify Key fields.

The **Compare DB2 Base/Results Tables - Specify Key Columns** panel view is displayed following the Compare DB2 Base/Results Tables - Re-synchronisation options panel view if syncronisation type **Keyed (Sorted)** is selected.

Keyed synchronsiation relies on the both NEW and OLD tables being supplied in **ascending** sequence on all key columns.

If either the NEW or OLD table is defined using explicit **SQL** then the user is responsible for ensuring the sort order of that table by supplying **ORDER BY** as part of the SQL "select" clause.

Otherwise, the sort order for table input is handled automatically, with the compare process ensuring that all input rows are sorted in **ascending** sequence on all key columns.

Panel Input Fields

Specify Synchronisation Key Columns:

Columns>
A list of comma separated field names that define the synchronisation key columns. e.g.

```
ALBUM_ID,DISC_NUMBER,TRACK_NUM,PERSISTENT_ID
```

This field corresponds to the COMPFILE parameter SYNC KEY.

Index>
The name of a DB2 index defined on the NEW table from which the key column names will be extracted.

A selectable list of index names will be provided if '?' (question mark) is entered or the field is left blank.

Note that the sort sequence will be treated as **ascending** for all columns even if they are defined as **descending** by the selected index.

This field corresponds to the COMPFILE parameter SYNC KEYINDEX.

Compare DB2 Base/Results Tables - Options / Action

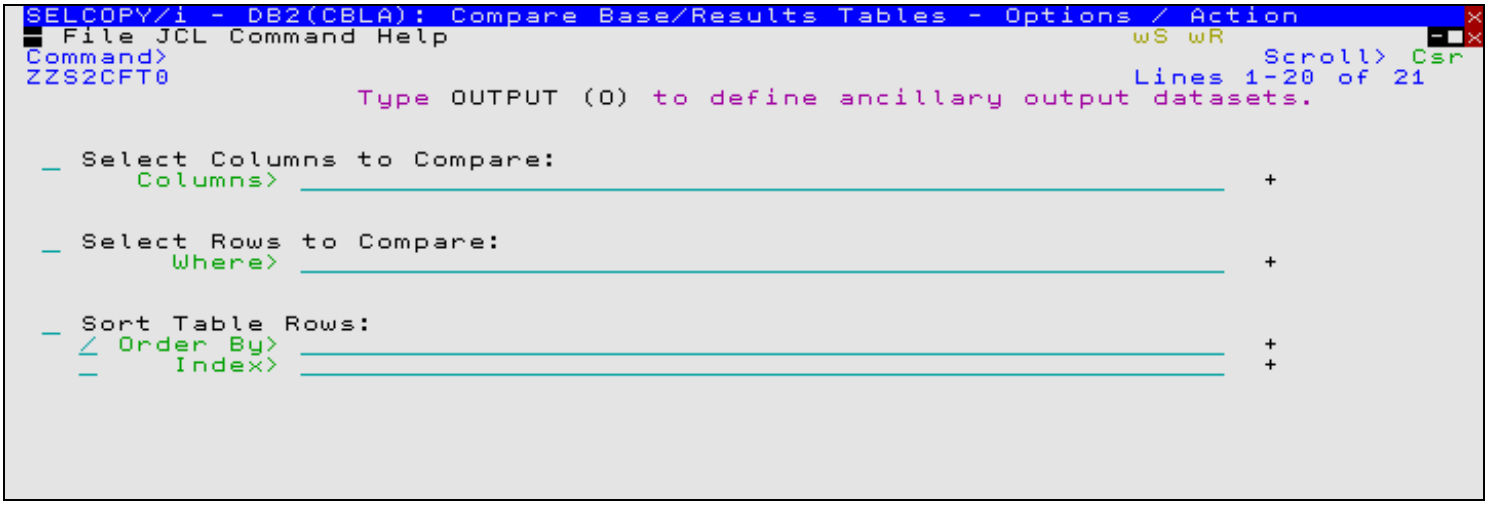


Figure 300. SELCOPY/i - Compare DB2 Base/Results Tables - Specify Key fields.

Panel Input Fields

Select Columns to Compare:

Columns>
A list of comma separated column names whose contents from the NEW table is to be compared with the contents of the corresponding column from the OLD table. e.g.

```
ALBUM_ID, DISC_NUMBER, TRACK_NUM, PERSISTENT_ID
```

If left blank then all columns that exist in both NEW and OLD tables will be compared. Any column specified that does not exist in the OLD table will be ignored and any column specified that does not exist in the NEW table will result in an error.

This field corresponds to the COMPFILE parameter SELECT.

Select Rows to Compare:

Where>
An SQL **"WHERE"** clause used to select rows from both NEW and OLD tables. e.g.

```
NAME like '%(Live)%'
```

This field corresponds to the COMPFILE parameter WHERE.

Order By>
An SQL **"ORDER BY"** clause used to define the sort order of both NEW and OLD tables.

```
ALBUM_ID desc, DISC_NUMBER, TRACK_NUM,PERSISTENT_ID
```

This field corresponds to the COMPFILE parameter SORT.

Index>
The name of a DB2 index defined on the NEW table from which the key column names will be extracted. A selectable list of index names will be provided if '?' (question mark) is entered or the field is left blank.

This field corresponds to the COMPFILE parameter SYNC SORTINDEX.

Compare DB2 Base/Results Tables - Ancillary Output Files

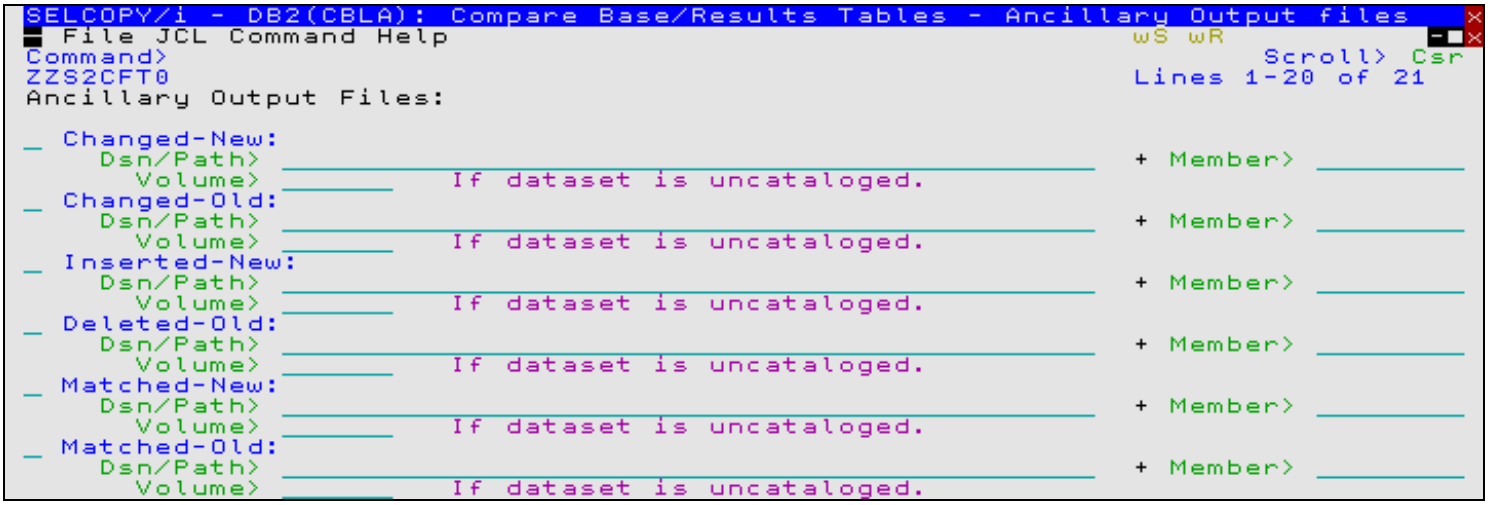


Figure 301. SELCOPY/i - Compare DB2 Base/Results Tables - Output Tables.

The **Compare DB2 Base/Results Tables - Ancillary Output Files** panel view is displayed if primary command **OUTPUT (O)** is issued from the Compare DB2 Base/Results Tables - Options / Action panel view.

This panel view identifies the output files to which a row from the NEW and/or OLD tables are to be copied, based on its flagged status (matched, changed, inserted or deleted). The output fileid may be an HFS table path, sequential data set or PDS/PDSE library member.

Data set names must be fully qualified, quotes being unnecessary but permitted.

A selectable list of files will be presented if wildcards are entered, or if a dataset is specified which is a PDS/PDSE library and the member field is left blank.

If a specified output file is non-HFS and does not already exist, then it will automatically be allocated by the compare utility, relying on SMS ACS to select a suitable storage group of eligable DASD volumes. The data set is allocated using DCB RECFM, LRECL and BLKSIZE geometry that best matches the NEW or OLD table as appropriate.

Panel Input Fields

Changed-New:
Dsn/Path>
Member>
Volume>

If the Changed-New option field is selected, then these fields identify the file-id of the file to which NEW table rows, flagged as having been changed (CN), are to be copied.

This file-id corresponds to the COMPFILE parameter WRITECN *cn_file-id*.

Changed-Old:
Dsn/Path>
Member>
Volume>

If the Changed-Old option field is selected, then these fields identify the file-id of the file to which OLD table rows, flagged as having been changed (CO), are to be copied.

This file-id corresponds to the COMPFILE parameter WRITECO *co_file-id*.

Inserted-New:
Dsn/Path>
Member>
Volume>

If the Inserted-New option field is selected, then these fields identify the file-id of the file to which NEW table rows, flagged as having been inserted (I), are to be copied.

This file-id corresponds to the COMPFILE parameter WRITEIN *in_file-id*.

Deleted-Old:
Dsn/Path>
Member>
Volume>

If the Deleted-Old option field is selected, then these fields identify the file-id of the file to which OLD table rows, flagged as having been deleted (D), are to be copied.

This file-id corresponds to the COMPFILE parameter WRITEDO *do_file-id*.

Matched-New:
Dsn/Path>
Member>
Volume>

If the Matched-New option field is selected, then these fields identify the file-id of the file to which NEW table rows, flagged as being matched, are to be copied.

This file-id corresponds to the COMPFILE parameter WRITEMN *mn_file-id*.

Matched-Old:
Dsn/Path>
Member>
Volume>

If the Matched-Old option field is selected, then these fields identify the file-id of the file to which OLD table rows, flagged as being matched, are to be copied.

This file-id corresponds to the COMPFILE parameter WRITEMO *mo_file-id*.

Rename DB2 Objects

Rename DB2 Objects Panel

The Rename DB2 Object sequence of panel views (ZZS2R001) generate an SQL SQL RENAME statement to rename the selected object type (Table or Index) from the current DB2 server. Note that the current DB2 server (subsystem name) is displayed in the panel window title bar.

These sequence of panel views are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select Rename, option 10, from the SELCOPYi DB2 primary option menu. (DB2 10)

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made in a the panel views. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL RENAME syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceding forwards again.

The first panel view allows the user to select the type of object to be renamed by entering the relevant option number or by positioning the cursor on the required option and pressing the <Enter> key or, if configured, double-clicking the left mouse button.

Menu Bar Items

File
The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Help
Open the general help for the Rename DB2 Objects option menu panel.

Options

1 Table
2 Index

Rename DB2 Table

A DB2 table object is to be renamed from the current DB2 server.

Renaming a table will also rename all aliases, synonyms, views indexes and privileges on that table; all referential constraints in which the table is a parent or dependent and, if implicitly created, the table space containing the table.

Panel Field Entries

Old Table Owner>
Identifies the owner (schema) of the DB2 table to be renamed from the current server. Maximum length of a table schema is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of tables eligible for rename.

Old Table Name>
Identifies the name of the DB2 table to be renamed from the current server. Maximum length of a table name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of tables eligible for rename.

New Table Name>
Identifies the new name of the DB2 table to be renamed.

No selection list is available for this field as it must specify a non-existant table name.

Rename DB2 Index

A DB2 index object is to be renamed from the current DB2 server.

Renaming an index will also rename the index space containing the index.

Panel Field Entries

Old Index Owner>
Identifies the owner (schema) of the DB2 index to be renamed from the current server. Maximum length of a index schema is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of indexes eligible for rename.

Old Index Name>
Identifies the name of the DB2 index to be renamed from the current server. Maximum length of a index name is 128 characters.

Enter wild card "*" or "%" in this field to select from a list of indexes eligible for rename.

New Index Name>
Identifies the new name of the DB2 index to be renamed.

No selection list is available for this field as it must specify a non-existant index name.

Generate SQL

The Generate SQL panel view is the last view in the sequence, displayed following any of the Rename object panel views.

This view allows the user to select how the generated SQL RENAME statement is to be implemented.

Panel Field Entries

Rename Object:
A non-enterable field displaying the type of object being renamed.

Action:
Enter "/" (slash) or any non-blank character to select the the action to be performed with the generated SQL statement on completion of the rename object sequence of panel views.

For all the actions below, except Execute immediately, the SQL will be displayed in an edit view. The appropriate command or facility may be issued by the user to subsequently execute the generated SQL statement.

Copy to a file
Copy the generated SQL statement to the output file specified by the SQL Output File fields below.

The SQL statement may subsequently be executed using the EXECSQL primary command.

Display an in-storage copy
Copy the generated SQL statement to an in-storage output file with a temporary DSN.

The SQL statement may subsequently be executed using the EXECSQL primary command.

Display as an executable line command
Copy the generated SQL statement to an in-storage output file in a format suitable for subsequent execution using the ACTION (default >F16>) facility.

Display within generated batch JCL
Copy the generated SQL statement to an in-storage output file and enclose it within JCL which executes the DSNTIAD load module.

The SQL statement may subsequently be executed using the SUBMIT primary command.

Execute immediately
Opens the Execute SQL Statements panel and immediately executes the generated SQL statement to rename the DB2 object. DB2 SQL messages are also displayed in this panel.

If the Confirm foreground execution of DB2 object rename option was selected in the Rename DB2 Object Menu panel view, a confirmation pop-up window will be displayed before actioning the rename.

SQL Output File:
Applicable only if the Copy to file action has been selected.

Input fields which together identify a single output file (sequential data set, HFS file path or PDS/PDSE library member) to which the generated SQL statement will be copied. This output file may be a new or existing data set, HFS file or library

member.

DSN/Path>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

If a DSN is specified for a data set (of orgaisation PS or PO) that does not already exist, the Allocate NonVSAM data set dialog window will be opened to create the new output file.

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of a new or existing member within that library.

A selectable list of members will be presented if the entered value contains wildcard characters "*" (asterisk) or "%" (percent), or is blanked out.

Model Dsn>
Applicable only if the Copy to file action has been selected and **DSN/Path>** specifies a new data set or PDS/PDSE library name.

This field specifies the DSN of an existing sequential or PDS/PDSE library that will be used to model a new data set in the Allocate NonVSAM dialog window.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Append>
Applicable only if the Copy to file action has been selected.

Enter "/" (slash) or any non-blank character in this field to append the generated SQL statement to existing text in the output file. If not, existing text will be replaced by the SQl statement.

Create/Edit DB2 Structure (SDO)

Create/Edit DB2 Structure Panel

The DB2 Structure panel (ZZS2CSDO) assists the user in creating a structure (SDO) for a table that may be referenced when browsing or editing that table with SELCOPYi. While it's not necessary to specify an SDO when editing a table (the structure of the tables columns is known to DB2 itself) an external structure does allow the user to specify defaults such as:

- the columns selected, their order and columns widths
- a generated WHERE clause to select rows
- various DB2 options such Concurrency and Commit options.

These sequence of panel views are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select "Structure", option 11, from the SELCOPYi DB2 primary option menu. (DB2 11)

Menu Bar Items

File
The File rename-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Help
Open the general help for the Create/Edit DB2 Structure panel.

Panel Fields - Create/Edit DB2 Structure

DB2 Base Table:

SSN>
The DB2 subsytem on which the table to be mapped is located.

If left blank then the current subsystem (identified in the window title bar) will be used.

Location>
The server location of the table to be mapped. If left blank, the local server for the connected DB2 sub-system is used.

Owner>
The owner (schema) of the table to be mapped.
A table selection list will be provided if wildcards (*) are supplied.

Name>
The name of the table to be mapped.
A table selection list will be provided if wildcards (*) are supplied.

Structure File to Create/Edit:
Defines fields which together specify a structure file to create or edit. If an existing structure is specified then it will be loaded and its contents used to set all other panel input fields. This means that an existing structure may be edited, modified then saved over the original or saved as a new name.

Dsn>
Identifies the fully qualified data set name of a new or existing sequential data set or PDS/PDSE library. If not new, the dataset must containing an existing DB2 SDO structure.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must a specify a member name.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Title:
Short descriptive title for this structure, up to 30 characters in length.

Description:
Longer description for this structure, up to 124 characters in length.

Create/Edit DB2 Structure Options

The Create/Edit DB2 Structure Options panel view is displayed only if command **OPTION** is executed.

This panel view determines DB2 specific options used when accessing table data for edit and also options used specifically by SELCOPYi DB2 table edit.

Panel Fields - Create/Edit DB2 Structure Options

Load Options:

Skip Locked Rows
Ignored unless an isolation level of Cursor Stability (CS) or Read Stability (RS) is in effect, this option specifies that any selected rows that are already locked by another process should be skipped and not be included in the edit display. See "DB2 SQL Reference" for details on the SKIP LOCKED DATA clause.
Default is to allow display of locked rows whenever possible.

Execute Commit following Load
Perform a COMMIT following the initial load of rows to be edited, thus releasing all DB2 table locks performed during load of the data. This includes any explicit table locks applied via the LOCKTABLE parameter.
Default is not to perform a COMMIT following load of the table rows.

Miscellaneous Options:

Create Audit File
Open a new SELCOPYi DB2 audit data set to record changes to the edited table made during this edit session. See Audit Trail Functions for details.
Default is not to perform edit auditing.

Do not protect Prime Key
Specifies that data occupying columns that comprise the table's primary key is eligible for update.
Default is that this data is read-only.

COMMIT Options:

Commit on SAVE with no error
COMMIT only if SAVE is executed without errors.

Commit on SAVE
COMMIT on SAVE regardless of errors.

Commit on exit from edit session
COMMIT only on exit of the edit session.

Explicit Table Lock:

None
No explict table locking prior to load. (Recommended)

Share mode
Prevents anything other than read-only operations being performed on the table whilst it is being edited.

Exclusive mode
Prevents another process from performing any operations on the table whilst it is being edited, unless the process is running with an isolation level of Uncommitted Read (UR) in which case read-only (dirty read) operations may be performed.

Concurrency (Isolation) Options:

Use DB2 Default Isolation level
No "WITH" clause is added to the SQL statement used to fetch data.

Uncommitted Read (least restrictive)
"WITH UR" clause is added to the SQL statement used to fetch data.

Cursor Stability
"WITH CS" clause is added to the SQL statement used to fetch data.

Read Stability
"WITH RS" clause is added to the SQL statement used to fetch data.

Repeatable Read (most restrictive)
"WITH RR" clause is added to the SQL statement used to fetch data.

Use/Keep Locks:

None
No "KEEP" clause is added to the SQL statement used to fetch data.

Share
"KEEP SHR" clause is added to the SQL statement used to fetch data. Applicable only if either "Read Stability" (RS) or "Repeatable Read" (RR) isolation levels are selected.

Update
"KEEP UPDATE" clause is added to the SQL statement used to fetch data. Applicable only if either "Read Stability" (RS) or "Repeatable Read" (RR) isolation levels are selected.

Exclusive
"KEEP EXCLUSIVE" clause is added to the SQL statement used to fetch data. Applicable only if either "Read Stability" (RS) or "Repeatable Read" (RR) isolation levels are selected.

Create/Edit DB2 Structure SQL Clauses

The Create DB2 Structure SQL Clauses panel view is displayed only if command **SQL** is executed.

This panel view displays and supports editing of the SQL query clauses generated by the DB2 Row Selection (WHERE) and DB2 Column Selection and Ordering (SELECT/SORT) panels.

Panel Fields - Create/Edit DB2 Structure SQL CLauses

SQL:

Select>
The comma separated list of DB2 table column names that will be initially selected (e.g. visible in edit/browse). Although this field may be modified manually, it will be automatically updated (with all manually modification lost) whenever the SELECT primary command is used to select columns using the panel interface.

Where>
The "WHERE" clause passed as part of the SQL SELECT used to load data rows. Although this field may be modified manually, it will be automatically updated (with all manually modification lost) whenever the WHERE primary command is used to set row selection criteria using the panel interface.

Order By>
The "ORDER BY" clause passed as part of the SQL SELECT used to load data rows. Although this field may be modified manually, it will be automatically updated (with all manually modification lost) whenever the SELECT primary command is used to set row sort order using the panel interface.

Miscellaneous:

ColWidth>
A series of "OPTION(**COLWIDTH command**)" clauses passed to edit/browse to define the desired restricted visible width of long data columns. Although this field may be modified manually, it will be automatically updated (with all manually modification lost) whenever the SELECT primary command is used to set column width values using the panel interface.

RColour>
A series of "OPTION(**RCOLOUR command**)" clauses passed to edit/browse to define the desired row colouring conditions.

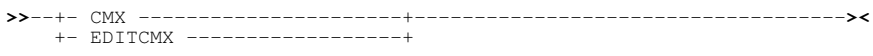
CColour>
A series of "OPTION(**CCOLOUR command**)" clauses passed to edit/browse to define the desired column colouring conditions.

InitCmd>
A series of quoted blank delimited structured edit commands to be executed immediately by edit/browse.

Primary Commands

The following primary commands are supported by the Create DB2 Structure (SDO) panels.

CMX



Applicable only to the **Create/Edit DB2 Structure** panel, CMX generates the SELCOPYi CREATE STRUCTURE command and copies it to an in-storage output file in a format suitable for subsequent execution using the ACTION (default >F16>) facility.

CMX is assigned to <F5> by default.

COLUMNS

```
>>---- COLumns ------------------------------------------------------><
```

Applicable only to the **Create/Edit DB2 Structure SQL Clauses** panel, COLUMNS sets the "Order By" field to a comma separated list of all columns names in the specified table.

Each column name (including its trailing comma) is deliberately blank padded up to 50 bytes. This is designed so that, when "EXPAND" (F14) is executed with the cursor in the "Order By" field so that the field contents are displayed in a separate Text Editor view, each column appears on a new line. This make it easy to re-order the fields and delete unwanted sort field names.

INDEX

```
>>--+- INDEX --------------------+-----------------------------------><
    +- IX ----------------------+
```

Applicable only to the **Create/Edit DB2 Structure SQL Clauses** panel, INDEX displays a selectable list of DB2 indexes that apply to the specified base table.

On selection of an individual index, its column names are used to populate the "Order By" field.

JCL

```
>>--+- JCL ---------------------+-----------------------------------><
    +- EDITJCL ----------------+
```

Applicable only to the **Create/Edit DB2 Structure** panel, JCL generates the SELCOPYi CREATE STRUCTURE command and copies it to an in-storage output file with JCL statements that execute the SDEAMAIN program. This job may be submitted to batch using the SELCOPYi text editor SUBMIT primary command.

JCL is assigned to <F6> by default.

OPTIONS

```
>>---- OPTions ------------------------------------------------------><
```

Applicable only to the **Create/Edit DB2 Structure** panel, OPTION opens the Create/Edit DB2 Structure Options panel view to tailor DB2 concurrency and locking options, etc.

OPTIONS is assigned to <F19> by default.

SELECT

```
>>--+- SELect -------------------+-----------------------------------><
    +- SORT --------------------+
```

Applicable only to the **Create/Edit DB2 Structure** panel, SELECT opens the DB2 Column Selection and Ordering panel to select table columns and establish the row (order by) sequence.

SELECT is assigned to <F17> by default.

SQL

```
>>---- SQL -------------------------------------------------------><
```

Applicable only to the **Create/Edit DB2 Structure** panel, SQL opens the to view and optionally modify the DB2 SELECT, WHERE and ORDER BY clauses generated by the DB2 Row Selection (WHERE) and DB2 Column Selection and Ordering (SELECT/SORT) panels.

SQL is assigned to <F20> by default.

WHERE

```
>>---- WHere ------------------------------------------------------><
```

Applicable only to the **Create/Edit DB2 Structure** panel, WHERE opens the DB2 Row Selection panel to define the table row selection criteria.

WHERE is assigned to <F18> by default.

DB2 Utilities

List DB2 Utilities Menu Panel

The DB2 Utilities menu panel (ZZS2UTIL) is an interactive panel window, opened on selection of option 12. in the DB2 Primary options menu.

This panel allows the user to invoke panel driven interfaces to standard DB2 utilities.

Menu Bar Items

`File`
The File drop-down menu contains the single item, Exit, to close the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

`Help`
Open the general help for the List DB2 Objects option menu panel.

Options

1 Unload	Generate Unload DB2 table job
2 Load	Generate Load DB2 table job

DB2 UNLOAD Utility

The DB2 Unload utility panel (ZZS2ULD0) assists the user in creating a batch JCL job to unload data from a DB2 table to a sequential output dataset.

Options are provided to:

- Select table columns and set row sort order.
- Set table row selection criteria.
- Set DB2 concurrency and locking options etc.
- Output data in delimited format e.g. comma separated (CSV).
- Map table columns to output fields defined by a structure/copybook.

These sequence of panel views are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select "Unload", option 1, from the SELCOPYi DB2 Utilities menu. (DB2 12.1)
- Type the "U" line-command against an entry in a DB2 Tables List.

UNLOAD Utility Table Name & Location

Enter the name of the table within the current DB2 subsystem from which rows will be unloaded The current DB2 subsystem is displayed in the panel window title bar.

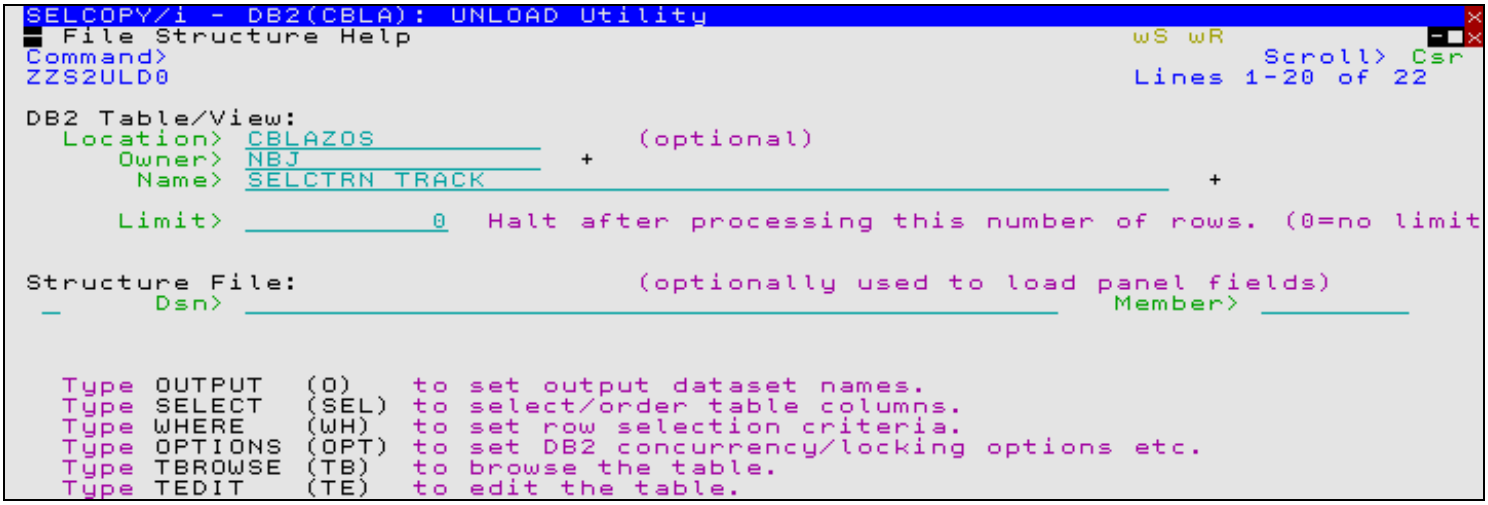


Figure 302. DB2: Unload Table - Table Name & Location.

UNLOAD Utility Table Name & Location - Menu Bar Items

The following menu bar items are common to all Unload Utility panel views and sub-panels.

File
The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Structure
Open the Create DB2 Table Edit Structure panel to edit or create a permanent SELCOPYi DB2 structure that may be used in the Unload operation.

Help
Display help for this panel view.

UNLOAD Utility Table Name & Location - Panel Fields

DB2 Base Table:

SSN>
The DB2 subsytem on which the table to be unloaded is located.

If left blank then the current subsystem (identified in the window title bar) will be used.

Location>
The server location of the table to be unloaded. If left blank, the local server for the connected DB2 sub-system is used.

Owner>
The owner (schema) of the table to be unloaded.
A table selection list will be provided if wildcards (*) are supplied.

Name>
The name of the table to be unloaded.
A table selection list will be provided if wildcards (*) are supplied.

Limit>
Indicates the maximum number of rows that are to be unloaded from the table. Zero indicates there will be no limit. If the specified number is less than zero, no row is unloaded from the table.

Structure File:
If activated, defines fields which together specify a structure file used to map the table data and to load panel fields that specify various options for the unload procedure.

If no structure is specified then SELCOPYi will generate one using the DB2 SQLDA chain for the specified results table columns.

Dsn>
Identifies the fully qualified data set name of an existing sequential data set or PDS/PDSE library. The dataset must containing an existing DB2 SDO structure.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>

If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must a specify a member name.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

UNLOAD Utility Output Datasets

The DB2 Unload Table Utility Output Datasets panel view is displayed only if command **OUTPUT** is executed.

This panel view defines the name of the output dataset for the unloaded data and optionally the name of an unload PUNCH output dataset. A structure/copybook dataset or library member name may also be specified.

An output structure may be generated and written to the specified output structure dataset/member. Alternatively, the structure may already exist and be used to define the format of the unloaded output records (i.e. remap the table column data).

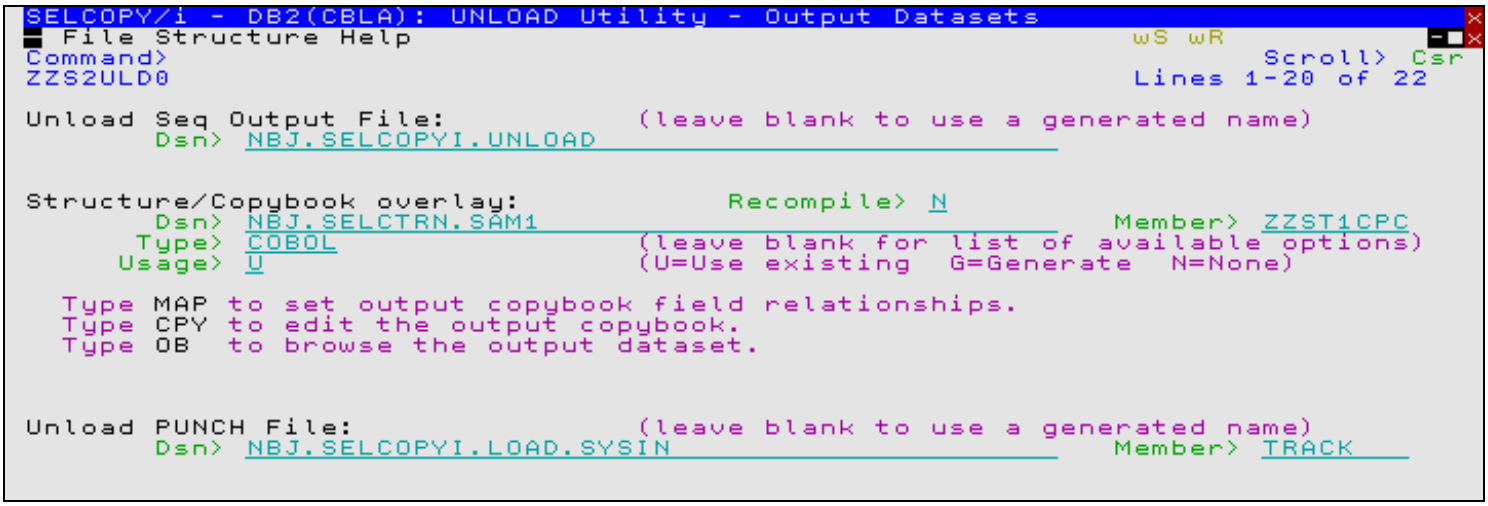


Figure 303. DB2: Unload Table - Output Datasets.

UNLOAD Utility Output Datasets - Column Remap

If the Usage value is set to be "U", then the specified structure or copybook must already exist and is to be used to define the layout of the output records.

The SELCOPYi DB2 Unload utility uses an output structure to generate the DB2 UNLOAD syntax required to remap the order, length and/or source datatype of the DB2 table column data when it is written to the output dataset.

Since a DB2 table has only one structure which maps all rows of the table, SELCOPYi uses the first or only record-type mapping definition within the output structure when matching table column names to output record field names.

The contents of each selected table column will be unloaded to the position, length and datatype of its matching output field. By default, unload will occur only for data in DB2 table columns whose names match those of the output record-type fields. However, the Remap Record Layout sub-panel (see command MAP) may be used to match table columns with output record fields that have a different name.

If the output record-type mapping contains an unmatched field, then the output record will contain blanks at that field's location within the record. However, for variable record format output, an unmatched field that occurs after the last matched field will not be included in the output record.

UNLOAD Utility Output Datasets - Panel Fields

Unload Seq Output File:

Dsn>

Identifies the fully qualified data set name of a new or existing sequential data set to receive the unloaded table data.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Structure/Copybook overlay:

This option comprises fields which together specify a cataloged structure file (Assembler, COBOL or PL1 Copybook, ADATA file or a SELCOPYi SDO) used to the define the location and data-types of fields that will receive values from the unloaded table columns.

If "Usage" option "G" (Generate) is specified then this is the name of a SELCOPYi SDO that will be created. This SDO may subsequently used to to map the output dataset using SELCOPYi's Data-Edit features.

Dsn>
Identifies the fully qualified data set name of an existing sequential data set or PDS/PDSE library. The dataset must containing an existing DB2 SDO structure.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must a specify a member name.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Type:
Indicate the type of structure (ASM, COBOL, PL1, ADATA or SDO).

Usage:
Indicate the action.

U	Use existing structure to define the layout of output fields.
G	Generate a new structure (SDO) to map the output fields.
N	None (ignore any named structure).

Unload PUNCH File:

Dsn>
Identifies the fully qualified data set name of a new or existing sequential data set or PDS/PDSE library to receive the LOAD statements for subsequent reload of the table data. These statements will be automatically generated when the DB2 UNLOAD utilty itself is run.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must a specify a member name.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

UNLOAD Utility Options

The DB2 Unload Table Utility Options panel view is displayed only if command **OPTIONS** is executed.

This panel view identifies DB2 UNLOAD processing options.

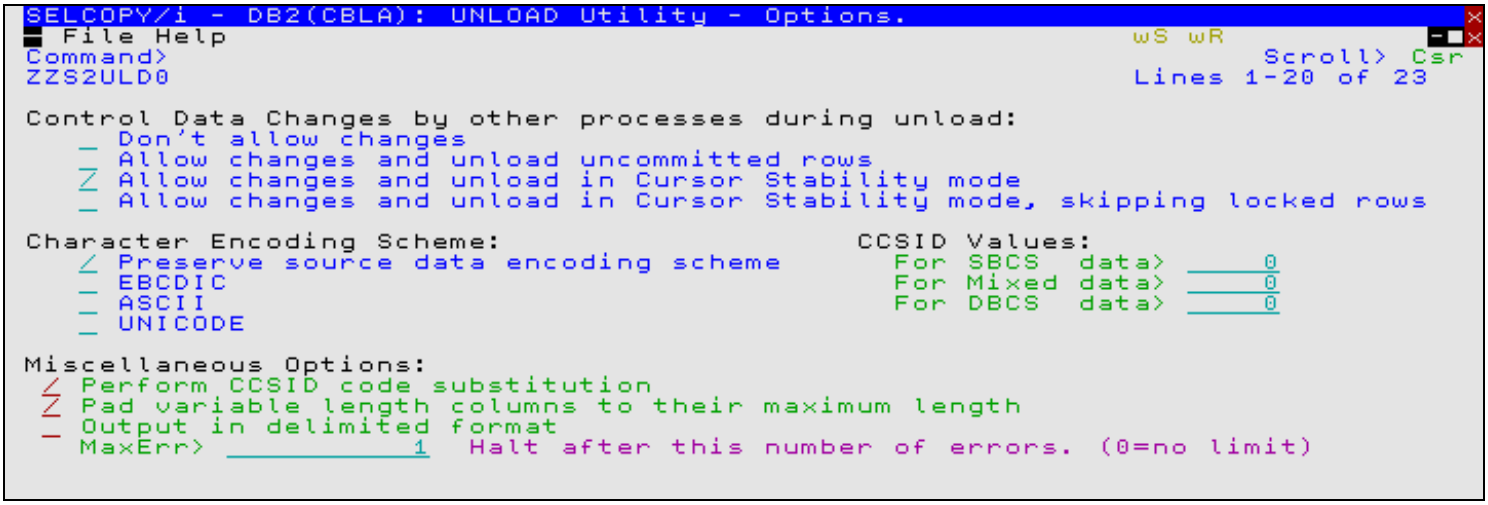


Figure 304. DB2: Unload Table - Options.

UNLOAD Utility Options - Panel Fields

Control Data Changes by other processes during unload:

Don't allow changes
SHRLEVEL REFERENCE
Specifies that during the unload operation, rows of the tables can be read, but cannot be inserted, updated, nor deleted by other DB2 threads.

Allow changes and unload uncommitted rows
SHRLEVEL CHANGE ISOLATION UR
Specifies that rows can be read, inserted, updated, and deleted from the table space or partition while the data is being unloaded. Uncommitted rows, if they exist, are to be unloaded. The unload operation is performed with minimal interference from the other DB2 operations that are applied to the objects from which the data is being unloaded

Allow changes and unload in Cursor Stability mode
SHRLEVEL CHANGE ISOLATION CS
Specifies that rows can be read, inserted, updated, and deleted from the table space or partition while the data is being unloaded. The UNLOAD utility is to read rows in cursor stability mode. With CS, the UNLOAD utility assumes CURRENTDATA(NO).

Allow changes and unload in Cursor Stability mode, skipping locked rows
SHRLEVEL CHANGE ISOLATION CS SKIP LOCKED DATA
As above, but addtionally specifies that the UNLOAD utility is to skip rows on which incompatible locks are held by other transactions.

Character Encoding Scheme:

Preserve source data encoding scheme
The encoding scheme of the source data is preserved.

EBCDIC
Specifies that all output data of the character type is to be in EBCDIC. If a different encoding scheme is used for the source data, the data (except for bit strings) is converted into EBCDIC.

ASCII
Specifies that all output data of the character type is to be in ASCII. If a different encoding scheme is used for the source data, the data (except for bit strings) is converted into ASCII.

UNICODE
Specifies that all output data of the character type is to be in UNICODE. If a different encoding scheme is used for the source data, the data (except for bit strings) is converted into UNICODE.

CCSID Values:
Specifies three coded character set identifiers (CCSIDs) that are to be used for the data of character type in the output records, including data that is unloaded in the external character formats.

For SBCS data>
The CCSID for SBCS data.

For Mixed data>
The CCSID for Mixed data.

For DBCS data>
The CCSID for DBCS data.

Miscellaneous Options:

Perform CCSID code substitution
Specifies that CCSID code substitution is to be performed during unload processing (default).

When a string is converted from one CCSID to another (including EBCDIC, ASCII, and Unicode), a substitution character is sometimes placed in the output string.

For example, this substitution occurs when a character (referred to as a code point) that exists in the source CCSID does not exist in the target CCSID.

You can deselect this option to prevent the UNLOAD utility from allowing this substitution.

If you do so, and character substitution is attempted while data is being unloaded, this action is treated as a conversion error.

The record with the error is not unloaded, and the process continues until the total error count reaches the number that is specified by MAXERR

Pad variable length columns to their maximum length
Selecting this option specifies that:

- Default UNLOAD processing pads variable-length columns in the unloaded records to their maximum length.
- The padded data fields are preceded by the length fields that indicate the size of the actual data without the padding.

- When the output records are reloaded with the LOAD utility, padded data fields are treated as varying-length data.

Deselecting this option specifies that the variable-length columns in the unloaded records are to occupy the actual data length without additional padding.

Output in delimited format
Selecting this option specifies that the output will be in a delimited format, with all fields produced as character strings or external numeric values. and each column value will be separated from the next column by a column delimiter e.g. a comma

Options specific to delimited output, such as choice of delimiter character, may be specified by executing the **DELIM** primary command.

MaxErr>
Specifies the maximum number of records in error that are to be allowed; the unloading process terminates when this value is reached.

If you specify 0 or any negative number, execution continues regardless of the number of records that are in error.

UNLOAD Utility Floating Point Options

The DB2 Unload Table Utility Floating Point Options panel view is displayed only if command **FLOAT** is executed.

This panel view identifies the format of floating point values in the unloaded output records.

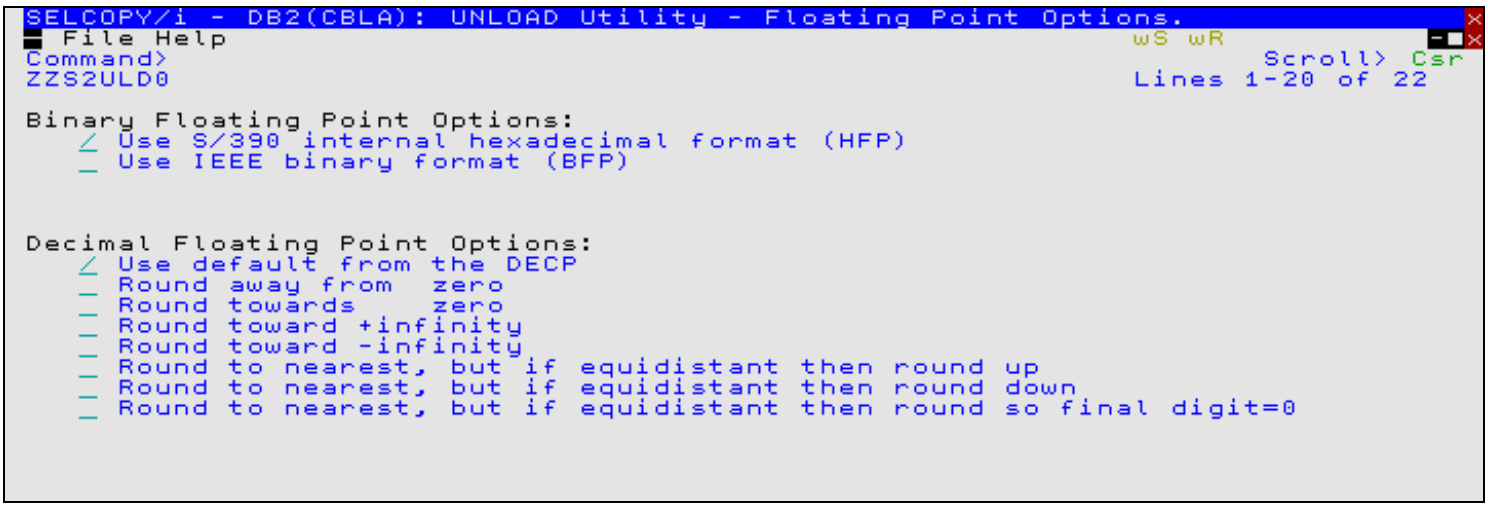


Figure 305. DB2: Unload Table - Floating Point Options.

UNLOAD Utility Floating Point Options - Panel Fields

Binary Floating Point Options:

Use S/390 internal hexadecimal format (HFP)
Indicates that the binary floating point data is written to the output records in the "S/390" internal format (also known as the hexadecimal floating point, or HFP).

Use IEEE binary format (BFP)
Indicates that the binary floating-point data is written to the output records in the IEEE format (also known as the binary floating point, or BFP).

Decimal Floating Point Options:
Specifies the rounding mode (DECFLOAT_ROUNDMODE) to be used when DECFLOATs are manipulated.

Use default from the DECP
Use the DECFLOAT ROUNDING MODE from the DECP

Round away from zero
ROUND_UP
If all of the discarded digits are 0, the result is unchanged. Otherwise, the result coefficient should be incremented by 1 (rounded up).

Round towards zero
ROUND_DOWN
Truncation - the discarded digits are ignored

Round toward +infinity
ROUND_CEILING
The discarded digits are removed if they are all zero or if the sign is negative. Otherwise, the result coefficient should be incremented by 1 (rounded up).

Round toward -infinity
ROUND_FLOOR
The discarded digits are removed if they are all zero or positive. Otherwise, the sign is negative and the result coefficient should be incremented by 1 (rounded up).

Round to nearest, but if equidistant then round up
ROUND_HALF_UP
If equidistant, round up. If the discarded digits are greater than or equal to 0.5, the result coefficient should be incremented by 1 (rounded up). Otherwise the discarded digits are ignored.

Round to nearest, but if equidistant then round down
ROUND_HALF_DOWN
If equidistant, round down. If the discarded digits are greater than 0.5, the result coefficient should be incremented by 1 (rounded up). The discarded digits are ignored if they are 0.5 or less.

Round to nearest, but if equidistant then round so final digit=0
ROUND_HALF_EVEN
If equidistant, round so that the final digit is even. If the discarded digits are greater than .05, the result coefficient should be incremented by 1 (rounded up). The discarded digits are ignored if they are less than 0.5. If the result coefficient is .05 and the rightmost digit is even, the result coefficient is not altered. If the result coefficient is .05 and the rightmost digit is odd, the result coefficient should be incremented by 1 (rounded up)

UNLOAD Utility Delimited Output Options

The DB2 Unload Table Utility Delimited Output Options panel view is displayed only if command **DELIM** is executed.

Applicable to delimited format output only, this panel view identifies the delimiter and punctuation characters used in unloaded output records.

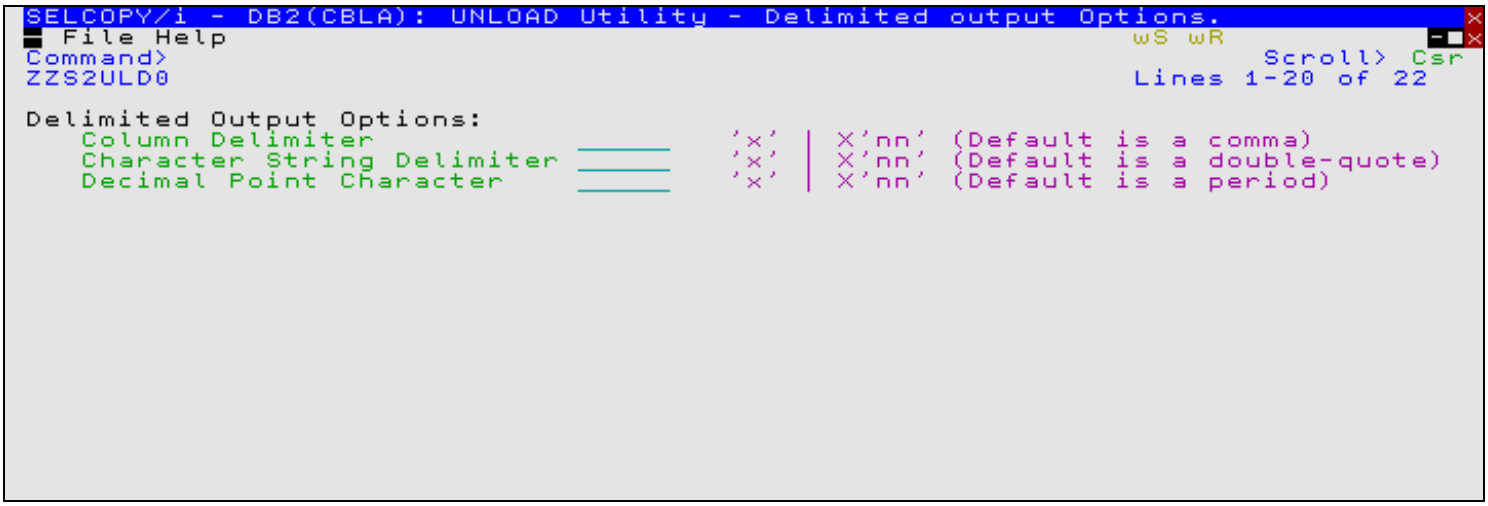


Figure 306. DB2: Unload Table - Delimited Output Options.

UNLOAD Utility Delimited Output Options - Panel Fields

Delimted Output Options:

Column Delimiter>
Specified the column delimiter used to separate fields in the output file. Default is a "," (comma).

Character String Delimiter>
Specifies the delimiter used for character string values within the output file. The default is a "" (quotation mark).

The UNLOAD utility adds this character before and after every character string. To delimit character strings that contain the character string delimiter, the UNLOAD utility repeats the character string delimiter where it used in the character string. The LOAD utility will interpret any pair of character delimiters that are found between the enclosing character delimiters as a single character.

Decimal Point Character>
Specifies the decimal point character that is used in the output file. The default is a "." (dot/period).

UNLOAD Utility Primary Commands

The following primary commands are supported by selected views in the DB2 Unload utility sequence of panel views.

CPY

```
>>---- CPY ------------------------------------------------------------><
```

Applicable only to the **Output Datasets** panel view. CPY opens a Text Editor view to edit the specified output copybook.

DELIM

```
>>--+- DELIM --------+-------------------------------------------------><
    +- DLM ----------+
```

Applicable to all DB2 UNLOAD utility panel views, DELIM opens the DB2 UNLOAD Delimited Output Options panel view.

DELIM is assigned to <F11> by default.

FLOAT

```
>>---- FLOAT ----------------------------------------------------------><
```

Applicable to all DB2 UNLOAD utility panel views, FLOAT opens the DB2 UNLOAD Floating Point Options panel view.

FLOAT is assigned to <F10> by default.

MAP

```
>>---- MAP ------------------------------------------------------------><
```

Applicable only to the **Output Datasets** panel view. MAP opens the Remap Record Layout to allow remap of table column names to output structure record-type field names.

OB

```
>>---- OB -------------------------------------------------------------><
```

Applicable only to the **Output Datasets** panel view. OB opens a Data Editor view to browse the specified output dataset.

OPTIONS

```
>>---- OPTions ---------------------------------------------------------><
```

Applicable to all DB2 UNLOAD utility panel views, OPTIONS opens the DB2 UNLOAD Options panel view.

OPTIONS is assigned to <F6> by default.

OUTPUT

```
>>---- Output ---------------------------------------------------------><
```

Applicable to all DB2 UNLOAD utility panel views, OUTPUT opens the DB2 UNLOAD Output Datasets panel view.

OUTPUT is assigned to <F5> by default.

SELECT

```
>>---- SELect ---------------------------------------------------------><
```

Applicable to all DB2 UNLOAD utility panel views, SELECT opens the DB2 UNLOAD utility Select Table Columns panel which is similar to the SDE Select Columns panel. This panel allows specific selection of the table columns to be unloaded and also to establish the input row (order by) sequence.

SELECT is assigned to <F17> by default.

TBROWSE

```
>>---- TBrowse --------------------------------------------------------><
```

Applicable to all DB2 UNLOAD utility panel views, TBROWSE opens a Data Editor view to browse the specified input DB2 table.

TBROWSE is assigned to <F22> by default.

TEDIT

```
>>---- TEdit ------------------------------------------------------------><
```

Applicable to all DB2 UNLOAD utility panel views, TEDIT opens a Data Editor view to edit the specified input DB2 table.

TEDIT is assigned to <F23> by default.

WHERE

```
>>---- WHere ------------------------------------------------------------><
```

Applicable to all DB2 UNLOAD utility panel views, WHERE opens the DB2 Row Selection panel to define the table row selection criteria.

WHERE is assigned to <F18> by default.

DB2 LOAD Utility

The DB2 Load Utility panel (ZZS2LOD0) assists the user in creating a batch job to load data into a DB2 table from a sequential input dataset or PDS/PDSE library member.

Options are provided to:

- Select table columns.
- Input data in delimited format, e.g. comma separated (CSV).
- Map table columns to input fields defined by a structure/copybook.

These sequence of panel views are interactive panel windows (window class WINWIPO0) and may be started via the following:

- Select "Load", option 2, from the SELCOPYi DB2 Utilities menu. (DB2 12.2)
- Type the "L" line-command against an entry in a DB2 Tables List.

The DB2 LOAD utility panels closely resemble those for the UNLOAD utility.

LOAD Utility Table Name & Location

Enter the name of the table within the current DB2 subsystem to which rows will be loaded The current DB2 subsystem is displayed in the panel window title bar.

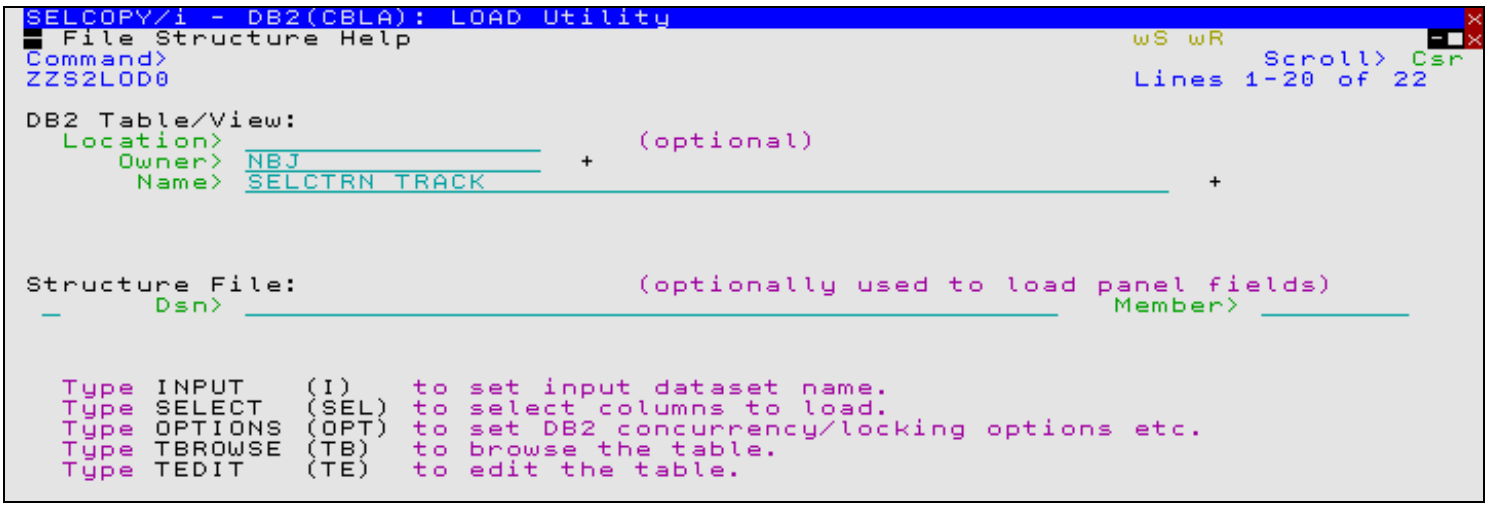


Figure 307. DB2: Load Table - Table Name & Location.

LOAD Utility Table Name & Location - Menu Bar Items

The following menu bar items are common to all LOAD utility panel views and sub-panels.

File
: The File drop-down menu contains the single item "Exit" which simply closes the panel and, if the last panel open in the current DB2 panel hierarchy, close (disconnect) the connection to the relevant DB2 subsystem.

Structure
: Open the Create DB2 Table Edit Structure panel to edit or create a permanent SELCOPYi DB2 structure that may be used in the LOAD operation.

Help
: Display help for the current panel view.

LOAD Utility Table Name & Location - Panel Fields

DB2 Base Table:

SSN>
: The DB2 subsytem on which the table to be loaded is located.

  If left blank then the current subsystem (identified in the window title bar) will be used.

Location>
: The server location of the table to be loaded. If left blank, the local server for the connected DB2 sub-system is used.

Owner>
The owner (schema) of the table to be loaded.
A table selection list will be provided if wildcards (*) are supplied.

Name>
The name of the table to be loaded.
A table selection list will be provided if wildcards (*) are supplied.

Structure File:
If activated, defines fields which together specify a structure file used to map the table data and to load panel fields that specify various options for the load procedure.

If no structure is specified then SELCOPYi will generate one using the DB2 SQLDA chain for the specified results table columns.

Dsn>
Identifies the fully qualified data set name of an existing sequential data set or PDS/PDSE library. The dataset must containing an existing DB2 SDO structure.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must a specify a member name.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

LOAD Utility Input Dataset

The DB2 Load Table Utility Input Datasets panel view is displayed only if command **INPUT** is executed.

This panel view defines the name of the input dataset from which data will be loaded. A structure/copybook dataset or library member name may also be specified.

If no input structure is specified and activated, records in the input dataset must be in a format that will load directly into the selected DB2 table columns.

If the format of the data in the input dataset does not exactly match that of the DB2 table columns, then an input structure may be supplied to identify the data type and location within the input records of the fields to be loaded.

If activated, the input structure specified in the panel field must already exist and will be used to define the format of the input records.

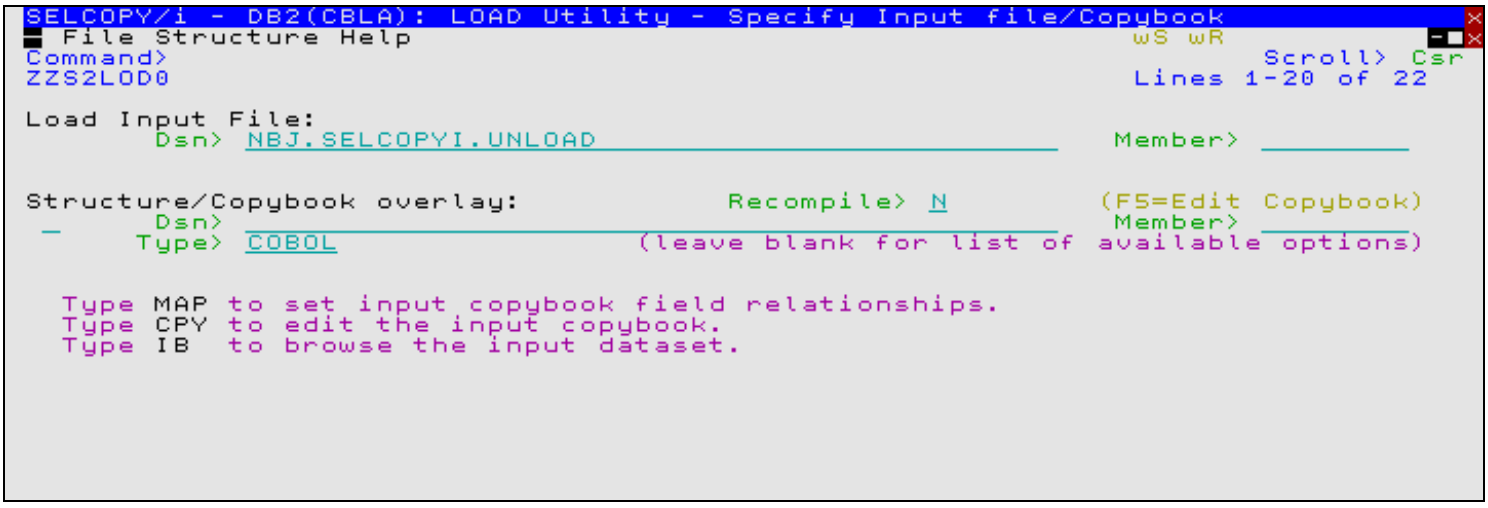


Figure 308. DB2: Load Table - Input Dataset.

LOAD Utility Input Dataset - Field Identification

The SELCOPYi DB2 Load utility may uses a supplied structure to generate the DB2 LOAD syntax required to identify the location, length and/or datatype of the source fields within the input dataset.

Since a DB2 table has only one structure which maps all rows of the table, SELCOPYi uses the first or only record-type mapping definition within the input structure when matching table column names to input record field names.

The contents of each selected table column will be loaded from the position, length and datatype of its matching input field. By default, load will occur only for data in DB2 table columns whose names match those of the input record-type fields. However, the Remap Record Layout sub-panel (see command MAP) may be used to match table columns with input record fields that have a different name.

LOAD Utility Input Dataset - Panel Fields

Load Input File:

Dsn>
Identifies the fully qualified data set name of a new or existing sequential data set or PDS/PDSE library to send the loaded table data.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must a specify a member name.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Structure/Copybook overlay:
If activated, this option comprises fields which together specify a cataloged structure file (Assembler, COBOL or PL1 Copybook, ADATA file or a SELCOPYi SDO) used to the define the location and data-types of fields that will receive values from the loaded table columns.

Dsn>
Identifies the fully qualified data set name of an existing sequential data set or PDS/PDSE library. The dataset must containing an existing DB2 SDO structure.

A selectable list of data sets will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the Dsn> field contains the DSN of a PDS/PDSE library, then this field must a specify a member name.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

Type:
Indicate the type of structure (ASM, COBOL, PL1, ADATA or SDO).

LOAD Utility Options

The DB2 Load Table Utility Options panel view is displayed only if command **OPTIONS** is executed.

This panel view identifies DB2 LOAD processing options.

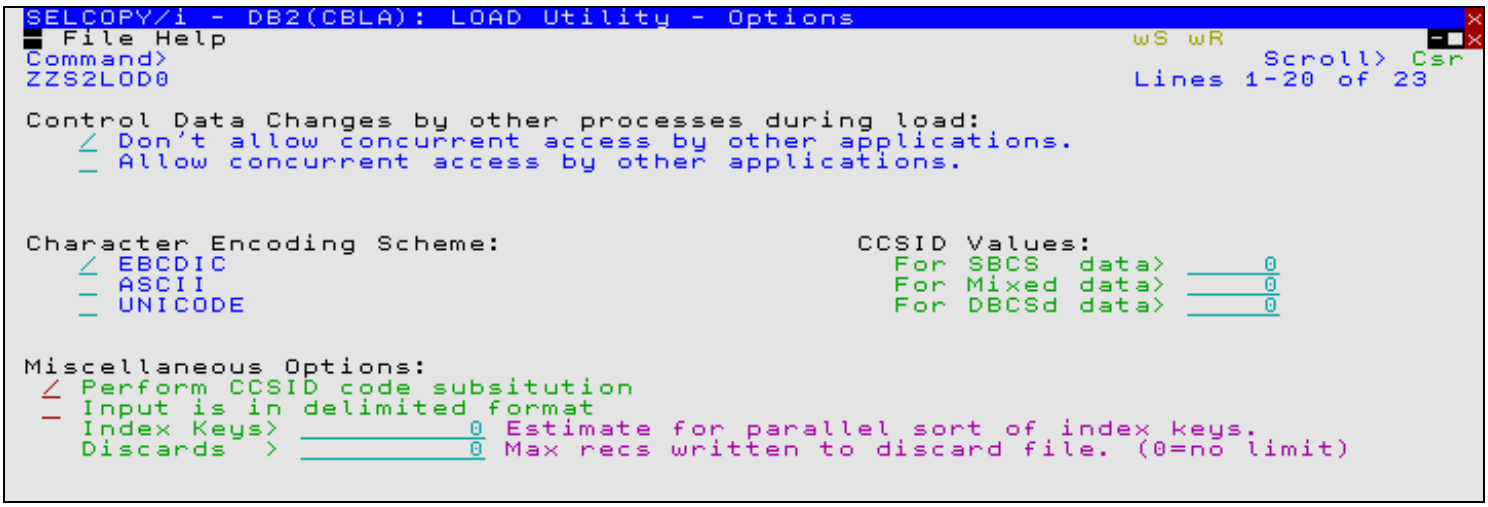


Figure 309. DB2: Load Table - Options.

LOAD Utility Options - Panel Fields

Control Data Changes by other processes during load:

Don't allow concurrent access by other applications
SHRLEVEL NONE
Specifies that applications have no concurrent access to the table space or partition

Allow concurrent access by other applications
SHRLEVEL CHANGE
Specifies that applications can concurrently read from and write to the table space or partition into which LOAD is loading data

Character Encoding Scheme:

EBCDIC
Specifies that the input data file is EBCDIC.

ASCII
Specifies that the input data file is ASCII. Numeric, date, time, and timestamp internal formats are not affected by the ASCII option

UNICODE
Specifies that the input data file is Unicode. The UNICODE option does not affect the numeric internal formats.

CCSID Values:
Specifies three coded character set identifiers (CCSIDs) for the input file.

For SBCS data>
The CCSID for SBCS data.

For Mixed data>
The CCSID for Mixed data.

For DBCS data>
The CCSID for DBCS data.

Miscellaneous Options:

Perform CCSID code substitution
Specifies that CCSID code substitution is to be performed during load processing (default).

Input is in delimited format
Specifies that the input data file is in a delimited format. When data is in a delimited format, all fields in the input data set are character strings or external numeric values. In addition, each column in a delimited file is separated from the next column by a column delimiter character. e.g. a comma.

Options specific to delimited input, such as choice of delimiter character, may be specified by typing the "DELIM" (DLM) primary command.

Index Keys>
Specifies that index keys are to be sorted in parallel during the SORTBLD phase to improve performance.

Specifies an integer to provide an estimate of the number of index keys that are to be sorted.

Discards>
Specifies the maximum number of source records that are to be written on the discard data set.

A value of 0 specifies that you do not want to set a maximum value. The entire input data set can be discarded. The default value is 0.

LOAD Utility Floating Point Options

The DB2 Load Table Utility Floating Point Options panel view is displayed only if command **FLOAT** is executed.

This panel view identifies the format of floating point values in the input records.

LOAD Utility Floating Point Options - Panel Fields

Binary Floating Point Options:

Use S/390 internal hexadecimal format (HFP)
Indicates that binary floating point data in the input records is in the "S/390" internal format (also known as the hexadecimal floating point, or HFP).

Use IEEE binary format (BFP)
Indicates that binary floating-point data in the input records is in the IEEE format (also known as the binary floating point, or BFP).

Decimal Floating Point Options:
Specifies the rounding mode (DECFLOAT_ROUNDMODE) to be used when DECFLOATs are manipulated.

Use default from the DECP
Use the DECFLOAT ROUNDING MODE from the DECP

Round away from zero
ROUND_UP
If all of the discarded digits are 0, the result is unchanged. Otherwise, the result coefficient should be incremented by 1 (rounded up).

Round towards zero
ROUND_DOWN
Truncation - the discarded digits are ignored

Round toward +infinity
ROUND_CEILING
The discarded digits are removed if they are all zero or if the sign is negative. Otherwise, the result coefficient should be incremented by 1 (rounded up).

Round toward -infinity
ROUND_FLOOR
The discarded digits are removed if they are all zero or positive. Otherwise, the sign is negative and the result coefficient should be incremented by 1 (rounded up).

Round to nearest, but if equidistant then round up
ROUND_HALF_UP
If equidistant, round up. If the discarded digits are greater than or equal to 0.5, the result coefficient should be incremented by 1 (rounded up). Otherwise the discarded digits are ignored.

Round to nearest, but if equidistant then round down
ROUND_HALF_DOWN
If equidistant, round down. If the discarded digits are greater than 0.5, the result coefficient should be incremented by 1 (rounded up). The discarded digits are ignored if they are 0.5 or less.

Round to nearest, but if equidistant then round so final digit=0
ROUND_HALF_EVEN
If equidistant, round so that the final digit is even. If the discarded digits are greater than .05, the result coefficient should be incremented by 1 (rounded up). The discarded digits are ignored if they are less than 0.5. If the result coefficient is .05 and the rightmost digit is even, the result coefficient is not altered. If the result coefficient is .05 and the rightmost digit is odd, the result coefficient should be incremented by 1 (rounded up)

LOAD Utility Delimited Output Options

The DB2 Load Table Utility Delimited Output Options panel view is displayed only if command **DELIM** is executed.

Applicable to delimited format input only, this panel view identifies the delimiter and punctuation characters used in the input text records.

LOAD Utility Delimited Output Options - Panel Fields

Delimted Output Options:

Column Delimiter>
Specifies the column delimiter used to separate fields in the input file. Default is a "," (comma).

Character String Delimiter>
Specifies the delimiter used for character string values within the input file records. The default is a "" (quotation mark).

The LOAD utility strips this character from the start and end of a character string. Any occurrence of an adjacent pair of the delimiter character within the string is translated to be a single occurrence when loaded into the DB2 table column.

Decimal Point Character>
Specifies the decimal point character used in numeric values within the input file records. The default is a "." (dot/period).

LOAD Utility Primary Commands

The following primary commands are supported by selected views in the DB2 Load utility sequence of panel views.

CPY

```
>>---- CPY ------------------------------------------------------------><
```

Applicable only to the **Input Datasets** panel view. CPY opens a Text Editor view to edit the specified input copybook.

DELIM

```
>>--+- DELIM --------+-------------------------------------------------><
    +- DLM ----------+
```

Applicable to all DB2 LOAD utility panel views, DELIM opens the DB2 LOAD Delimited Input Options panel view.

DELIM is assigned to <F11> by default.

FLOAT

```
>>---- FLOAT ----------------------------------------------------------><
```

Applicable to all DB2 LOAD utility panel views, FLOAT opens the DB2 LOAD Floating Point Options panel view.

FLOAT is assigned to <F10> by default.

IB

```
>>---- IB -------------------------------------------------------------><
```

Applicable only to the **Input Datasets** panel view. IB opens a Data Editor view to browse the specified input dataset.

INPUT

```
>>---- Input ----------------------------------------------------------><
```

Applicable to all DB2 LOAD utility panel views, INPUT opens the DB2 LOAD Input Datasets panel view.

INPUT is assigned to <F5> by default.

MAP

```
>>---- MAP ------------------------------------------------------------><
```

Applicable only to the **Input Datasets** panel view. MAP opens the Remap Record Layout to allow remap of table column names to input structure record-type field names.

OPTIONS

```
>>---- OPTions --------------------------------------------------------><
```

Applicable to all DB2 LOAD utility panel views, OPTIONS opens the DB2 LOAD Options panel view.

OPTIONS is assigned to <F6> by default.

SELECT

```
>>---- SELect --------------------------------------------------------><
```

Applicable to all DB2 LOAD utility panel views, SELECT opens the DB2 LOAD utility Select Table Columns panel which is similar to the SDE Select Columns panel. This panel allows specific selection of the table columns to be loaded. Although this panel has a Seq column, used by SELCOPYi to define an ORDER BY clause, the order in which rows are stored is maintained by DB2. Therefore, values should not be entered in the Seq column.

SELECT is assigned to <F17> by default.

TBROWSE

```
>>---- TBrowse -------------------------------------------------------><
```

Applicable to all DB2 LOAD utility panel views, TBROWSE opens a Data Editor view to browse the specified output DB2 table. TBROWSE is assigned to <F22> by default.

TEDIT

```
>>---- TEdit ------------------------------------------------------------><
```

Applicable to all DB2 LOAD utility panel views, TEDIT opens a Data Editor view to edit the specified output DB2 table.

TEDIT is assigned to <F23> by default.

WHERE

```
>>---- WHere ------------------------------------------------------------><
```

Applicable to all DB2 LOAD utility panel views, WHERE opens the DB2 Row Selection panel to define the table row selection criteria.

WHERE is assigned to <F18> by default.

Remap record Layout

The Remap record Layout panel (ZZSGRMR0) allows the user to view and modify the relationship between fields (columns) in two different "record" layouts.

This panel describes the current relationship between receiving fields in the "Destination" structure/copybook to sending fields in the "Source" structure/copybook.

The "Source" field name may be altered (or blanked out) by overtyping it. Enter '*', or any non-blank name that is not a valid source field name, to display a selectable list of all fields in the source record layout. To select a field from the list, just place the cursor on the name and press "ENTER".

Press F3 to close the window when all modifications are completed.

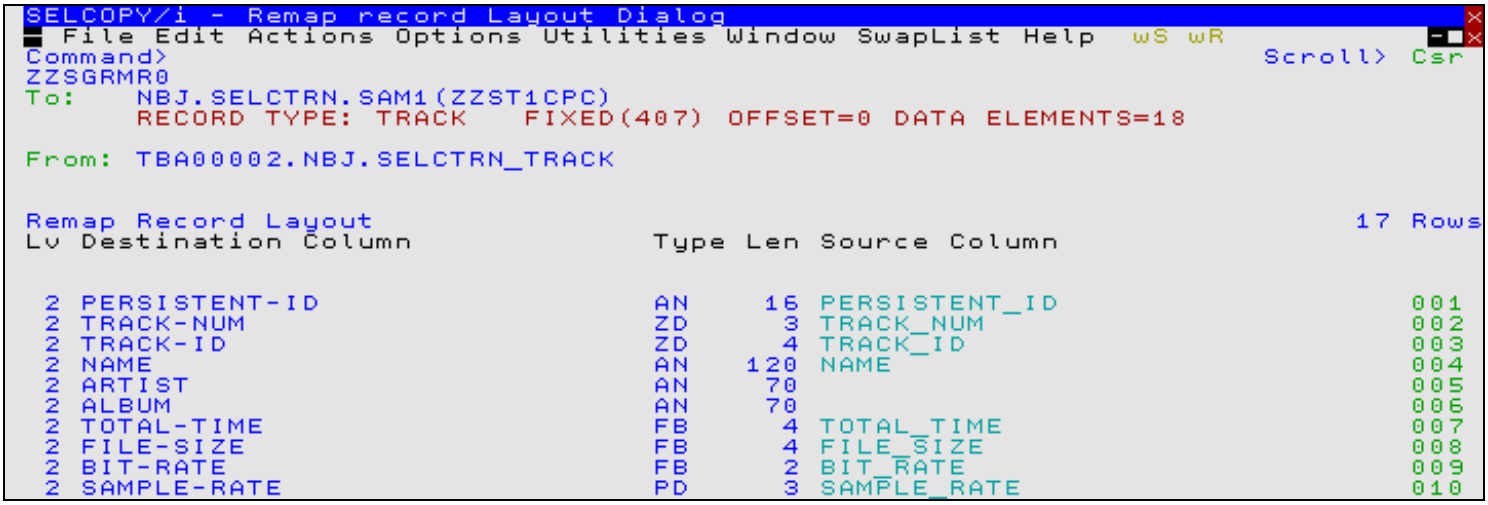


Figure 310. Remap Record Layout.

Create DB2 Table Report

The Create a Report using DB2 Table Data panel (ZZS2RPTM) is an interactive panel window opened on selection of option 3. from the SELCOPYi Print/Report Features menu or option 13. from the SELCOPYi DB2 Utilities menu.

Report output may be generated directly from a DB2 table or view specification, or from a DB2 result table generated by an SQL query statement. For a result table report, the SQL query statement may be passed directly as a parameter to the REPORT utility or indirectly as text in a data set or library member.

Options

1 **Single Table**	Report data sourced from a single DB2 Table.
2 **SQL File**	Report data sourced from SQL Query stored in a file.
3 **SQL**	Report data sourced from SQL Query entered on panel.

A full description of the REPORT utility panels for DB2 table input, together with working samples and report definition control statement syntax may be found in the ***SELCOPYi REPORT Utility*** manual.

Setup SELCOPYi DB2 Training Material

Use this panel to create your own DB2 Training tables for use while following the SELCOPY/i Training Manual, available online at:

```
www.cbl.com/pdf/SELCOPYi_Training_Manual_Rel340.pdf
```

The tables will be created in the specified database, which will be created with default settings if necessary.

The database name defaults to your current SQLID.

SMF Features (=13)

SMF Features Menu Panel

The MVS System Managed Facilities (SMF) panel (ZZSGSMF0) is an interactive panel window opened on selection of option 13. from the SELCOPYi Primary option menu.

Click here for further information about SELCOPYi's handling of SMF data.

Options

1 Browse	**SMFB**	Browse an SMF dataset with optional selection
2 Extract	**SMFEXTRC**	Make a selective extract from an SMF dataset
3 Report	**SMFRPT**	Create a report from an SMF dataset

SMF Features Further Information

SMF Field Mapping

The mappings for each supported SMF record type have been generated from the IBM manual *"MVS System Management Facilities (SMF)".*

https://www-304.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R3sa380667/$file/ieag200_v2r3.pdf

In general the field names used by SELCOPYi are shortened versions of those defined in the IBM documentation.

So where the IBM name begins with **"SMFnnn**" (nnn being the SMF record-type number), we have removed "SMFnnn" and replaced it with **"z"**. e.g. **"SMF17DSN"** has become **"zDSN"** within higher level record-type structure **"SMF017_Scratch_Dataset_Status"**.

In this example we would refer to **"SMF017_Scratch_Dataset_Status"** as the **record-type name** and **"zDSN"** as the **field name**.

However, some exceptions to this naming convention have been made in order introduce **readability** and **consistency** across different record types. e.g.
"SMF14JBN" and **"SMF60JNM"** have both become **"zJOBNAME".**
"SMF14UID" and **"SMF62UIF"** have both become **"zUSERID".**

HELP Key (F1) gives Field Info

The description supplied by the IBM documentation for each field has been made available to the SELCOPYi user browsing an SMF dataset.

Just place your cursor on the field heading or value and press the **"HELP"** key **(F1**) and the description will be displayed as a "message" at the top or bottom of the browse window. Just press enter again and it will disappear.

Primary/Secondary Segments

A feature of many SMF record-types is that they include within themselves many potentially **repeating group** (sub-structure) fields.

These are typically addressed by what IBM refer to as **"triplet"** fields. i.e. There will be:

1. A field **(xxxOF**) containing the **offset** within the base record to the repeating group.
2. A field **(xxxLN** usually directly following xxxOF) containing the **length** of each repeating group.
3. A field **(xxxON** usually directly following xxxLN) containing the **number of occurrences** of that group that actually exist in the record.

SELCOPYi treats these repeating groups almost as if they were different record-types. More accurately, they are treated as separate **"Secondary Segments"** that belong to a **"Base"** or **"Primary Segment"**.

When browsing an SMF dataset in SELCOPYi, the initial "Table" view of the file will display all segment types at once.

The user may choose whether secondary segments are initially displayed in full or as **"shadowed"** lines.

If the **"shadowed"** option is not selected then an instance of a record that has a number of secondary segments will occupy several lines within the table view, with each new group of attached secondary segments of the same type having a set of column headings displayed above.

Using standard Data-Edit techniques, the display may easily be configured to exclude/include any segment type, as well as include/exclude and re-order the fields within in each type.

Layout Source Text

Source text for the record layouts is supplied in members of the library "*<SystemHLQ>*.**SZZSDIST.SMFMAP**". It is here that you can find out which fields exist in any given record-type.

Base and associated secondary segments are kept in the same member.

SELCOPYi primary command **"SD QUERY SMFMAPLIB"** may be used to determine the name of the active definitions library.

From time to time, as further SMF record-types are mapped, an updated version of this library will be made available for download from the CBL web/ftp sites.

For convenience, an individual user may acquire an updated definitions library and name it under their own userid's HLQ. The name of this library may then be specified via the SMF item of the **SELCOPYi Settings panel (=0.10)**.

Note that there is an option on the SMF Browse panel to **"Reload Full Layout Definitions"**. This corresponds to the **RELOAD** parameter on the SMFB primary command.

Select this option only when you have updated your definitions library since there is a significant performance implication to generating the layout Structure Definition Object (SDO) from the text library members.

Another implication of recreating the layout SDO is that any user updates will be lost. These may include:

- Any "permanent" alterations to the field display order/selection and/or column-widths made via the SDE SELECT Columns panel.
- Any row colouring options specified via the RCOLOUR command.
- Any column colouring options specified via the CCOLOUR command.

Useful Commands

The following commands are particularly handy while browsing an SMF dataset.

VBASE

Type **VBASE** during your browse session to display secondary segments as shadow lines. This is the initial display mode when option **"Segment> SHADOW"** is selected from the SMF Browse entry panel.

Type **VBASE OFF** during your browse session to display secondary segments in full detail. This is the initial display mode when option **"Segment> SHOW"** is selected from the SMF Browse entry panel.

Note that while secondary segments are shadowed (thereby occupying much less screen space) the detail for any single shadowed segment may be displayed in a separate window by placing the cursor on the shadow line and pressing the **"ZoomW"** key **(Shift-F5).**

The shadow lines themselves may be suppressed using the SHADOW (SHAD) set option and/or the HIDE primary command.

NEXT/PREV

The NEXT (N) and PREV (P) commands are particularly useful when browsing an SMF dataset that contains many different base record types.

To navigate to the next occurrence of SMF Record Type *"nn"* type **`N SMF0`*`nn`*** e.g. **`N SMF014`**

To navigate to the previous occurrence of SMF Record Type *"nn"* type **`P SMF0`*`nn`*** e.g. **`P SMF014`**

ZoomW Key

The **"ZoomW"** key **(Shift-F5)** may be used from an SMF Browse session in order to display the focus record/segment as a single record formatted view.

The ZoomW key (which issues primary command "SDEZOOMW MAP") actually opens a separate window which will be right adjusted on the screen, provided the user is operating on a sufficiently large screen (e.g. 62 lines x 160 columns).

This is particularly useful when issued against a shadowed segment which then becomes visible in full detail in the new window.

Using a standard screen size (e.g. 32 lines x 80 columns) SELCOPYi does not operate in "windowed" display mode, in which case a new "full-screen" window which completely overlays the current display may be undesirable. In this case the user may prefer to use the MAP (FMT) and VFMT commands to switch display format. MAP may also be entered as a line-command in the prefix area of any record/segment/shadow line.

SELECT

The SELECT command opens a dialog used to tailor the display of focus record-type. This includes selection of the fields to be displayed, their order on screen, their column widths and "held" status.

Modifications may be temporary or permanent. Details of permanent modifications are saved in the Structure Definition Object (SDO).

VIEW

The VIEW command may be used to restrict visible record/segment types. Excluded record-types will be represented on screen by "shadow" lines.

Line-commands **"V", "V+"** and **"V-"** may also be entered into the prefix area.

PRINT / XMLGEN / CSVGEN / JSONGEN

The PRINT, XMLGEN (XML), CSVGEN (CSV) and JSONGEN (JSON) primary commands may be used to create various external charcater based versions of the currently browsed SMF data.

SMF Formatted Browse Utility (=13.1)

The SMF Formatted Browse Utility panel (ZZSGSMFB) is an interactive panel window, used to view a dataset containing SMF records.

The utility supports viewing either the **"SYS1.xxxx.MANn"** dataset format (VSAM ESDS with 4-byte "RDW" prefix) that is written to directly by SMF, or the dataset format written by the **SMF UNLOAD utility**(normally **RECFM=VBS** with no 4-byte "RDW" prefix).

Primary command SMFB provides a Command Line Interface (CLI) to the options on this panel.

SMF Features Further Information provides information about SELCOPYi's handling of SMF data.

The panel may be opened via the following:

- Select option 1. 'Browse' from the SMF option menu (=13).
- Execute the primary command "SMFB" with no parameters from any command line.
- Execute the line-command "SMFB /" from a file list window.

If no record selection options are specified then only the first 100 records are immediately loaded. Further groups of 100 records will be loaded (and unloaded) as needed, should you scroll down the file or execute a search-type command. Beyond a few hundred records, SELCOPYi will do its best to minimise the amount of storage used by unloading as many groups of 100 record as possible.

If record selection options are specified then the same technique applies but the group size starts at 1 and increases by 1 up to a maximum of 50. The intention of this is to minimize the amount of storage and I/O required in order to display each screen full of data.

Panel Input Fields

By default, field entries are populated with arguments and options that were entered the last time it was used.

`SMF Dataset:`
Input fields which together identify a single existing, sequential or VSAM data set, GDG file generation, HFS file or PDS/PDSE library member containing SMF records that is to be browsed.

`DSN/Path>`
Identifies the fully qualified data set name or an absolute or relative HFS file path.

Dataset names beginning with "." (dot) will be treated as though they actually begin with your own userid/hlq.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of an existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

This field may also be used to enter the **relative generation number** of a **GDG.** e.g. 0 or -10

Record Selection:
Fields which together identify criteria by which a subset of records from the file are selected for browse.

Please be aware that if any record selection options are specified then an immediate full pass of the SMF dataset will occur, and furthermore all selected records will be kept loaded in available storage for the whole duration of the browse session.

However, if no record selection options are specified then only the first 100 records are immediately loaded. Further groups of 100 records will be loaded (and unloaded) as needed, should you scroll down the file or execute a search-type command. Beyond a few hundred records, SELCOPYi will do its best to minimise the amount of storage used by unloading as many groups of 100 record as possible.

Type(s)> *type-subtype*
Specifies the SMF record types/subtypes to be selected.

A list of record type numbers may be provided, each separated by either blank or comma.

You may also specify a range of type numbers as **rr1:rr2** e.g. "60:69" is equivalent to "60,61,62,63,64,65,66,67,68,69".

To request subtype **ss** of record type **rr** you may specify either **rr-ss** or **rr#ss**

Lo-Date/Time>
The minimum timestamp for record selection in **yyyy/mm/dd hh:mm:ss.t** format, which may be specified in as much detail as necessary.
e.g. **2018/09/12** will be treated as **2018/09/12 00:00:00.0**

A date relative to the current day may also be supplied as **+/-nnn**
e.g. If today were **2016/03/05** then **-5** will be treated as **2016/02/29** (leap year).

Hi-Date/Time>
The maximum timestamp for record selection in **yyyy/mm/dd hh:mm:ss.t** format, which may be specified in as much detail as necessary.
e.g. **2018/09** will be treated as **2018/09/99 99:99:99.9**

A date relative to the current day may also be supplied as **+/-nnn**
e.g. If today were **2016/03/05** then **-5** will be treated as **2016/02/29** (leap year).

Input Limit>
The maximum number of records that may be read from the SMF Dataset for potential selection.

See Output Limit for further discussion.

Output Limit>
The maximum number of records that may be selected for browse.

If record selection is based on **multiple entries** in the Type(s)> field then this limit is applied per selected record type, provided that either:

1. The Find String>, User Id> and Job Name> selection options are all left blank, or
2. The Logic: option is set to **AND**.

An input limit or output limit of "0" (zero) is not sensible and so is totally ignored.

Examples ...

```
|    DSN/Path> CBL.SMF.GDG
|
|    Type(s)     > 14
|
|     Input Limit> 5000
|    Output Limit>
```

Only the first 5000 records will be read from "CBL.SMF.GDG" from which any number of type 14 records may be selected and those will be displayed in the browse session.

All selected records will be kept in storage for the duration of the browse session.

```
|    DSN/Path> CBL.SMF.GDG
|
|    Type(s)     > 14
|
|     Input Limit>
|    Output Limit> 50
```

Records will be read from "CBL.SMF.GDG" until 50 type 14 records have been selected.

If the first 50 records read are all type 14, then only 50 records will be read.

If there are less than 50 type 14 records in "CBL.SMF.GDG" then the whole file will be read regardless of how many records it contains.

```
|    DSN/Path> CBL.SMF.GDG
|
|    Type(s)     > 14
|
|     Input Limit> 5000
|    Output Limit> 50
```

No more than 5000 records will be read from "CBL.SMF.GDG".

But if the first 100 records include 50 type 14 records, then only 100 records will be read.

```
|    DSN/Path> CBL.SMF.GDG
|
|    Type(s)     > 14 15 30
|
|     Input Limit>
|    Output Limit> 10
```

Records will be read from "CBL.SMF.GDG" until up to 10 of each of the specified record types (14, 15 and 30) are found (see above caveats).

Therefore, a maximum of 30 records will be selected in this particular case.

Find String>

Specifies one or more strings to be located within each SMF record.

Multiple strings may be specified, each **separated by a comma**. A record containing any one of the strings will be selected. e.g.

```
 Find String> SYS1.MACLIB, SYS1.MIGLIB, SYS1.MODGEN, SYS1.MSGEN
```

Each string may be specified:

- Using an unquoted character literal. e.g. ABC will match any record containing 'ABC' (case-insensitive).
- Using a quoted character literal. e.g. 'ABC ' will match any record containing 'ABC ' (case-insensitive).
- Using a quoted character literal prefixed with "C". e.g. C'aBc ' will match any record containing 'aBc ' (case-sensitive).
- Using a quoted hex literal prefixed with "X". e.g. X'81C2C340' will match any record containing 'aBc ' (case-sensitive).
- Using the single-character wildcard '%' (percent). e.g. 'ABC%DEF' will match any record containing 'ABC' followed by any other single character followed by 'DEF'. e.g.

  ```
  Find String> SYS1.M%%LIB
  ```

- Using the multiple-character wildcard '*' (asterisk). e.g. 'ABC*DEF' will match any record containing 'ABC' followed by any number of other characters followed by 'DEF'. e.g.

  ```
  Find String> SYS1.M*LIB
  ```

The panel entry field for this option is 45 bytes in length. To specify values longer than this, first type **"EXPAND"** on the command line, then place your cursor in the **Find String>** entry field before pressing ENTER. A "Text-Edit" window will be displayed allowing you to enter long values over multiple lines.

User Id>

Indicates that all SMF Record-Types known to contain a User Id field **(zUserId)** will be tested for a match with any one of the listed user id specifications.

UserId specifications should be **separated by a comma**.

UserId may be specified:

◊ Using an unquoted or quoted literal. e.g. ABC will match exactly on User Id 'ABC '.

◊ Using single-character wildcard '%' (percent). e.g. 'ABC%DEF' will match any User Id beginning with 'ABC' followed by any other single character followed by 'DEF'.

◊ Using multiple-character wildcard '*' (asterisk). e.g. '*1' will match any User Id ending with '1'.

The following SMF Record-types are known to contain a User Id field.

004	010	017	025	034	040	062	065	068	110
005	014	018	026	035	060	063	066	069	118
006	015	020	030	036	061	064	067	080	119

Job Name>

Indicates that all SMF Record-Types known to contain a Job Name field **(zJobName)** will be tested for a match with any one of the listed job name specifications.

JobName specifications should be **separated by a comma**.

JobName may be specified:

◊ Using an unquoted or quoted literal. e.g. ABC will match exactly on Job Name 'ABC '.

◊ Using single-character wildcard '%' (percent). e.g. 'ABC%DEF' will match any Job Name beginning with 'ABC' followed by any other single character followed by 'DEF'

◊ Using multiple-character wildcard '*' (asterisk). e.g. '*1' will match any Job Name ending with '1'

The following SMF Record-types are known to contain a Job Name field.

004	010	017	025	034	040	061	064	067	080
005	014	018	026	035	042	062	065	068	110
006	015	020	030	036	060	063	066	069	118

System Id>

Indicates that all SMF Record-Types will be tested for a match with any one of the listed System Id specifications.

System Id specifications should be **separated by a comma**.

System Id may be specified:

◊ Using an unquoted or quoted literal. e.g. ABC will exactly match System Id 'ABC '.

◊ Using single-character wildcard '%' (percent). e.g. 'AB%Z' will match any System Id beginning with 'AB' followed by any other single character followed by 'Z'

◊ Using multiple-character wildcard '*' (asterisk). e.g. '*Z' will match any System Id ending with 'Z'

All SMF Record-types contain a System Id field (zSID).

Logic: AND | OR

OR indicates that SMF records will be selected when (apart from Lo- and Hi-Date/Time range) **ANY** one of the subset options is satisfied.

AND indicates that SMF records will be selected only when **ALL** subset options are satisfied. Note that within individual subset options such as **"Type(s)", "Find String", "User Id"** and **"System Id"** where the user may supply multiple items, these items are still OR'd i.e. satisfied if any one of the items is matched.

Options:

Format> ONLINE | OFFLINE

ONLINE indicates that the SMF dataset is in the format as written directly by SMF. These are typically **SYS1.xxxx.MAN1/2/3/etc** datasets (SELCOPYi does not support browse of SMF records directly from the System Logger).

Browse of online datasets should be undertaken with caution as SELCOPYi will keep a SHR enqueue on the file for the duration of the browse session, which may interfere with SMF's archiving process.

ONLINEdatasets include a **4-byte record descriptor word (RDW) prefix** at the start of each record, so record-type field mapping must be offset by this amount.

OFFLINE indicates that the SMF dataset is the format as written by the SMF archiving tool (IFASMFDP) which does not include a 4-byte (RDW) record prefix.

Layout> F (Full)

Use comprehensive mapping for supported record types.

Each supported SMF record/sub-type will be assigned a separate layout **SMF*nnn_xxx_yyyy_zzz***, where *nnn* represents the variable record-type number, and *xxx_yyyy_zzz* is the short description.
e.g. **SMF014_INPUT_or_RDBACK_Dataset**

SMF records that comprise potentially repeating groups will be mapped over several logical Primary/Secondary Segments.

Any selected records whose SMF record type is currently unsupported will be mapped using the "Basic Layout" record type **SMFnnn.**

Layout> B (Basic)
All records will be mapped using the basic layout **SMFnnn**. (Note that **nnn** is a literal and does not represent the variable record-type number.)

The basic layout includes the common header fields only (e.g. SMF record/sub-type, timestamp), followed by the field **"Rest"** which displays the tail end of the record as uninterpreted character data.

Use HEX ON to display hexadecimal representations.

Segment> SHOW | SHADOW
Applicable only when **"Full Layout"** is selected and affects only SMF records that are mapped using Base/Secondary segments.

SHOW indicates that secondary segments should initially display in full detail.

SHADOW indicates that secondary segments should initially display as shadow lines.

Note that while secondary segments are shadowed (thereby occupying much less screen space) the detail for any single shadowed segment may be displayed in a separate window by placing the cursor on the shadow line and pressing the **"ZoomW"** key **(Shift-F5).**

The shadow lines themselves may be suppressed using the SHADOW (SHAD) set option and/or the HIDE primary command.

Regardless of your initial setting for this option, the VBASE primary command may be used to switch back and forth throughout your browse session.

Menu-Bar Options:

Upd-Layouts
Select this option only when you wish to refresh your own copy of SMF Record layout structures from the SELCOPYi supplied defaults.

This will typically be because Compute (Bridgend) Ltd have added layouts for previously unmapped SMF record types, or made enhancements/corrections to existing ones.

The SMF-Browse utility will normally automatically detect CBL supplied updates and carry out this operation unprompted.

SMF Extract Utility (=13.2)

The SMF Extract Utility panel (ZZSGSMFX) is an interactive panel window, used to extract a selection of SMF records from one or more input files and copy them to a single output file.

The utility may be run in the foreground directly from the panel. Alternatively, the panel will generate JCL to run the process in batch.

Primary command SMFEXTRC provides a Command Line Interface (CLI) to the SMF Extract Utility and the panel itself includes an option to generate CLI corresponding to the currently entered panel field values.

SMF Features Further Information provides information about SELCOPYi's handling of SMF data.

The panel may be opened via the following:

- Select option 2. 'Extract' from the SMF option menu (=13).
- Execute the primary command "SMFEXTRC" with no parameters from any command line.
- Execute the line-command "SMFEXTRC" from a file list window.

Panel Input Fields

By default, field entries are populated with arguments and options that were entered the last time it was used.

SMF Source Dataset(s):
Input fields which together identify up to three existing, sequential or VSAM data sets, GDG file generations, HFS files or PDS/PDSE library members from which SMF records are to be extracted.

Should more than three input datasets be required then select Runtype> B to generate a batch job or Runtype> C to generate a command line, both of which may be edited to specify an unlimited number of input datasets.

Also, please note that if a **GDG base** is specified with no relative generation number (via the Member input field) then all generations will be input, starting with the oldest.

DSN/Path>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

Dataset names beginning with "." (dot) will be treated as though they actually begin with your own userid/hlq.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of an existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

This field may also be used to enter the **relative generation number** of a **GDG.** e.g. 0 or -10.

SMF Extract Dataset:
Input fields which together identify a single sequential or VSAM data set, GDG file generation, HFS file or PDS/PDSE library member that is to receive the extracted SMF records.

DSN/Path>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

Dataset names beginning with "." (dot) will be treated as though they actually begin with your own userid/hlq.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of an existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

This field may also be used to enter the **relative generation number** of a **GDG.** e.g. 0 or -10.

Record Selection:
Fields which together identify criteria by which a subset of records from the file are selected for browse.

Type(s)> *type-subtype*
Specifies the SMF record types/subtypes to be selected.

A list of record type numbers may provided, each separated by either blank or comma.

You may also specify a range of type numbers as **rr1:rr2** e.g. "60:69" is equivalent to "60,61,62,63,64,65,66,67,68,69".

To request subtype **ss** of record type **rr** you may specify either **rr-ss** or **rr#ss**.

Lo-Date/Time>
The minimum timestamp for record selection in **yyyy/mm/dd hh:mm:ss.t** format, which may be specified in as much detail as necessary.
e.g. **2018/09/12** will be treated as **2018/09/12 00:00:00.0**

A date relative to the current day may also be supplied as **+/-nnn**
e.g. If today were **2016/03/05** then **-5** will be treated as **2016/02/29** (leap year).

Hi-Date/Time>
The maximum timestamp for record selection in **yyyy/mm/dd hh:mm:ss.t** format, which may be specified in as much detail as necessary.
e.g. **2018/09** will be treated as **2018/09/99 99:99:99.9**

A date relative to the current day may also be supplied as **+/-nnn**
e.g. If today were **2016/03/05** then **-5** will be treated as **2016/02/29** (leap year).

Input Limit>
The maximum number of records that may be read from the SMF Dataset for potential selection.

See Output Limit for further discussion.

Output Limit>
The maximum number of records that may be selected for output.

If record selection is based on **multiple entries** in the Type(s)> field then this limit is applied per selected record type, provided that either:

1. The Find String>, User Id> and Job Name> selection options are all left blank, or

2. The Logic: option is set to **AND**.

An input limit or output limit of "0" (zero) is not sensible and so is totally ignored.

Examples ...

```
|    DSN/Path> CBL.SMF.GDG
|
|    Type(s)     > 14
|
|     Input Limit> 5000
|    Output Limit>
```

Only the first 5000 records will be read from "CBL.SMF.GDG" from which any number of type 14 records may be selected for extraction.

```
|    DSN/Path> CBL.SMF.GDG
|
|    Type(s)     > 14
|
|     Input Limit>
|    Output Limit> 50
```

Records will be read from "CBL.SMF.GDG" until 50 type 14 records have been selected.

If the first 50 records read are all type 14, then only 50 records will be read.

If there are less than 50 type 14 records in "CBL.SMF.GDG" then the whole file will be read regardless of how many records it contains.

```
|    DSN/Path> CBL.SMF.GDG
|
|    Type(s)     > 14
|
|     Input Limit> 5000
|    Output Limit> 50
```

No more than 5000 records will be read from "CBL.SMF.GDG".

But if the first 100 records include 50 type 14 records, then only 100 records will be read.

```
|    DSN/Path> CBL.SMF.GDG
|
|    Type(s)     > 14 15 30
|
|     Input Limit>
|    Output Limit> 10
```

Records will be read from "CBL.SMF.GDG" until up to 10 of each of the specified record types (14, 15 and 30) are found (see above caveats).

Therefore, a maximum of 30 records will be selected in this particular case.

Find String>

Specifies one or more strings to be located within each SMF record.

Multiple strings may be specified, each **separated by a comma**. A record containing any one of the strings will be selected. e.g.

```
Find String> SYS1.MACLIB, SYS1.MIGLIB, SYS1.MODGEN, SYS1.MSGEN
```

Each string may be specified:

- Using an unquoted character literal. e.g. ABC will match any record containing 'ABC' (case-insensitive).
- Using a quoted character literal. e.g. 'ABC ' will match any record containing 'ABC ' (case-insensitive).
- Using a quoted character literal prefixed with "C". e.g. C'aBc ' will match any record containing 'aBc ' (case-sensitive).
- Using a quoted hex literal prefixed with "X". e.g. X'81C2C340' will match any record containing 'aBc ' (case-sensitive).
- Using the single-character wildcard '%' (percent). e.g. 'ABC%DEF' will match any record containing 'ABC' followed by any other single character followed by 'DEF'. e.g.

  ```
  Find String> SYS1.M%%LIB
  ```

- Using the multiple-character wildcard '*' (asterisk). e.g. 'ABC*DEF' will match any record containing 'ABC' followed by any number of other characters followed by 'DEF'. e.g.

```
Find String> SYS1.M*LIB
```

The panel entry field for this option is 45 bytes in length. To specify values longer than this, first type **"EXPAND"** on the command line, then place your cursor in the **Find String>** entry field before pressing ENTER. A "Text-Edit" window will be displayed allowing you to enter long values over multiple lines.

User Id>

Indicates that all SMF Record-Types known to contain a User Id field **(zUserId)** will be tested for a match with any one of the listed user id specifications.

UserId specifications should be **separated by a comma**.

UserId may be specified:

◊ Using an unquoted or quoted literal. e.g. ABC will match exactly on User Id 'ABC '.

◊ Using single-character wildcard '%' (percent). e.g. 'ABC%DEF' will match any User Id beginning with 'ABC' followed by any other single character followed by 'DEF'.

◊ Using multiple-character wildcard '*' (asterisk). e.g. '*1' will match any User Id ending with '1'.

The following SMF Record-types are known to contain a User Id field.

004	010	017	025	034	040	062	065	068	110
005	014	018	026	035	060	063	066	069	118
006	015	020	030	036	061	064	067	080	119

Job Name>

Indicates that all SMF Record-Types known to contain a Job Name field **(zJobName)** will be tested for a match with any one of the listed job name specifications.

JobName specifications should be **separated by a comma**.

JobName may be specified:

◊ Using an unquoted or quoted literal. e.g. ABC will match exactly on Job Name 'ABC '.

◊ Using single-character wildcard '%' (percent). e.g. 'ABC%DEF' will match any Job Name beginning with 'ABC' followed by any other single character followed by 'DEF'

◊ Using multiple-character wildcard '*' (asterisk). e.g. '*1' will match any Job Name ending with '1'

The following SMF Record-types are known to contain a Job Name field.

004	010	017	025	034	040	061	064	067	080
005	014	018	026	035	042	062	065	068	110
006	015	020	030	036	060	063	066	069	118

System Id>

Indicates that all SMF Record-Types will be tested for a match with any one of the listed System Id specifications.

System Id specifications should be **separated by a comma**.

System Id may be specified:

◊ Using an unquoted or quoted literal. e.g. ABC will exactly match System Id 'ABC '.

◊ Using single-character wildcard '%' (percent). e.g. 'AB%Z' will match any System Id beginning with 'AB' followed by any other single character followed by 'Z'

◊ Using multiple-character wildcard '*' (asterisk). e.g. '*Z' will match any System Id ending with 'Z'

All SMF Record-types contain a System Id field (zSID).

Logic: AND | OR

OR indicates that SMF records will be selected when (apart from Lo- and Hi-Date/Time range) **ANY** one of the subset options is satisfied.

AND indicates that SMF records will be selected only when **ALL** subset options are satisfied. Note that within individual subset options such as **"Type(s)", "Find String", "User Id"** and **"Job Name"** where the user may supply multiple items, these items are still OR'd i.e. satisfied if any one of the items is matched.

Options:

Format> ONLINE | OFFLINE

ONLINE indicates that the SMF dataset is in the format as written directly by SMF. These are typically **SYS1.xxxx.MAN1/2/3/etc** datasets (SELCOPYi does not support browse of SMF records directly from the System Logger).

Browse of online datasets should be undertaken with caution as SELCOPYi will keep a SHR enqueue on the file for the duration of the browse session, which may interfere with SMF's archiving process.

ONLINEdatasets include a **4-byte record descriptor word (RDW) prefix** at the start of each record, so record-type field mapping must be offset by this amount.

OFFLINE indicates that the SMF dataset is the format as written by the SMF archiving tool (IFASMFDP) which does not include a 4-byte (RDW) record prefix.

Run Type> F | B | C
F indicates that immediate foreground execution is required as soon as the **ENTER** key is pressed.

B indicates that JCL should be produced for submission to batch.

C indicates that command line interface should be produced. The SMFEXTRC primary command is displayed in a Text-Edit window in a format suitable for execution using the ACTION key (Shift-F4) ready to be copied into your HOME file (=4).

Append> APP | blank
APP indicates that selected records should be appended to the output dataset.

SMF Report Utility (=13.2)

The SMF Report Utility panel (ZZSGSMFR) is an interactive panel window, used to generate a printable report from a dataset containing SMF records.

The layout of your desired report should be specified using control statements saved in the Report Definition dataset.

The report produced will consist mainly of data extracted from a list of data fields from a single SMF Record-Type. The report can reference fields from both Primary (Base) and Secondary segments.

A user definable heading will be printed at the top of each page, followed by user definable columns heading for each selected field.

Grand totals will automatically be printed for any selected field containing integer data, and sub-totals will also be printed if a sort/control-break has been requested.

See SMF Report Definition Control Statements for full details of supported report definition control statements.

The utility supports sourcing SMF data from either the **"SYS1.xxxx.MANn"** dataset format (VSAM ESDS with 4-byte "RDW" prefix) that is written to directly by SMF, or the dataset format written by the **SMF UNLOAD utility**(normally **RECFM=VBS** with no 4-byte "RDW" prefix).

Primary command REPORT provides a Command Line Interface (CLI) to the options on this panel.

SMF Features Further Information provides information about SELCOPYi's handling of SMF data.

The panel may be opened via the following:

- Select option 3. 'Report' from the SMF option menu (=13).
- Execute the primary command **REPORT** with no parameters from any command line and select option 4. 'SMF Report' from the PRINT/REPORT Features menu.

Sample Report Definition

```
<----1----+----2----+----3----+----4----+----5----+----6
TITLE:
 Job/DD EXCPs Report (from SMF Type 30 Subtype 5)

COLUMNS:
 SMF030_Identification.zJOBNAME            'Job Name'
 SMF030_Identification.zSIT                'Job Start'
 SMF030_Common_Address_Space_Work.ZTME     'Job End'
 SMF030_EXCP.zDDN                          'DDName'
 SMF030_EXCP.zBLK                          'EXCP Blks'

REPEAT:
 SMF030_EXCP
* * * End of File * * *
```

Figure 310. Report Control Statements.

Sample Report Output

```
<---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
12018/09/14 12:04   Job/DD EXCPs Report (from SMF Type 30 Subtype 5)  PAGE    1

 Job Name Job Start               Job End                 DDName   EXCP Blks
 -------- ----------------------- ----------------------- -------- -----------
 SMFCLEAR 2018/09/04 01:09:18.03 2018/09/04 01:09:29.73 INDD1          14400
                                                        DUMPOUT           786
                                                        SYSPRINT            0
                                                        SYSIN               2
 SMFCLEAR 2018/09/04 02:55:15.33 2018/09/04 02:55:26.97 INDD1          14400
                                                        DUMPOUT           783
                                                        SYSPRINT            0
                                                        SYSIN               2
 SMFCLEAR 2018/09/04 04:41:32.55 2018/09/04 04:41:45.63 INDD1          14400
                                                        DUMPOUT           786
                                                        SYSPRINT            0
                                                        SYSIN               2
 SMFCLEAR 2018/09/04 06:27:59.30 2018/09/04 06:28:09.90 INDD1          14400
                                                        DUMPOUT           788
                                                        SYSPRINT            0
                                                        SYSIN               2
--------------------------- 6178 line(s) not displayed --------------------------
12018/09/14 12:04   Job/DD EXCPs Report (from SMF Type 30 Subtype 5)  PAGE  113
 SMFCLEAR 2018/09/07 09:54:14.26 2018/09/07 09:54:25.94 INDD1          14400
                                                        DUMPOUT           796
                                                        SYSPRINT            0
                                                        SYSIN               2
 SMFCLEAR 2018/09/07 11:31:38.69 2018/09/07 11:31:51.49 INDD1          14400
                                                        DUMPOUT           792
                                                        SYSPRINT            0
                                                        SYSIN               2
 SMFCLEAR 2018/09/07 13:08:53.59 2018/09/07 13:09:05.39 INDD1          14400
                                                        DUMPOUT           786
                                                        SYSPRINT            0
                                                        SYSIN               2


                                                                  ===========
                                                                      1139389
                                                                  ===========
* * * End of File * * *
```

Figure 310. Report Output.

Panel Input Fields

By default, field entries are populated with arguments and options that were entered the last time it was used.

Report Definition:
Input fields which together identify a single, sequential or VSAM data set, GDG file generation, HFS file or PDS/PDSE library member that contains (or will contain) the SMF report definition control statements.

Execute primary command **EDIT** (or **E**) to edit the report definition file.

DSN/Path>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

Dataset names beginning with "." (dot) will be treated as though they actually begin with a high level qualifier equal to the user's own userid.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>
If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of an existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

SMF Dataset:
Input fields which together identify a single existing, sequential or VSAM data set, GDG file generation, HFS file or PDS/PDSE library member containing SMF records that are to be reported upon.

Execute primary command **INPUT** (or **I**) to browse the SMF dataset using full layout.

Type primary command **BINPUT** (or **BI**) to browse the SMF dataset using basic layout.

DSN/Path>
Identifies the fully qualified data set name or an absolute or relative HFS file path.

Dataset names beginning with "." (dot) will be treated as though they actually begin with a high level qualifier equal to the user's own userid.

A selectable list of data sets or HFS files will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent).

Member>

If the DSN/Path> field contains the DSN of a PDS/PDSE library, then this field may specify the name of an existing member within that library.

A selectable list of members will be presented if the entered value contains wildcards characters "*" (asterisk) or "%" (percent), or is blanked out.

This field may also be used to enter the **relative generation number** of a **GDG.** e.g. 0 or -10.

Record Selection:

Fields which together identify criteria by which a subset of records from the file are selected for the report.

Type(s)> *type-subtype*

Specifies the SMF record types/subtypes to be selected.

A list of record type numbers may provided, each separated by either blank or comma.

You may also specify a range of type numbers as **rr1:rr2** e.g. "60:69" is equivalent to "60,61,62,63,64,65,66,67,68,69".

To request subtype **ss** of record type **rr** you may specify either **rr-ss** or **rr#ss**

Lo-Date/Time>

The minimum timestamp for record selection in **yyyy/mm/dd hh:mm:ss.t** format, which may be specified in as much detail as necessary. e.g.
2018/09/12 will be treated as **2018/09/12 00:00:00.0**

A date relative to the current day may also be supplied as **+/-nnn**
e.g. If today were **2016/03/05** then **-5** will be treated as **2016/02/29** (leap year).

Hi-Date/Time>

The maximum timestamp for record selection in **yyyy/mm/dd hh:mm:ss.t** format, which may be specified in as much detail as necessary. e.g.
2018/09 will be treated as **2018/09/99 99:99:99.9**

A date relative to the current day may also be supplied as **+/-nnn**
e.g. If today were **2016/03/05** then **-5** will be treated as **2016/02/29** (leap year).

Input Limit>

The maximum number of records that may be read from the SMF Dataset for potential selection.

See Output Limit for further discussion.

Output Limit>

The maximum number of records that may be selected for reporting.

If record selection is based on **multiple entries** in the Type(s)> field then this limit is applied per selected record type, provided that either:

1. The Find String>, User Id> and Job Name> selection options are all left blank, or
2. The Logic: option is set to **AND**.

An input limit or output limit of "0" (zero) is not sensible and so is totally ignored.

Examples ...

```
|    DSN/Path> CBL.SMF.GDG
|
|    Type(s)     > 14
|
|     Input Limit> 5000
|    Output Limit>
```

Only the first 5000 records will be read from "CBL.SMF.GDG" from which any number of type 14 records may be selected and those will be reported upon.

```
|    DSN/Path> CBL.SMF.GDG
|
|    Type(s)     > 14
|
|     Input Limit>
|    Output Limit> 50
```

Records will be read from "CBL.SMF.GDG" until 50 type 14 records have been selected.

If the first 50 records read are all type 14, then only 50 records will be read.

If there are less than 50 type 14 records in "CBL.SMF.GDG" then the whole file will be read regardless of how many records it contains.

```
|    DSN/Path> CBL.SMF.GDG
|
|    Type(s)     > 14
|
|     Input Limit> 5000
|    Output Limit> 50
```

No more than 5000 records will be read from "CBL.SMF.GDG".

But if the first 100 records include 50 type 14 records, then only 100 records will be read.

```
|    DSN/Path> CBL.SMF.GDG
|
|    Type(s)     > 14 15 30
|
|     Input Limit>
|    Output Limit> 10
```

Records will be read from "CBL.SMF.GDG" until up to 10 of each of the specified record types (14, 15 and 30) are found (see above caveats).

Therefore, a maximum of 30 records will be selected in this particular case.

Find String>

Specifies one or more strings to be located within each SMF record.

Note that this option is ignored if your report definition control statements include a FILTER section, which should be coded to handle all your selection requirements.

Multiple strings may be specified, each **separated by a comma**. A record containing any one of the strings will be selected. e.g.

```
Find String> SYS1.MACLIB, SYS1.MIGLIB, SYS1.MODGEN, SYS1.MSGEN
```

Each string may be specified:

- Using an unquoted character literal. e.g. ABC will match any record containing 'ABC' (case-insensitive).
- Using a quoted character literal. e.g. 'ABC ' will match any record containing 'ABC ' (case-insensitive).
- Using a quoted character literal prefixed with "C". e.g. C'aBc ' will match any record containing 'aBc ' (case-sensitive).
- Using a quoted hex literal prefixed with "X". e.g. X'81C2C340' will match any record containing 'aBc ' (case-sensitive).
- Using the single-character wildcard '%' (percent). e.g. 'ABC%DEF' will match any record containing 'ABC' followed by any other single character followed by 'DEF'. e.g.

  ```
  Find String> SYS1.M%%LIB
  ```

- Using the multiple-character wildcard '*' (asterisk). e.g. 'ABC*DEF' will match any record containing 'ABC' followed by any number of other characters followed by 'DEF'. e.g.

  ```
  Find String> SYS1.M*LIB
  ```

The panel entry field for this option is 45 bytes in length. To specify values longer than this, first type **"EXPAND"** on the command line, then place your cursor in the **Find String>** entry field before pressing ENTER. A "Text-Edit" window will be displayed allowing you to enter long values over multiple lines.

User Id>

Indicates that all SMF Record-Types known to contain a User Id field **(zUserId)** will be tested for a match with any one of the listed user id specifications.

UserId specifications should be **separated by a comma**.

Note that this option is ignored if your report definition control statements include a FILTER section, which should be coded to handle all your selection requirements.

UserId may be specified:

◊ Using an unquoted or quoted literal. e.g. ABC will match exactly on User Id 'ABC '.

◊ Using single-character wildcard '%' (percent). e.g. 'ABC%DEF' will match any User Id beginning with 'ABC' followed by any other single character followed by 'DEF'.

◊ Using multiple-character wildcard '*' (asterisk). e.g. '*1' will match any User Id ending with '1'.

The following SMF Record-types are known to contain a User Id field.

004	010	017	025	034	040	062	065	068	110
005	014	018	026	035	060	063	066	069	118
006	015	020	030	036	061	064	067	080	119

Job Name>

Indicates that all SMF Record-Types known to contain a Job Name field **(zJobName)** will be tested for a match with any one of the listed job name specifications.

JobName specifications should be **separated by a comma**.

Note that this option is ignored if your report definition control statements include a FILTER section, which should be coded to handle all your selection requirements.

JobName may be specified:

◊ Using an unquoted or quoted literal. e.g. ABC will match exactly on Job Name 'ABC '.

◊ Using single-character wildcard '%' (percent). e.g. 'ABC%DEF' will match any Job Name beginning with 'ABC' followed by any other single character followed by 'DEF'

◊ Using multiple-character wildcard '*' (asterisk). e.g. '*1' will match any Job Name ending with '1'

The following SMF Record-types are known to contain a Job Name field.

004	010	017	025	034	040	061	064	067	080
005	014	018	026	035	042	062	065	068	110
006	015	020	030	036	060	063	066	069	118

System Id>

Indicates that all SMF Record-Types will be tested for a match with any one of the listed System Id specifications.

System Id specifications should be **separated by a comma**.

System Id may be specified:

◊ Using an unquoted or quoted literal. e.g. ABC will exactly match System Id 'ABC '.

◊ Using single-character wildcard '%' (percent). e.g. 'AB%Z' will match any System Id beginning with 'AB' followed by any other single character followed by 'Z'

◊ Using multiple-character wildcard '*' (asterisk). e.g. '*Z' will match any System Id ending with 'Z'

All SMF Record-types contain a System Id field (zSID).

Logic: AND | OR

OR indicates that SMF records will be selected when (apart from Lo- and Hi-Date/Time range) **ANY** one of the subset options is satisfied.

AND indicates that SMF records will be selected only when **ALL** subset options are satisfied. Note that within individual subset options such as **"Type(s)", "Find String", "User Id"** and **"Job Name"** where the user may supply multiple items, these items are still OR'd i.e. satisfied if any one of the items is matched.

Options:

Format> ONLINE | OFFLINE

ONLINE indicates that the SMF dataset is in the format as written directly by SMF. These are typically **SYS1.xxxx.MAN1/2/3/etc** datasets (SELCOPYi does not support reporting on SMF records directly from the System Logger).

Reporting on online datasets should be undertaken with caution as SELCOPYi will keep a SHR enqueue on the file for the duration of the report, which may interfere with SMF's archiving process.

ONLINE datasets include a **4-byte record descriptor word (RDW) prefix** at the start of each record, so record-type field mapping must be offset by this amount.

OFFLINE indicates that the SMF dataset is the format as written by the SMF archiving tool (IFASMFDP) which does not include a 4-byte (RDW) record prefix.

Run Type> B | C | F

B indicates that JCL should be produced for submission to batch.

C indicates that command line interface should be produced. The REPORT primary command is displayed in a Text-Edit window in a format suitable for execution using the ACTION key (Shift-F4) ready to be copied into your HOME file (=4).

F indicates that immediate foreground execution is required as soon as the **ENTER** key is pressed.

In this case the report output will be collected in-storage and displayed in a Text-Edit window. The output will not be saved to disk but the user may enter the CREATE or REPLACE commands to do so.

If the expected report output is larger than your available foreground region then you should choose the B option to generate a batch job which will write the report to the DD name **SDEPRINT**.

Output Type> P=Print | C=CSV | J=JSON | X=XML | B=Browse

P requests standard **Print** output with page headings, optional sort, control breaks and totals/sub-totals.

C requests **Comma Separated Variable (CSV)** output suitable for loading into various external formats such as **DB2**. The first row will contain the column headings as defined by your report definition file.
Defined control breaks and totals/sub-totals are all ignored for **CSV** output.

J requests **Java Script Object Notation (JSON)** output suitable for loading into various external formats such as a web page.
Defined control breaks and totals/sub-totals are all ignored for **JSON** output.

X requests **eXtensible Markup Language (XML)** output suitable for loading into various external formats including directly into **Microsoft Excel.**
Defined control breaks and totals/sub-totals are all ignored for **XML** output.

B requests an online **Browse** session is started with the format of the displayed data controlled by the report definition. i.e. fields selected from various primary/secondary segments will all appear on a single line.

The fields will be displayed grouped together by their source segment, so not strictly in accordance with your report definition.

This option will also define a permanent user default display format for the primary record type involved, meaning any future online browse will display only the selected fields. To revert to the default just type "SEL *" on the command line of the browse session.
Defined sort, control breaks and totals/sub-totals are all ignored for **BROWSE** output.

Page Depth> nnn

Specifies the number of lines printed per page.

If left blank (or specified as zero) then the user's own prevailing Data-Edit PAGEDEPTH value will be used.

Type **"SD Q PAGEDEPTH"** to query your current Data-Edit pagedepth value.

Type **"SD SET PAGEDEPTH nnn"** to set your current Data-Edit pagedepth value.

SMF Report Definition Control Statements

The following description of Report Definition control statements is valid and may be used to generate SMF report output. However, a more comprehensive and detailed description of the report definition statements and the overall report generation operation may be found in the "***REPORT Utility Manual***".

SMF Report Definition Sections

The SMF Report Utility is used to generate a printable report from a dataset containing SMF records.

The layout of desired report must be specified using control statements saved in a dataset which is split into sections e.g. **COLUMNS:**

Each section marker (xxxxx:) will be followed by one or more control statements.

The columns section for instance is followed by a line specifying each column to appear in the report. The order of the lines determines the order of the columns on the report page.

The report produced will consist mainly of data extracted from a list of data fields from a single SMF Record-Type. The report can reference fields from both Primary (Base) and Secondary segments.

A user definable heading will be printed at the top of each page, followed by user definable column headings for each selected field.

Grand totals will automatically be printed for any selected field containing valid numeric data, and **sub-totals** will also be printed if a sort/control-break has been requested.

Report definition lines starting with *asterisk* ("*") will be treated as **comment** data, but comments are not supported in the *TITLE:* section.

Example report definition.

```
      <---+----1----+----2----+----3----+----4----+----5----+----6----+---
00001 TITLE:
00002
00003     *** Selective SMF 014 Report
00004           (sorted by Job+DSN) ***
00005
00006 COLUMNS:
00007   SMF014_INPUT_or_RDBACK_Dataset.zJobName   'Job Name'
00008   SMF014_INPUT_or_RDBACK_Dataset.zRST       'Reader Timestamp'  16
00009   SMF014#3_Step_Info.zSPN                   'Step'
00010   SMF014#3_Step_Info.zPGN                   'PgmName'
00011   SMF014_INPUT_or_RDBACK_Dataset.SMFTIOE5   'DDName'
00012
00013   SMF014_INPUT_or_RDBACK_Dataset.DSN        'Dataset Name'      31
00014   SMF014_INPUT_or_RDBACK_Dataset.SMFEXCP(1) 'EXCPs'              6 R
00015   SMF014_INPUT_or_RDBACK_Dataset.VOLS       'VolSer'              6
00016   SMF014_INPUT_or_RDBACK_Dataset.RECFM      'Fmt'                 3
00017   SMF014_INPUT_or_RDBACK_Dataset.RECL       'Lrecl'               5 R
00018   SMF014_INPUT_or_RDBACK_Dataset.BLKSZ      'BlkSz'               5 R
00019   SMF014_INPUT_or_RDBACK_Dataset.Disp
00020   SMF014_INPUT_or_RDBACK_Dataset.Shr
00021
00022 SORT:
00023   SMF014_INPUT_or_RDBACK_Dataset.zJobName
00024   SMF014_INPUT_or_RDBACK_Dataset.DSN
00025
00026 BREAK:
00027   SMF014_INPUT_or_RDBACK_Dataset.zJobName
00028
00029 TOTALS:
00030   SMF014_INPUT_or_RDBACK_Dataset.SMFEXCP(1)
00031
00032 BLANKWHENZERO:
00033   SMF014_INPUT_or_RDBACK_Dataset.SMFEXCP(1)
00034
00035 FILTER:
00036 * Select if DDNAME not SYSxxxx and DSN begins "CBL", "JGE" or "NBJ".
00037
00038           SMF014_INPUT_or_RDBACK_Dataset.SMFTIOE5 \>> 'SYS'
00039  and (    SMF014_INPUT_or_RDBACK_Dataset.DSN       >> 'CBL'
00040        or SMF014_INPUT_or_RDBACK_Dataset.DSN       >> 'JGE'
00041        or SMF014_INPUT_or_RDBACK_Dataset.DSN       >> 'NBJ'
00042      )
00043 * * * End of File * * *
```

Figure 310. Report Definition.

TITLE: (Optional)
Data that follows "TITLE:" section header defines the title of the report which will appear centralised on the first line of each page.

The report title may span multiple lines in the report definition file, and immediately follow the section header (you don't need to start a new line).

The date/time in **"yyyy/mm/dd hh:mm"** format will automatically be added (left adjusted) on the report page heading line.

Right adjusted on this line will be **page number**.

Comments (lines begining with asterisk) are not supported within the TITLE: section and will be treated as part of the report title.

COLUMNS: (Required)
Each line following defines a report column.

◊ **Token 1** on the line identifies the SMF **record-type** and **field name** in **"rectype.field"** notation. e.g.

SMF014_INPUT_or_RDBACK_Dataset.*zTME*

Where **SMF014_INPUT_or_RDBACK_Dataset** is the record-type and *zTME* is the field name.

It is also permissable to refer to fields within Secondary segments that belong to the **Primary** (or Base) segment. e.g.

SMF014#3_Step_Info.*zPGN*

Restrictions:

· Only one "Base" segment may be be reported upon.
· At least one field must be selected from that "Base" segment.

◊ **Token 2** is optional. It defines a **column heading** which must be supplied in quotes.
e.g. **'RACF user ID'**

If no heading is supplied then the field name is used. e.g. "zRUD"

To supply subsequent options (column width etc) without overriding the column heading just include a null heading by supply two simultaneous quotes. e.g.

SMF100_Prod_Standard.*zIFCIDText* '' 20

◊ **Token 3** is optional. It defines a **column width** override allowing you to truncate a long field or pad (with blanks) a short field.

◊ **Token 4** is optional. It specifies whether **truncation** of the column due to the column width override occurs from the **left (L)** or **right (R)**. "L" is default.

SORT: (Optional)
Each line following defines a report column on which sort is to performed.

◊ **Token 1** on the line identifies the SMF **record-type** and **field name** in **"rectype.field"** notation as described for COLUMNS:.

In addition **"(A)"** or **"(D)"** may be supplied to define the **sort order** for that column as **ascending** or **descending**.

"(A)" is the default.

TOTALS: (Optional)
This section is optional since the default action is to produce totals for all fields that contain at least one valid number.

To prevent this default action and suppress all totals, supply NOTOTALS in the OPTIONS: section.

Each line following defines a report column for which totals/sub-totals are to be reported.

◊ **Token 1** on the line identifies the SMF **record-type** and **field name** in **"rectype.field"** notation as described for COLUMNS:.

Column data that is not a valid number does not contribute to the reported total value.

BREAK: (Optional)
Each line following defines a column on which a report break will occur.

◊ **Token 1** on the line identifies the SMF **record-type** and **field name** in **"rectype.field"** notation as described for COLUMNS:.

A change in the value of any of the break columns will result in a **report gap** that will include **sub-totals** for any numeric columns.

Any break column must also appear as a **"SORT:"** column.

◊ **Token 2** is optional and may be used to supply a number which restricts the length of data (starting at the left of the field) used to detect a change in value.

A useful example of this would be where the break column is a timestamp field in *yyyy/mm/dd hh:mm:ss* format, but the break is required on the date portion only (i.e. change of day). e.g.

```
BREAK:
  SMF042#24_DFSMS.zTME  10
```

BLANKIFZERO: (Optional)
Each line following defines a column which will have a value of "0" translated to blank for **readabilty** i.e. making non-zero values stand out more clearly.

◊ **Token 1** on the line identifies the SMF **record-type** and **field name** in **"rectype.field"** notation as described for COLUMNS:.

"BIZ:", **"BLANKWHENZERO:"** and **"BWZ:"** are all valid synonyms.

FILTER: (Optional)
Use the FILTER: section to define an optional record selection criteria expression, which may include multiple (possibly parenthesised) terms and supports the standard Data-Edit operators.

e.g. Select SMF Record Type 14 only when the DDNAME does not begin with "SYS" and DSN begins with either "CBL", "JGE" or "NBJ".

```
FILTER:
            SMF014_INPUT_or_RDBACK_Dataset.SMFTIOE5 \>> 'SYS'

    and (   SMF014_INPUT_or_RDBACK_Dataset.DSN      >> 'CBL'
          or SMF014_INPUT_or_RDBACK_Dataset.DSN     >> 'JGE'
          or SMF014_INPUT_or_RDBACK_Dataset.DSN     >> 'NBJ'
        )
```

Note that explicit qualification of the field names by prefixing the record/segment name (followed by a dot) is not necessary, provided the field is within the "Primary" (Base) Segment. So the above example could equally be specified as follows.

```
FILTER:
            SMFTIOE5 \>> 'SYS'

    and (   DSN      >> 'CBL'
          or DSN     >> 'JGE'
          or DSN     >> 'NBJ'
        )
```

However, where the field referenced is part of a "Secondary" segment, the segment name prefix is required. e.g.

```
FILTER:
    SMF030_Completion.zSCC > 0
```

REPEAT: (Optional)
Each line following identifies a secondary segment name.

The default action for the report tool is that it outputs a report line for each **base** record (primary segment) having potentially gathered information from one or more of its attached **secondary** segments.

In fact it is currently a limitation that at least one field must be selected from the base segment and selection may be made from one base segment only.

Because many SMF record types contain repeating groups of secondary segments, the "REPEAT:" option may be used to identify the segment record-type on each of which you wish to generate a report line.

REQUIRED: (Optional)
Should be unnecessary.

Each line following identifies a secondary segment name that must be included in the temporarily generated structure required to produce the report.

It is intended that the report tool be sophisticated enough to automatically recognise (from the "COLUMNS:" definition and other information availble in the supplied fully comprehensive SMF Record Type structures) the names of any secondary segments required to build the most basic structure capable of producing the desired report.

The fewer segments that are included, the faster the report will run.

This option is intended only for those occasions when the automatic process has failed to recognise a dependency and the SELCOPY support team should be notified in that event.

RESET: (Optional)
Each line following identifies a report column whose value should be reset to null following the output of a report line.

If a column is not identified in the RESET: section and is sourced from a secondary segment that does not appear again before the next report line is ready to be output, then the previous value will be repeated for that column.

So, as a general rule, if your report includes columns from segments that are not guaranteed to be present for every base segment, then a RESET: entry should be added to ensure a "blank" value is reported whenever the segment is missing.

OPTIONS: (Optional)

The OPTIONS: section currently supports only one keyword, that being **NOTOTALS** which prevents the tool's default action which is to report totals for any column containing at least one valid numeric value.

See also TOTALS:.

Window List (=W)

The Window List window may be opened via the following:

- Select option W. 'Window' from the Primary Option Menu panel menu bar.
- Select 'Window List' from the Primary Option Menu panel.
- Select 'All Windows' from the Window menu in the CBLe main window menu bar.
- Enter the command WINDOWLIST on the command line of any window.

The Window List window displays all open windows and their associated window names and allows the user to place focus on a specific window by selecting it from the list.

The hierarchy of parent/child windows is illustrated by indentation of the entries in the list.

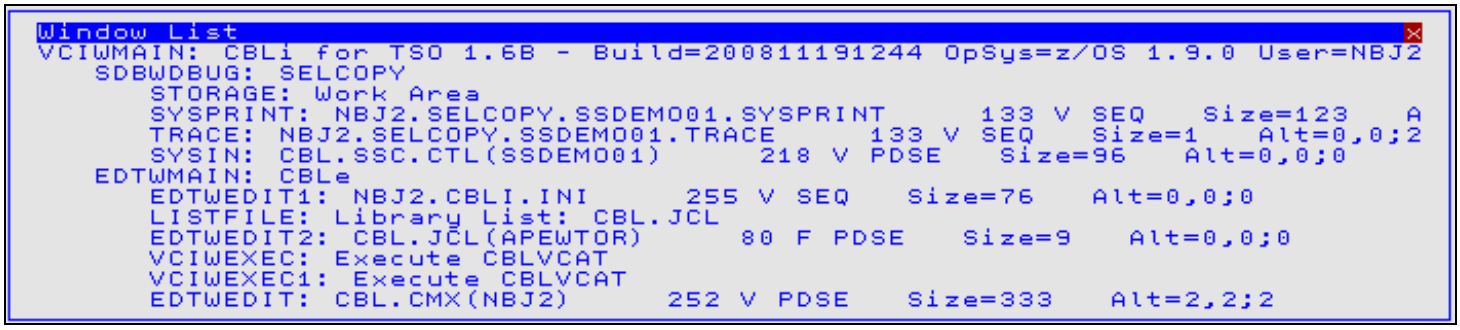


Figure 311. Window List window.

SELCOPYi Command Reference

SELCOPYi commands may be issued from a command line at the Command> prompt.
All SELCOPYi main window menu bar commands have a command line equivalent.

Command	Description
ABOUT	Display the About SELCOPYi window.
ALIAS	Open the Create ALIAS dialog window (includes support for PDSE Load libraries).
AMS	Open the IDCAMS Command window. An IDCAMS command may be passed as a parameter.
AMSDIALOG	Open the Execute IDCAMS dialog panel.
APE	Open the SELCOPYi Module List.
AUDPRINT	Display formatted DB2 Audit log output.
BOTTOM	Display the last lines of data.
BROWSE	Open the CBLe text editor to edit a file read-only.
CALENDAR	Open the calendar window.
CALC	Open the calculator window.
CBLI	Pass a command directly to the SELCOPYi command processor.
CBLICANCEL	Exit the SELCOPYi session without opening the quit session confirmation
CBLNAME	Open a storage window containing the loaded CBLNAME module.
CFOUT	Display SDE formatted Compare File report output.
CLOSE	Close a window.
CMDS	Previously executed commands.
COMMANDLINE	Set the command line attributes.
COMPFILE	Execute the Compare File Utility.
COMPLIB	Execute the Compare Library Utility.
COMPTABLE	Execute the Compare DB2 Table Utility.
CRETRIEV	Retrieve previous commands.
CSVGEN	Generate CSV (Comma Separated Variables) Utility.
CURSORSELECT	Perform the default operation based on the cursor position.
DB2	Open the SELCOPYi DB2 panels.
DCMD	Execute the DB2 Command execution panel.
DOWN	Scroll the window display downwards.
DRAGBORDERMINUS	Drag the window's border closer to the top left corner of the display.
DRAGBORDERPLUS	Drag the window's border away from the top left corner of the display.
DSINFORMATION	Display information about a data set.
DSQL	Execute the SQL Statement execution panel.
EDIT	Open the CBLe text editor to edit a file.
EO	Display output queue listing.
ERASE	Erase a file.
EXECSQL	Execute DB2 SQL statements.
FAV	Open the Favourites Panel.
FCOPY	Execute the File Copy Utility.
FS	Open the File Search window.
FSU	Execute the File Search/Update/Copy/Remap Utility.
FSUEND	Close an FSU report output display.
FSUOUT	Display SDE formatted FSU report output.
FSUUNDO	Execute the FSU Undo Updated records utility.
GETXML	Export an XML document from a DB2 Table column to a dataset.
HELP	Open the help top level window.
HELPINDEX	Open the help index window.
HOME	Place focus on the CBLe edit view containing the user's command centre file.
IEBCOPYDIALOG	Open the IEBCOPY Dialog window.
ISPF	Toggle between TSO and ISPF 3270 I/O.
ISPFUTIL	Start ISPF Utilities Panel.
JBOTTOM	Justify focus window at the bottom of the display area.
JLEFT	Justify focus window at the first column of the display area.
JRIGHT	Justify focus window at the last column of the display area.
JSONGEN	Generate JSON (JavaScript Object Notation) Utility.
JTOP	Justify focus window at the top of the display area.
KEYS	Set a function key or open the function key dialog.

KEYLIST	Control function key lists.
LA	Open the list allocated files window.
LAS	List VSAM Associated objects.
LC	Open the cataloged files list window. A fileid mask (or DLBL) may be passed as a parameter.
LD	Open the dataset details list window. For MVS a partial dataset name may be passed as a parameter. For CMS a fn ft fm pattern may be passed as a parameter.
LEFT	Scroll the window display to the left.
LJQ	Open the Job Enqueues list window. A job name may be passed as a parameter.
LL	Open the library members list window. For MVS a PDS name may be passed as a parameter. For VSE a LIBR library, sublibrary and member name and type pattern may be passed as a parameter.
LLS	Open the loaded structures list window.
LLX	Execute the Locate Library Members utility
LP	Open the HFS Paths list window. An absolute or relative HFS path may be passed as a parameter.
LQ	Open the Enqueues list window. An enqueue major name and resource name may be passed as a parameter.
LSG	Open the SMS Storage Group list window.
LSGV	Open the SMS Storage Group Volumes list window.
LV	Open the VTOC files list window. A volume name may be passed as a parameter.
LVOL	Open the DASD volumes list window. A volume name pattern may be passed as a parameter.
LVR	Open the CBLVCAT Raw list window. A CBLVCAT command may be passed as a parameter.
LX	Open the VTOC extents list window. A volume name may be passed as a parameter.
MAXIMISE	Maximise a window.
MDINEXT	Place focus on the next MDI child window.
MDIPREV	Place focus on the previous MDI child window.
MERGE	Open the Merge Datasets Panel
MINIMISE	Minimise a window.
MOVEWINDOW	Move a window.
NEXTMAINWINDOW	Sets the focus window to the next main window within the SELCOPYi desktop.
NEXTWINDOW	Sets the focus window to the next window in the ring of all windows.
PFSHOW	Controls display and contents of the function key display area.
PFSHOWSTYLE	Sets the appearance of PFkeys F13-F24 in the function display area.
POWER	Open the POWER Command Output window.
PREVMAINWINDOW	Sets the focus window to the previous main window within the SELCOPYi desktop.
PREVWINDOW	Sets the focus window to the previous window in the ring of all windows.
PRINT	Starts the print utility.
PUTXML	Import an XML document to a DB2 Table column.
QUICKREF	Display the Quick Reference document.
QUIT	Exit and close the current window.
RENAME	Rename a file.
RESTORE	Restore a window.
RETRIEVE	Retrieve previously executed commands.
RIGHT	Scroll the window display to the right.
SDATA	Direct a command to the Structured Data Environment (SDE).
SDE	Same as SDATA but opens SDE dialog window if no parameters.
SDSF	Start the SDSF application.
SELCOPY	Start the SELCOPY Debug window.
SETCOLOUR	Remap the appearence of a colour and its associated highlighting.
SETFOCUS	Sets the focus window to a named window in the ring of all windows.
SHOWPOPUPMENU	For storage windows, display the options pop-up menu.
SHOWWATTR	Show Window Attributes.
SIZEWINDOW	Resize a window.
SQL	Open the Dynamic SQL window.
STRUCTURE	Manage SDE structure to dataset associations.
SVC	Display the status of the CBLVCAT Interactive SVC.
SYSAPF	Open the APF List window. (MVS only)
SYSCOMMAND	Pass a command directly to the local TSO or CMS command processor.
SYSI	Open the system information window.
SYSLL	Open the Link List window. (MVS only)
SYSLPA	Open the LPA Modules window. (MVS only)
SYSMENU	Open a window's System Menu.

SYSPGM	Open the Loaded Programs Menu.
SYSSTOR	Open the Storage Statistics window.
SYSTASK	Open the Task List window. (MVS only)
TASK	Execute a program as a sub-task of SELCOPYi.
TOP	Display the first lines of data.
TRACE	Start/stop the SELCOPYi diagnostic trace utility.
UP	Scroll the window display upwards.
VCAT	Open the Execute CBLVCAT window. A CBLVCAT command may be passed as a parameter.
VIEW	Open the CBLe text editor to edit a file read-only.
VOLSTATS	Open the Volume statistics window.
WINDOWLIST	Open the window list window. This lists all currently open windows.
WINDOWNAMES	Hide or display the window names in the title bar.
XMLGEN	Generate XML utility.

ABOUT

Syntax:

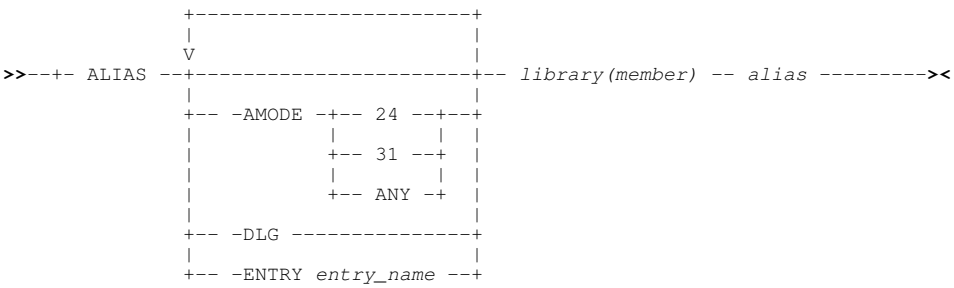
```
>>-- ABOUT ------------------------------------------------------------------><
```

Description:

ABOUT opens the About SELCOPY/i modal information window.

ALIAS

Syntax:



Description:

Use the ALIAS command to create a new PDS/PDSE library member alias, or open "Create ALIAS" dialog window.

Note that aliases for load-library members are created using the binder to relink the module in being aliased. This will result in an update to the module's **TTR**.

Use of the **-DLG** parameter invokes the Create ALIAS dialog window, which is also available through the **'A'** prefix-command in a Library List window.

Parameters:

library(member)
: The PDS/PDSE library member be be aliased.

alias
: The alias name to be added

-AMODE
: For a load-library member, used to specify the Addressing Mode for the new aliased entry-point. Valid arguments are 24, 31 and ANY.

-DLG
Invoke the "Create Alias" dialog window.

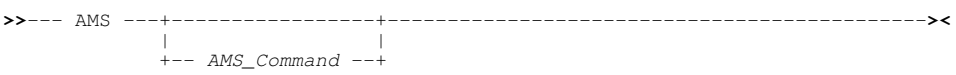
-ENTRY *entry_name*
For a load-library member, used to specify the symbolic name of the entry-point address to be used.

Examples:

```
< alias JGE.CBLINST.D070919.EXE.COPY(I141) CBLI
< alias JGE.CBLINST.D070919.EXE.COPY(I141) CBLAVTAM -entry VCIAVTAM
< alias JGE.CBLINST.D070919.EXE.COPY(I141) CBLIVTAM -entry VCIRVTAM
< alias JGE.CBLINST.D070919.EXE.COPY(V212) CBLV
< alias JGE.CBLINST.D070919.EXE.COPY(V212) CBLV31 -amode 31
< alias JGE.CBLINST.D070919.EXE.COPY(V212) CBLV24 -amode 24
< alias JGE.CBLINST.D070919.EXE.COPY(V212) CBLV24 -amode 24 -dlg
< alias JGE.CBLINST.D070919.EXE.COPY(S202) SELCOPY
```

AMS

Syntax:



Description:

Use the AMS command to open the IDCAMS Command window and optionally execute an AMS/IDCAMS command.

The IDCAMS Command window displays the SYSPRINT/SYSLST output generated on execution of the IDCAMS command.

Parameters:

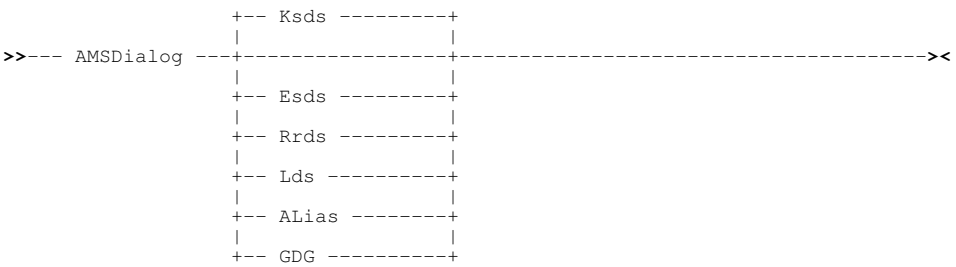
AMS_Command
AMS/IDCAMS command to be executed.

Examples:

```
AMS  LISTCAT  ENTRY(CBL.MCAT.CMP.P)  ALL
```

AMSDIALOG

Syntax:



Description:

Use the AMSDIALOG command to open the appropriate Define VSAM, Define GDG or Define Catalog ALIAS dialog window.

Parameters:

KSDS | ESDS | RRDS | LDS | ALIAS | GDG
Specifies the catalog entry type for which a Define dialog window will be opened.
Default is Define VSAM KSDS.

Examples:

`AMSD ESDS`
: Open the Define VSAM ESDS dialog window.

`AMSD ALIAS`
: Open the Define Catalog ALIAS dialog window.

APE

Syntax:

```
>>---- APE ------------------------------------------------------------------><
```

Description:

Use the APE (Assembler Programming Environment) command to display the Module List window containing the current status of all modules that comprise SELCOPY/i.

The Module List window may also be opened via the System menu of the SELCOPY/i main window menu bar.

APE may be executed to determine the level of a currently installed module.

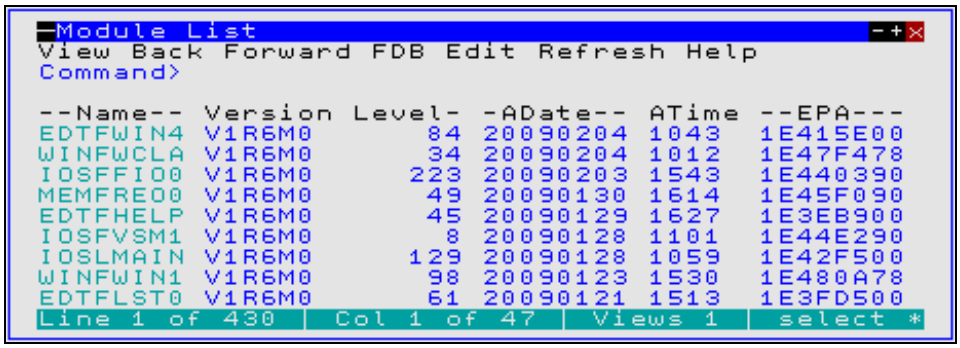


Figure 312. Module List window.

AUDPRINT

Syntax:

```
>>---- AUDPrint ----+--------------------+-------------------------------------><
                    |                    |
                    +-- db2_audit_dsn --+
```

Description:

Use the AUDPRINT command to format a SELCOPY/i DB2 audit log file into a readable report and display it in a CBLe text edit view.

If specified with no *db2_audit_dsn* parameter, the Print Audit Report (ZZS2AUDP) panel is opened.

Parameters:

db2_audit_dsn
: The DSN of a SELCOPY/i DB2 audit log file.

  The high level qualifier of this DSN is determined via the Audit Log Dataset Options (ZZS2AUDS) panel.

BOTTOM

Syntax:

```
>>--+- BOTTOM----+-------------------------------------------------------------><
    |             |
    +- BOT ------+
```

Description:

Use the BOTTOM command to display the bottom lines of the data in the focus window. The last line of the data becomes the last line of the display area.

Note that the SELCOPY/i BOTTOM command acts differently to the BOTTOM command available in a CBLe text edit or SDE edit document window.

BROWSE

Syntax:

```
>>-- Browse -- fileid ---| SDE BROWSE Opts |---------------------------------><
```

Description:

For **VSE** and **CMS** systems, BROWSE is a synonym for VIEW.

Use the BROWSE command to open a Structured Data Environment (SDE) BROWSE window view to browse a page of data from the specified fileid.

If the CBLe text editor main window has been stopped, BROWSE will start the CBLe main window and open an edit view for the user's HOME CMX file before opening the SDE browse window for the requested file.

Use BROWSE instead of VIEW to browse large data sets. Unlike EDIT and VIEW, BROWSE does not need to load the entire file into storage in order to display a page of records.

Parameters:

fileid
: The fileid of the file to be browsed.

  fileid may be any of the following:

  - ◊ The DSN of a physical sequential data set.
  - ◊ The DSN of a VSAM (KSDS, ESDS, RRDS, VRDS or LDS) data set.
  - ◊ The library DSN and parenthesised member name of a PDS or PDSE library member.
  - ◊ The library DSN, parenthesised member name and absolute or relative number of a PDSE library member generation as described under z/OS PDSE Library Member Generations. (PDSE version 2 with MAXGENS only.)
  - ◊ The name of a member to be edited from the same PDS or PDSE library as the member displayed in the current data edit window view. If no data edit window view is open and *fileid* is an IDCAMS alias name, then data belonging to the aliased dataset is browsed. Otherwise, it is treated as an HFS/ZFS file name in the user's current working directory.
  - ◊ The name of an HFS/ZFS file system fileid.

SDE BROWSE Opts
: See SDE BROWSE for supported parameters.

CALENDAR

Syntax:

```
>>--+- CALENDAR -+-------------------------------------------------------------><
    |             |
    +- CAL ------+
```

Description:

Use the CAL command to open the Calendar Window.

The Calendar Window may also be opened via the Utilities entry of the SELCOPY/i main menu.

CALC

Syntax:

```
>>--+- CALCULATE -+--+---------------------+---------------------------------><
    |              |  |                     |
    +- CALC ------+  +-- REXX_expression --+
```

Description:

Use the CALC command to open the Calculator Window and optionally evaluate a REXX expression.

The Calculator Window may also be opened via the Utilities menu of the SELCOPY/i main window menu bar.

Parameters:

REXX_expression
: The Calculator will evaluate any valid REXX expression. This may include the result of any REXX function built in to REXX or written by the user.

Examples:

```
CALC  (1024+281) / (3*2)

CALC  c2x(bitxor('af'x,'44'x))
```

CBLI

Syntax:

```
>>-- CBLI --- command -------------------------------------------------------><
```

Description:

Execute a SELCOPY/i command.

The specified command is passed to the SELCOPY/i environment. Any windows opened are child windows of the SELCOPY/i main window, not of the current window.

Parameters:

command
: SELCOPY/i command.

Examples:

```
cbli   vcat < cbl.vvc.ctl(vvrep01)
```
: Open a CBLVCAT Interactive window and execute CBLVCAT with control statements from file CBL.VVC.CTL(VVREP01).

CBLICANCEL

Syntax:

```
>>-- CBLICANcel ------------------------------------------------------------><
```

Description:

Exit and close the SELCOPY/i session without opening the quit session confirmation pop-up window.

CBLNAME

Syntax:

```
>>-- CBLNAME ---------------------------------------------------------------><
```

Description:

Use the CBLNAME command to open the CBLNAME window.

The CBLNAME storage display window may also be opened via the System menu item of the SELCOPY/i main window menu bar.

CFOUT

Syntax:

```
>>---- CFOUT --------+--------------------------+-------------------------------><
                     |                          |
                     +-- compfile_report_dsn --+
```

Description:

Use the CFOUT command to display a saved SELCOPY/i Compare Files utility output report in in an SDE Browse window view.

CFOUT will automatically attempt to format the report records using an SDE structure (SDO) with DSN *compfile_report_dsn***.SDO**. If this structure does not exist, CFOUT will fail with error ZZSD024E structure not found.

Parameters:

compfile_report_dsn
: The DSN of a SELCOPY/i Compare Files utility output report.

  The default report DSN is '*userid*.SELCOPYI.COMPFILE.REPORT' where *userid* is the user's RACF logon id.

CLOSE

Syntax:

```
>>--+- CLOSE -+---+--------------+-----------------------------------------><
    |         |   |              |
    +- CLO ---+   +- windowname -+
    |         |
    +- Quit --+
```

Description:

Use the CLOSE command to close a window.

By default this command is assigned to function key **PF3**. You can also use the close button if the window has one.

Parameters:

windowname
The window name of the window to close. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

CMDS

Syntax:

```
>>--- CMDS ----------------------------------------------------------------><
```

Description:

Opens the Retrievable Commands Selection List panel (ZZSGCMDS) to view, retrieve and, optionally, execute previously executed commands.

This panel is an interactive panel windows (window class WINWIPO0) displaying an embedded table of commands that have been executed since the start of the SELCOPY/i session.

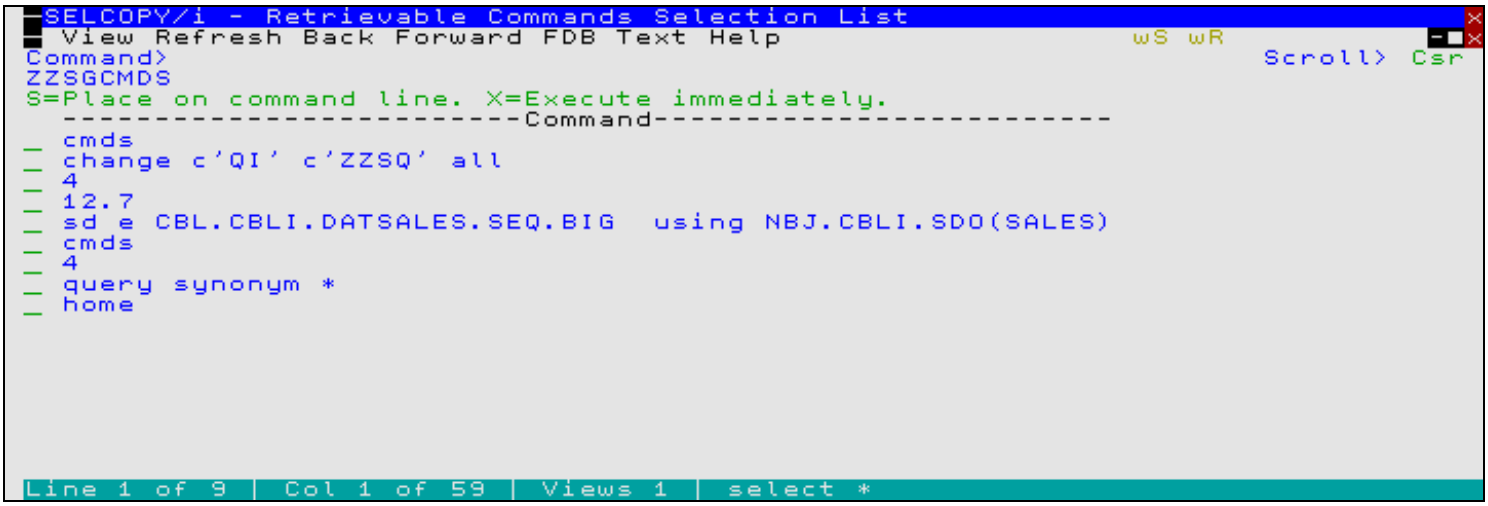


Figure 313. Retrievable Commands Selection List Panel.

COMMANDLINE

Syntax:

```
>>--+- COMMANDLINE -+--+--------------+--+---------+--+--------+---------------->
    |               |  |              |  |         |  |        |
    +- CLN ---------+  +- windowname -+  +- SHOW -+  +- TOP -+
                                        |         |  |       |
                                        +- HIDE -+  +- BOT -+

 >--+------------------+--+-------------+-------------------------------------><
    |                  |  |             |
    +- PROMPT=prompt -+  +- SCROLL ----+
                        |             |
                        +- NOSCROLL --+
```

Description:

If this command is issued with no parameters or with the window name parameter only, then a dialog box is opened in which the user can define the characteristics of the command line of the specified window.

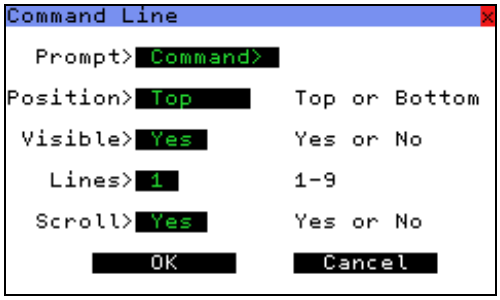


Figure 314. Command Line Dialog Window.

If parameters other than the window name are supplied then the specified change is made to the command line.

Parameters:

windowname
: The window name containing the command line. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

SHOW
HIDE
: Show or hide the command line.

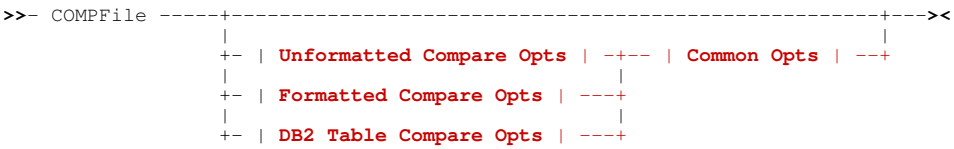
TOP
BOT
: Show the command line at the top or the bottom of the window.

PROMPT
: The character string following the PROMPT= keyword is used as the prompt prefixing the command line input area in the window.

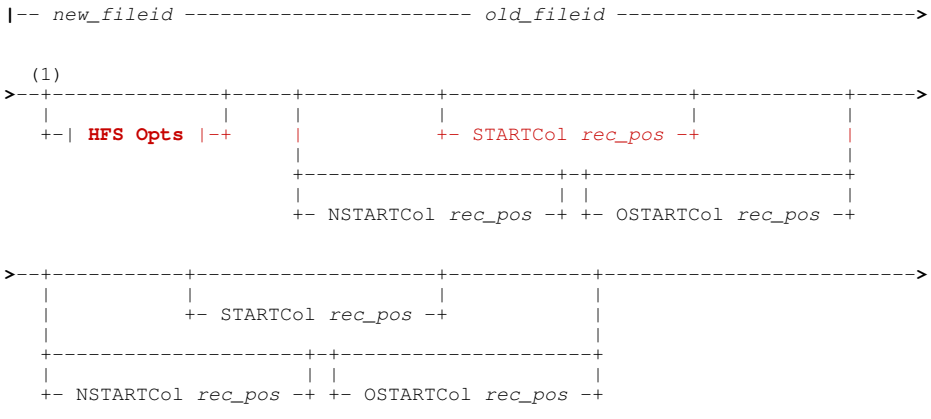
SCROLL
NOSCROLL
: For CBLe only, show or hide the ISPF Edit style scroll entry field.
For INTERFACE=ISPF, the default is SCROLL.
For INTERFACE=CBLE, the default is NOSCROLL.

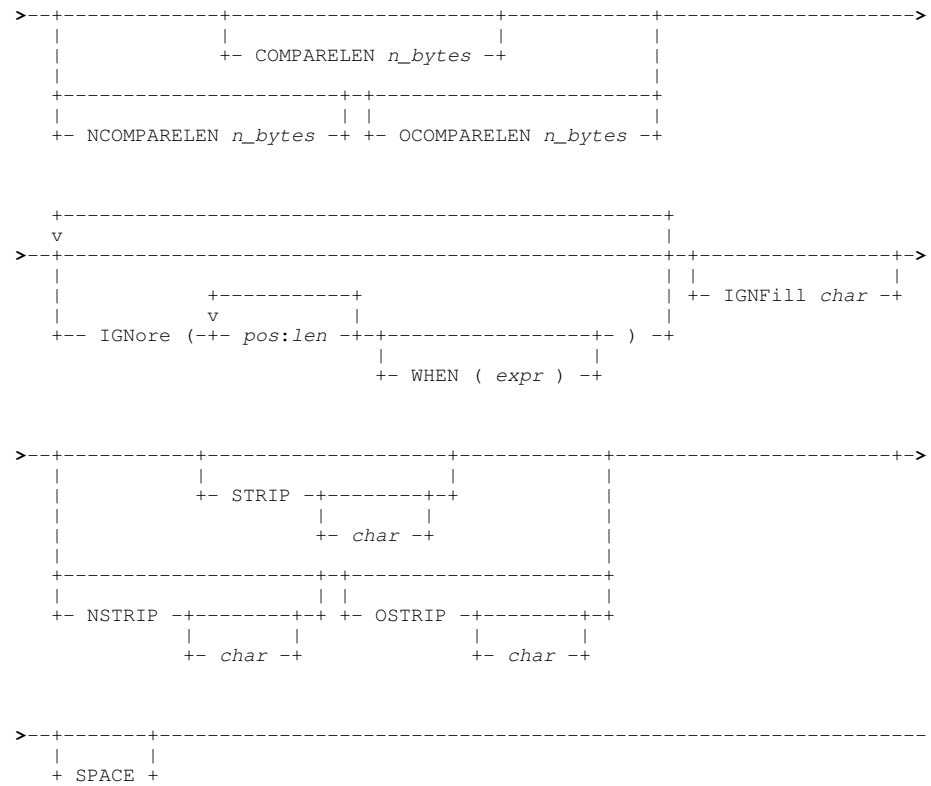
COMPFILE

Syntax:

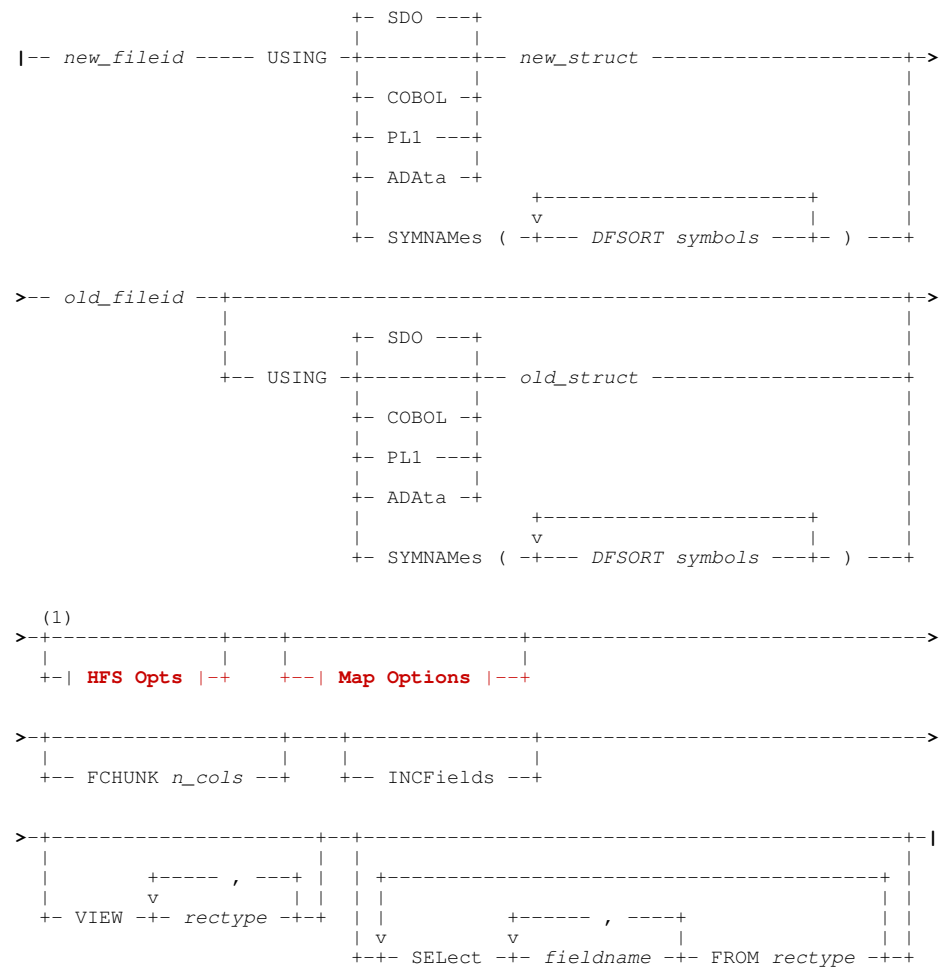


Unformatted Compare Opts:

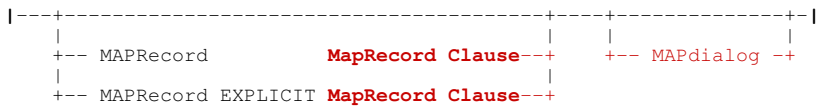




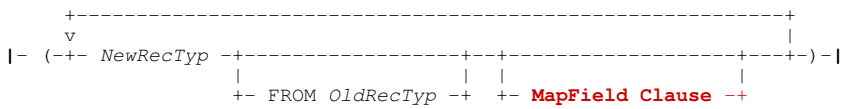
Formatted Compare Opts:



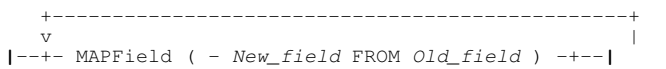
Map Options:



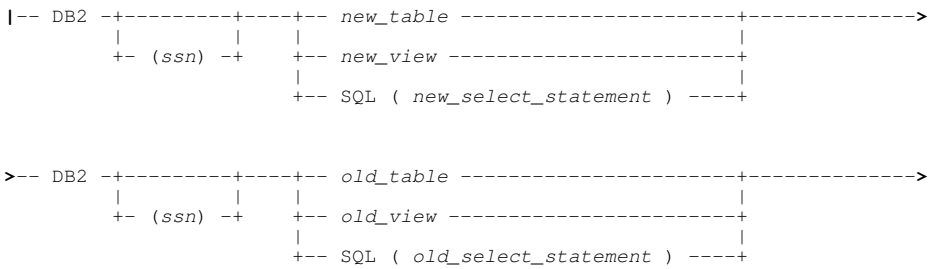
MAPRecord Clause:



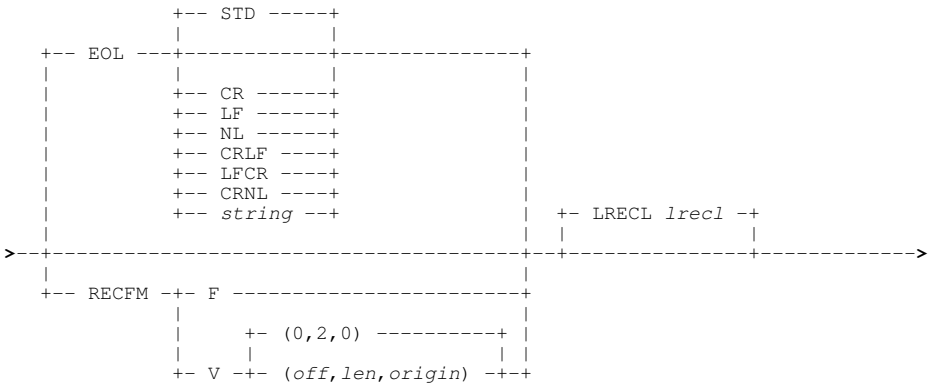
MAPField Clause:



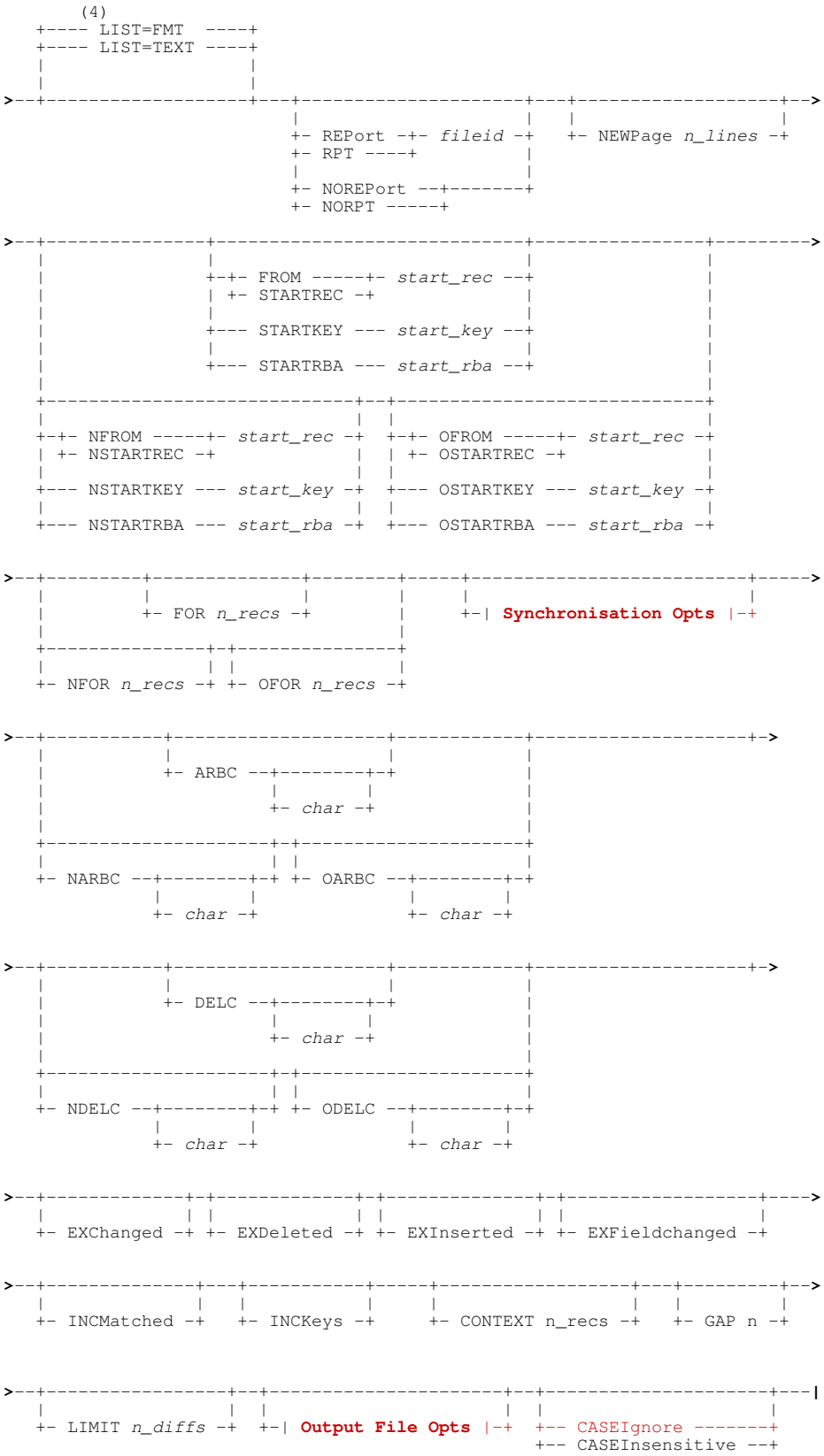
DB2 Table Compare Opts:



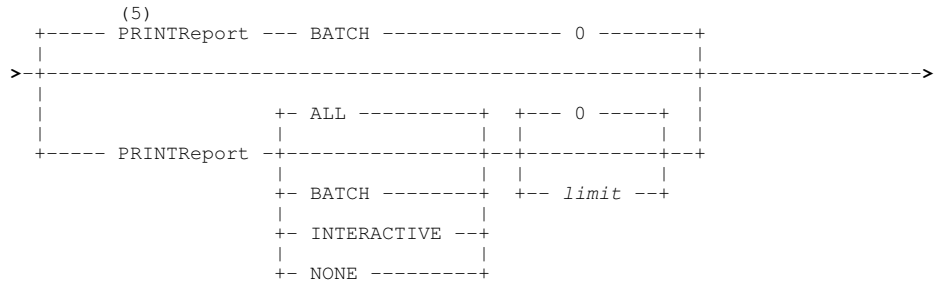
HFS Opts:



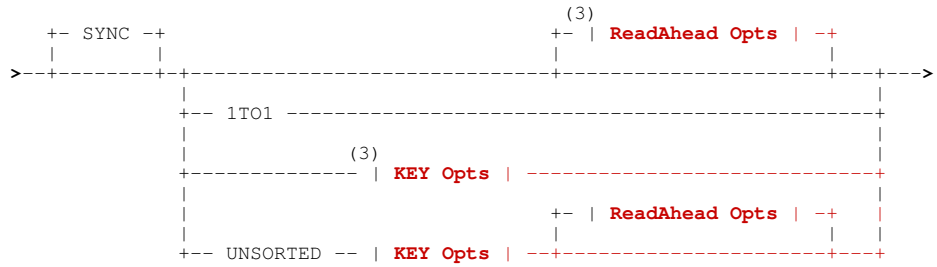
Common Opts:


(4)
LIST=FMT
LIST=TEXT
REPort -+- fileid -+
RPT ----+
NOREPort --+-------+
NORPT -----+
NEWPage n_lines -+
FROM -----+- start_rec --+
STARTREC -+
STARTKEY --- start_key --+
STARTRBA --- start_rba --+
NFROM -----+- start_rec -+
NSTARTREC -+
NSTARTKEY --- start_key -+
NSTARTRBA --- start_rba -+
OFROM -----+- start_rec -+
OSTARTREC -+
OSTARTKEY --- start_key -+
OSTARTRBA --- start_rba -+
FOR n_recs -+
NFOR n_recs -+
OFOR n_recs -+
Synchronisation Opts
ARBC --+--------+-+
char -+
NARBC --+--------+-+
OARBC --+--------+-+
DELC --+--------+-+
NDELC --+--------+-+
EXChanged -+
EXDeleted -+
EXInserted -+
EXFieldchanged -+
INCMatched -+
INCKeys -+
CONTEXT n_recs -+
GAP n -+
LIMIT n_diffs -+
Output File Opts
CASEIgnore -------+
CASEInsensitive --+

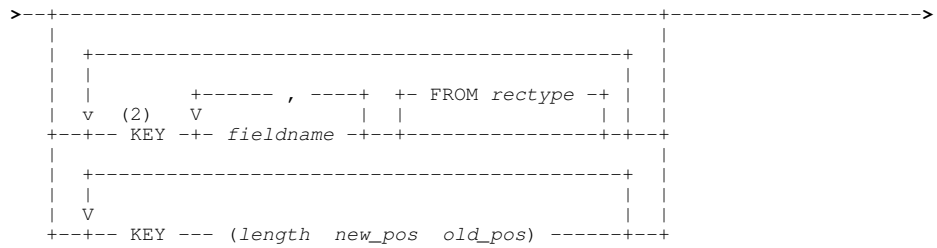
Obsolete Opts:



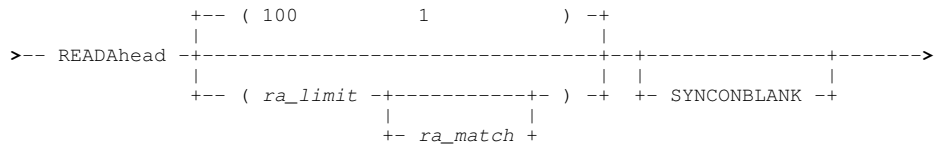
Synchronisation Opts:



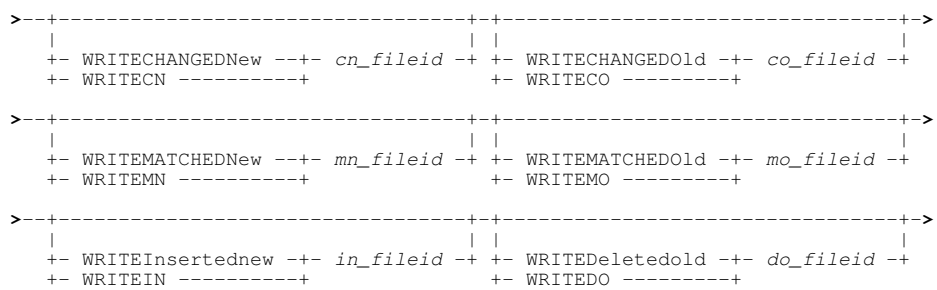
KEY Opts:



ReadAhead Opts:



Output File Opts:



Notes:

1. All parameter options that follow may be entered in any order.
2. Formatted Compare only.
3. If NEW and OLD are both KSDS with a defined key of equal length, then default synchronisation is **KEY(*length new_pos old_pos*)** with the key locations being automatically picked up from the VSAM file definitions.
4. "LIST=FMT" is default if running online from SELCOPYi, "LIST=TEXT" is default if running in batch.
5. "PRINTREPORT" options are now obsolete, superseded by "LIST=FMT|TEXT"..

Description:

Use the COMPFILE command to perform an Unformatted or Formatted compare of two files and optionally report record matches, inserts, deletions and/or changes. Execute COMPFILE with no parameters to open the Compare Files panel.

Files selected for compare may be any combination of cataloged or uncataloged sequential datasets or PDS/PDSE library members, VSAM datasets or HFS files.

File compare has been categorised as Basic (Unformatted), Extended Unformatted and Extended Formatted, each of which is discussed in detail under *File Compare* in chapter *SELCOPY/i Utilities*.

So long as the remaining command syntax is valid, any area or areas of the COMPFILE command string may be commented, and so ignored by the SELCOPY/i command parser, using REXX style comment delimiters. i.e. enclose areas of the command stream text between "/*" and "*/". This is particularly useful when COMPFILE is executed (with <F16>) from the user's HOME (CMX) data set command centre where, using the continuation character, commands may span several lines. e.g. To temporarily omit options EXChanged and INCMatched in the following COMPFILE command...

```
  <COMPF                                    \
    CBLMCT:SELCOPY.CBLI.CBLE(BOXTOT)        \
    CBL.CBLI310.CBLE(BOXTOT)                \
  /*                                        \
     EXCHANGED                              \
  */                                        \
     STARTREC    2                          \
     STARTCOL   11                          \
     COMPARELEN 70                          \
  /*                                        \
     INCMATCHED                             \
  */                                        \
     SYNC READAHEAD (100 1)
```

COMPFILE generates an output report in an SDE window view. See Compare Files Output.

Parameters:

new_fileid
: Specifies the PDS/PDSE member, HFS file path, Sequential or VSAM data set containing the NEW image of the file data to be compared.

  If *new_fileid* references an uncataloged sequential data set or PDS library, then a volume serial number prefix must be included on *new_fileid* in the format "*volser:data.set.name*".

  new_fileid may optionally be enclosed in single quotes/apostrophes (') which get stripped by COMPFILE. Note that, regardless of whether enclosing quotes are specified the TSO prefix is never added to the *new_fileid* DSN.

USING SDO|COBOL|PL1|ADATA *new_struct*
: Specifies the structure format (SDE structure, COBOL or PL1 Copybook, COBOL or PL1 ADATA output) and structure name (PDS/PDSE library member or Sequential data set) to be used to map record data in *new_fileid*.
  If no structure format is specified, the default is SDO (SDE structure).

  This option forces a formatted compare of the record data.

USING SYMNAMES (*DFSORT symbols*)
: Specifies DFSORT SYMNAME symbol definitions that are to be used to format the NEW file data records. The order in which symbol definitions are supplied dictate the order in which the fields will occur in the record type definition.

  The symbol name definitions within the SYMNAMES parentheses may be supplied directly in-line and/or via input data sets/library members.

```
SYMNAMES( Card,06,04,CH  Dept,46,03,CH  Amount,49,06,PD  )
SYMNAMES( SYS1.MACLIB(EDGSMFSY)  SYS1.MACLIB(EDGSRCSY) )
SYMNAMES( CBL.DFSORT.SYM(CBLATRAC)  TCB,*,4,BI )
```

old_fileid
: Specifies the PDS/PDSE member, HFS file path, Sequential or VSAM data set containing the OLD image of the file data to be compared.

  Rules for specification of *old_fileid* are the same as for *new_fileid*.

USING SDO|COBOL|PL1|ADATA *old_struct*
: Specifies the structure format and structure name used to map record data in *old_fileid* for a formatted compare. See USING *new_struct*.

Records that are assigned a record type defined in *old_struct* will be compared with records assigned the same record type in *new_struct*. Any records in the NEW and OLD files that are assigned a record type that does not exist in both *old_struct* and *new_struct* will be flagged as either inserted or deleted as appropriate.

If a field column in a record type defined in *new_struct* does not exist in the record type of the same name defined in *old_struct* or vice versa, then that field is excluded from the compare. Thus, only field columns of the same name in record types of the same name are compared.

If not specified, *new_struct* will be used to map record data in *old_fileid*.

USING SYMNAMES (*DFSORT symbols*)

Specifies DFSORT SYMNAME symbol definitions that are to be used to format the OLD file data records. The order in which symbol definitions are supplied dictate the order in which the fields will occur in the record type definition.

The symbol name definitions within the SYMNAMES parentheses may be supplied directly in-line and/or via input data sets/library members.

```
SYMNAMES( Card,06,04,CH  Dept,46,03,CH  Amount,49,06,PD  )
SYMNAMES( SYS1.MACLIB(EDGSMFSY)  SYS1.MACLIB(EDGSRCSY) )
SYMNAMES( CBL.DFSORT.SYM(CBLATRAC)  TCB,*,4,BI )
```

MAPDIALOG

MAPDIALOG (MAP) causes a dialog to be displayed allowing the user to interactively "match-up" input to output record-types, and fields within those record-types.

Record-type and Field names that are identical in both the input and output structure are automatically matched. See the **EXPLICIT** option for the **MAPRECORD** parameter which may be used to prevent this action.

MAPRECORD/MAPFIELD

Use the MAPRECORD (MAPR) clause to match up one or more NEW record-types to their corresponding OLD record-type when comparing files mapped by different layouts.

It's only necessary to explicitly define the match if the NEW record-type name is not identical to the OLD record-type name, or if there are fields within an identically named record-type that are not identically named in the NEW and OLD structures.

Each record-type match up clause may be immediately followed by a bracketed list of MAPFIELD clauses to match up field names within that particular record-type.

Again, it's only necessary to explicitly define the match if the NEW field name is not identical to the OLD field name.

To prevent automatic match up of identical record-type and field names, just add the **EXPLICIT** keyword immediately following **MAPRECORD**.

e.g.

```
MapRecord explicit
            /* New Rectype/Field          Old Rectype/Field */
 (             ZARTIST                from ARTIST  /* Record-type */
   ( MapField( ZRT                    from RT                        )
     MapField( ZARTIST                from ARTIST.ARTIST             )
   )
               ZALBUM                 from ALBUM   /* Record-type */
   ( MapField( ZRT                    from RT                        )
     MapField( ZALBUM                 from ALBUM.ALBUM               )
   )
               ZTRACK                 from TRACK   /* Record-type */
   ( MapField( ZRT                    from RT                        )
     MapField( ZRELEASE-DATE.YYYY     from RELEASE_DATE.RELEASE_YYYY )
     MapField( ZRELEASE-DATE.MM       from RELEASE_DATE.RELEASE_MM   )
     MapField( ZRELEASE-DATE.DD       from RELEASE_DATE.RELEASE_DD   )
     MapField( ZPERSISTENT-ID         from PERSISTENT_ID             )
     MapField( ZNAME                  from NAME                      )
     MapField( ZFILE-SIZE             from FILE_SIZE                 )
   )
 )
```

new_table/view/select_statement

Specifies the DB2 Base/Results Table containing the NEW image of the data to be compared.

new_table/view may be specified with 1, 2 or 3 qualifiers representing *name*, *schema.name* or *location.schema.name* respectively. Default location is the local DB2 server and the default schema is the value assigned to special register CURRENT SCHEMA (initially set to the user's SQLID).

SQL (*select_statement*) specifies that the results table of this select statement is to be used. Following the keyword *SQL* in parentheses is an SQL query.

old_table/view/select_statement

Specifies the DB2 Base/Results Table containing the OLD image of the data to be compared.

old_table/view may be specified with 1, 2 or 3 qualifiers representing *name*, *schema.name* or *location.schema.name* respectively. Default location is the local DB2 server and the default schema is the value assigned to special register CURRENT SCHEMA (initially set to the user's SQLID).

SQL (*select_statement*) specifies that the results table of this select statement is to be used. Following the keyword *SQL* in parentheses is an SQL query.

HFS Opts

Applicable to all (NEW and/or OLD) HFS files, the following options may be specified to determine how HFS data is processed by the utility.

For non-HFS files, HFS options are ignored.

HFS options may not currently be specified independently for the NEW and OLD files. Please contact CBL technical support to request this as a new feature if this is something you require.

`EOL=STD|NL|CR|LF|CRLF|LFCR|CRNL|`*`string`*

Specify the EOLIN (input end-of-line) delimiter used to identify the end of each record for unformatted (non-RECFM F or V) HFS file input. EOLIN delimiters are not included in the edited record data or record length. EOL parameter elements are as follow:

STD	-	Any standard line-end.
NL	X'15'	New Line.
CR	X'0D'	Carriage Return.
LF	X'0A'	Line Feed.
string	-	A 2-byte user specified character or hex string.

STD is default so that the EDIT operation scans the input data for any of the standard EOL combinations (not *string*), stopping when one is found. This EOL combination is used as EOLIN for the file.

`RECFM F | V (`*`off`*`,`*`len`*`,`*`origin`*`)`

Specifies that the data is to be treated as containing Fixed or Variable length format records.

RECFM F indicates that all records are of a fixed length as defined by the LRECL argument.

RECFM V allows the user to specify the location of the record length fields within the data as follows:

off	Offset of the record length field from the start of the record.
len	Length of the record length field.
origin	The start of the record data at which the record length is applied.

Default is (0,2,0) which describes standard RECFM V organisation data sets.
The length field will be included as part of the input record data, so, if a CHANGE operation is specified, care must be taken not to corrupt the length field.

`LRECL `*`lrecl`*

Specifies the maximum record length of input HFS file records.

For RECFM F HFS files, *lrecl* is the fixed length of the records processed. If the HFS file size is not a multiple of *lrecl* value, then the last record of the file will be short.
Default *lrecl* for this types of file is 80.

For RECFM V and unformatted (EOL delimitted records) HFS files, if a record length exceeds *lrecl*, processing is stopped for that particular file.
Default *lrecl* for these types of files is 32752.

`STARTCOL | NSTARTCOL | OSTARTCOL `*`rec_pos`*

For **unformatted compare** only, STARTCOL specifies the 1st position within the records of **both** the old and NEW files at which the data compare will begin. Record data at positions before *rec_pos* is not included in the compare.

If *rec_pos* is a position beyond the length of the record, then the compare length is always set to zero, regardless of any COMPARELEN value.

Alternatively, the record data start positions may be specified differently for the NEW and OLD files using NSTARTCOL and OSTARTCOL respectively. e.g. To compare records in NEW file starting at position 101 against records in OLD file starting at position 51.

If STARTCOL/NSTARTCOL/OSTARTCOL is not specified, *rec_pos* defaults to 1.

`COMPARELEN | NCOMPARELEN | OCOMPARELEN `*`n_bytes`*

For **unformatted compare** only, COMPARELEN specifies the maximum number of bytes *n_bytes* of data to be compared within the records of **both** the old and NEW files. The start of the data to be compared is *rec_pos* as specified by STARTCOL/NSTARTCOL/OSTARTCOL. Record data at position *rec_pos + n_bytes* and beyond is not included in the compare.

If *n_bytes* is greater than the amount of data remaining in the record, then the length of data compared is equal to 1 plus the record length minus *rec_pos*.

Alternatively, the record data maximum compare lengths may be specified differently for the NEW and OLD files using NCOMPARELEN and OCOMPARELEN respectively. Because COMPFILE will always flag a compare mismatch if the NEW and OLD compare lengths are different, specification of NCOMPARELEN and/or OCOMPARELEN is valid only when used with the STRIP/NSTRIP/OSTRIP option. e.g. To compare a blank padded character field of length 10 in OLD file against a blank padded character field of length 20 in NEW file, use `NCOMPARELEN 20 OCOMPARELEN 10 STRIP`.

If COMPARELEN/NCOMPARELEN/OCOMPARELEN is not specified, *n_bytes* defaults to 1 plus the length of the record minus *rec_pos*.

`IGNORE pos:len WHEN ( expr )`

For **unformatted compare** only, each IGNORE clause may specify one or more area within the record that will effectively be ignored for the purpose of the comparison.

An optional condition expression (*expr*) may be provided in order to restrict the records on which the IGNORE funtion is triggered. e.g.

```
ign( 11:8
     21:6
    131:999
    when(    left(record,1) =  '3'
          & (     record << 'blues'
             or substr(record,209,4) = '1997'
            )
        )
   )

ign( 31:4
     22:12
     99:72
    when(   (left(record,1) =  '1'  & lrecl =  72)
         or (left(record,1) =  '2'  & lrecl =  71)
        )
   )
```

For efficiency reasons, the technique used by the comparison process is that for every record that satifies the optional **WHEN** condition, each specified location (*pos*:*len*) is overlayed with "-" minus signs (or the specified IGNFILL character) prior to the record comparison taking place.

Then, provided both the NEW and OLD records satisfy the WHEN condition, any differences that existed within the original content of these fields will be masked.

Again for reasons of efficiency, if read-ahead synchronisation is in use, then any records displayed in the output report will show the minus signs in the masked regions.

For other syncronisation types the original data will be displayed in the masked regions. However, should a difference be detected outside of these regions, the report record provided to hilight the differences (using "#" underlining) will also include minus signs to indicate the masked regions.

`IGNFILL char`

For **unformatted compare** and applicable only if an IGNORE clause is used, IGNFILL specifies the character (*char*) that COMPFILE uses to overlay masked areas within the record.

The default is the "-" (hyphon/minus) symbol.

`STRIP | NSTRIP | OSTRIP char`

For **unformatted compare** only, STRIP indicates that trailing characters in the compare data of **both** the NEW and OLD files are to be stripped (so reducing the compare length) and optionally specifies the strip character *char*. For each occurrence of *char* occupying the last position of the compare data the compare length is reduced by 1.

Alternatively, a different strip *char* may be specified for compare data in the NEW and OLD files using NSTRIP and OSTRIP respectively.

If STRIP/NSTRIP/OSTRIP is not specified, then no character are stripped from the compare data. If any of these parameters are specified without *char*, the default is blank (X'40').

`ARBC | NARBC | OARBC char`

ARBC activates and specifies an arbitrary character, which is treated as a wildcard for comparison purposes and is typically used to mask acceptable differences in your NEW file when compared with an OLD master file.

Simply update your OLD master by typing the specified character over any areas such as timestamps, product release numbers etc where differences are expected and acceptable, then COMPFILE will ignore the difference.

For example:

OARBC='?' indicates that wherever a question mark is present in the OLD file record, it will be treated as equal to ANY character in the NEW file.

NARBC=X'00' indicates that wherever a NULL value is present in the NEW file record, it will be treated as equal to ANY character in the OLD file.

ARBC allows you set both OARBC and NARBC to the same character at the same time.

When specified on formatted compare, ARBC is applied to character fields only.

`DELC | NDELC | ODELC char`
All occurrences of the character specified using xDELC will be removed before comparison takes place.

For example:

ODELC='-' indicates that all question marks are to be removed from the OLD file record/field before comparison takes place.

NDELC=X'00' indicates that all NULL values are to be removed from the NEW file record/field before comparison takes place.

DELC allows you set both ODELC and NDELC to the same character at the same time.

When specified on formatted compare, DELC is applied to character fields only.

`SPACE`
For **unformatted compare** only, SPACE indicates that a record pair will be considered a match if the only difference is the number of blanks between words in the text. i.e. An equal condition is set when the record pair contains differences but are equal after all multiple blanks have been compressed to a single blank.

`VIEW rectype, ...`
For **formatted compare** only, VIEW specifies one or more record types (*rectype*) that are defined by the *new_struct* structure. Only records in the NEW and OLD files that are assigned one of these record types are selected for compare. All other records are excluded from the compare.

Note that record subsetting using parameters STARTREC/STARTKEY/STARTRBA and FOR is performed prior to the records being filtered by VIEW.

If VIEW is not specified, then all records in the NEW and OLD files are included in the compare regardless of whether they are assigned a record type in *new_struct*.

`SELECT fieldname, ... FROM rectype`
For **formatted compare** only, SELECT specifies one or more field column names (*fieldname*) to be included in the compare and the *new_struct* record types (*rectype*), to which the *fieldname* columns belong.

If a record is assigned a record type that has been selected for compare (see VIEW) and the record type matches one of *rectype* specified by SELECT, then only the *fieldname* columns will be included in the compare for that record. All other field columns are excluded.

If a record is assigned a record type that has been selected for compare and the record type does not match one of *rectype* specified by SELECT, then all field columns belonging to that record will be included in the compare.

If *fieldname* exists in *rectype* for *new_struct* but not in the *rectype* for *old_struct* then that *fieldname* is not included in the compare. If *fieldname* does not exist in *rectype* for *new_struct* then error ZZSD148E is returned.

If SELECT is not specified, then the compare includes all field columns of the same name, belonging to record types of the same name in *old_struct* and *new_struct*.

`STARTREC|FROM | NSTARTREC|NFROM | OSTARTREC|OFROM start_rec`
STARTREC (or FROM) specifies the record number *start_rec* of the first record in **both** the NEW and OLD files at which COMPFILE will begin the compare.

If *start_rec* is greater than the number of records in one file but not the other, then the remaining records in the file with the greater number of records will be flagged as inserts or deletions as appropriate. If *start_rec* is greater than the number of records in both the NEW and OLD file, then no records are compared.

Alternatively, *start_rec* may be specified differently for the NEW and OLD file using NSTARTREC (or NFROM) and OSTARTREC (or OFROM) respectively.

If STARTREC/NSTARTREC/OSTARTREC is not specified, *start_rec* defaults to 1.

`STARTKEY | NSTARTKEY | OSTARTKEY start_key`
For VSAM KSDS, VRDS files or PATHs only, STARTKEY specifies a full or partial key (*start_key*) used to identify the first record in **both** the NEW and OLD files at which COMPFILE will begin the compare.

start_key may be specified as a character or hex string using the standard notations (e.g. abc, 'abc', C'abc' or X'818283'). Note that upper casing of *start_key* will occur if specified as a character string without the "C" (or "c") prefix.

The record selected by *start_key* will be the first record with key field data which is greater than or equal to *start_key*. If STARTKEY selects a record from one file but not the other, then the remaining records in the file with a selected key record will be flagged as inserts or deletions as appropriate. If no record is selected from either file, then no records are compared.

Alternatively, *start_key* may be specified differently for the NEW and OLD file using NSTARTKEY and OSTARTKEY respectively.

If STARTKEY/NSTARTKEY/OSTARTKEY is not specified, *start_key* defaults to a number of hex zeroes (X'00') for the length of the key.

`STARTRBA | NSTARTRBA | OSTARTRBA start_rba`

For VSAM ESDS and RRDS files only, STARTRBA specifies a relative byte address (*start_rba*) used to identify the first record in **both** the NEW and OLD files at which COMPFILE will begin the compare.

start_rba may be specified as a decimal integer or hexadecimal value.

The record selected by *start_rba* will be the first record with a relative byte address which is greater than or equal to *start_rba*. If STARTRBA selects a record from one file but not the other, then the remaining records in the file containing a selected record will be flagged as inserts or deletions as appropriate. If no record is selected from either file, then no records are compared.

Alternatively, *start_rba* may be specified differently for the NEW and OLD file using NSTARTRBA and OSTARTRBA respectively.

If STARTRBA/NSTARTRBA/OSTARTRBA is not specified, *start_rba* defaults to 0.

`FOR | NFOR | OFOR` *`n_recs`*

FOR specifies the number of records (*n_recs*) in **both** the NEW and OLD files to be compared starting at the STARTREC/STARTKEY/STARTRBA record.

Record subsetting using parameters STARTREC/STARTKEY/STARTRBA and FOR is performed before any other record filtering. (e.g. using VIEW)

Alternatively, *n_recs* may be specified differently for the NEW and OLD file using NFOR and OFOR respectively.

If FOR/NFOR/OFOR is not specified, *n_recs* defaults to all remaining records in the file.

Synchronisation Opts

Following a mismatch in record data belonging to the NEW and OLD files, COMPFILE uses synchronisation options to determine the next record in the NEW file and the next record in the OLD file at which the compare will continue.

If no synchronisation method is specified, then Read-Ahead synchronisation is used. i.e. SYNC READHEAD(100 1)

Records in the NEW file that are skipped in order to perform this record synchronisation are flagged as having been **inserted**. Similarly, records that have been skipped in the OLD file are flagged as having been **deleted**.

`SYNC`

An optional keyword specified before the nominated synchronisation method.

`READAHEAD { (`*`ra_limit`* `{`*`ra_match`*`} ) }`

Applicable to both Read-Ahead and Unsorted Key synchronisation, READAHEAD specifies *ra_limit* and optionally *ra_match* values to identify the criteria used by COMPFILE to find a re-synchronisation point in the NEW and OLD files.

ra_limit is the maximum number of records that may be read in order to find a record that matches the mismatched record in the other file. Starting at the record immediately following the mismatched record pair, records are first read from the OLD file and compared with the mismatching record in the NEW file. The process is then repeated with records read from the NEW file compared with the mismathing record in the OLD file.

If a read-ahead match is found, the *ra_match* value is then used to indentify a minimum number of consecutive pairs of matching records that must exist for this to be considered a valid synchronisation point.

If a matching pair of blank records is encountered then, by default, the *ra_match* value is incremented by 1 so omitting these records from the re-synchronisation process. See option SYNCONBLANK to include matching pairs of blank records in the *ra_match* count.

For more details on Read-Ahead synchronisation, see topic File Compare.

Default for *ra_limit* is 100 and default for *ra_match* is 1.
Read-Ahead synchronosation (without key) is default for COMPFILE.

`SYNCONBLANK`

For Read-Ahead synchronisation only, COMPFILE will not include a matching pair of blank records in the *ra_match* count of consecutive matching record pairs. Therefore, a pair of matching blank lines will require a further match in the next pair of consecutive records in order to qualify the intitial match as a synchronisation point.

SYNCONBLANK will bypass this feature so that pairs of matching blank records are included in the *ra_match* count.

`1TO1`

1TO1 specifies that no synchronisation is to be performed by COMPFILE. Each mismatching pair of records is flagged as being changed.

`UNSORTED`

UNSORTED indicates that synchronisation is performed on key data within records that are not necessarily sorted in ascending key order. (See parameter KEY.)

Read-Ahead processing, as described by parameter READAHEAD, is used to synchronise records based only on data contained in the specified key fields.

KEY

KEY indicates that keyed synchronisation is to be performed and also defines one or more segments within the record data from which the key is comprised. Key segments are specified either as fields in a formatted record or as absolute positions and lengths within a formatted or unformatted record.

See File Compare for details of sorted and unsorted key synchronisation.

KEY *fieldname, ...* {FROM *rectype*}

For formatted compare only, specifies one or more field names *fieldname* (or field references) that belong to the nominated record type *rectype*. Each of these fields constitute a key segment within records that are assigned the record type *rectype*.

If a different structure is applied to OLD file data (i.e. USING *old_struct*), then each key *fieldname* must exist in record type *rectype* for both *new_struct* and *old_struct*.

The KEY parameter may be specified repeatedly to define key field segments in more than one record type. If FROM *rectype* is not specified, then the first record type defined in the structure (SDO) is assumed.

KEY (*length new_pos old_pos*)

For formatted or unformatted compare, this form of the KEY parameter expression identifies a key segment length (*length*) and positions within the new (*new_pos*) and old (*old_pos*) file records.

The KEY parameter may be specified repeatedly to define multiple corresponding key segments in the NEW and OLD file records. These key segments apply to all records in the NEW and OLD files. i.e. For formatted record compare, the same key is applied to records assigned any record type in their unformatted state.

EXCHANGED

EXCHANGED indicates that report record types Compare or Compare *record_type* which specifically describe changed records from the NEW (CN) and OLD (CO) files, are to be excluded from the output report.

Default is to include these records in the report.

EXDELETED

EXDELETED indicates that report record types Compare or Compare *record_type* which specifically describe records that have been deleted (D) from the OLD file, are to be excluded from the output report.

Default is to include these records in the report.

EXINSERTED

EXINSERTED indicates that report record types Compare or Compare *record_type* which specifically describe records that have been inserted (I) into the NEW file, are to be excluded from the output report.

Default is to include these records in the report.

EXFIELDCHANGED

Applicable to formatted compare only, EXFIELDCHANGED indicates that report record type "Field" which specifically identify the names of fields containing changed data (C), are to be excluded from the output report.

By default, for formatted compare only, a record flagged as changed will produce one or more of changed field report records each identifying the name of a field containing changed data. If EXCHANGED is not specified, these report records immediately follow the CN/CO report records.

Note that, specifying EXFIELDCHANGED may result in a significant performance improvement since the process of comparing field-by-field is terminated at the first mismatch in each record.

Each "Field" record identifies the name of a mapped field as well as its new and old values in external format.

See also "FCHUNK" which may be used to override the width of the values displayed.

INCMATCHED

INCMATCHED indicates that report record types Compare or Compare *record_type* which specifically describe matching record pairs, are to be included in the output report.

Default is to exclude these records from the report.

INCFIELDS

Applicable to formatted compare only, INCFIELDS indicates that report record type "Field" is eligible to be included for all matching, inserted or deleted records.

Each "Field" record identifies the name of a mapped field as well as its new or old value in external format.

See also "FCHUNK" which may be used to override the width of the value displayed.

FCHUNK *n_cols*

Applicable to formatted compare with LIST=FMT only, FCHUNK overrides the default width of 50 allocated to display the external formatted new and/or old value(s) on a "Field" record (see EXFIELDCHANGED and INCFIELDS options).

Each "Field" record identifies the name of a mapped field as well as its new or old value in external format.

Wherever the length of the external value of a given field exceeds the FCHUNK value then the complete value will be spread over multiple "Field" records.

Note that for LIST=TEXT this option does not apply:

◊ For changed records the old and new values are always displayed in 50 byte chunks.
◊ For matches, inserts and deletes the old or new value is always displayed in 100 byte chunks.

`INCKEYS`

INCKEYS indicates that, for sorted and unsorted KEY synchronisation only, records containing a matching key field will be displayed in the output report, regardless of whether INCMATCHED has been specified.

This is of primary use when performing a formatted compare of hierarchical records where records are assigned to different record types and keys fields are defined in multiple record types.

Default is to exclude these records from the report.

`CONTEXT n_lines`

CONTEXT indicates that the specified number of (possibly matching) records immediately before and after each detected difference are to be included in the report in order to provide context (without including all matching records, which for large files is likely to be prohibitive).

Note that this option will be ignored if INCMATCHED is specified.

`GAP n`

In conjunction with CONTEXT, GAP indicates that the specified number of context block separator records are to be included in the report.

For unformatted compare gap record are of record-type "Compare", with the "zId" set to " -" and all other fields blank.

For formatted compare a specific "Gap" record-type is used comprising a single field "zId" set to " -".

Note that this option will be ignored if INCMATCHED is specified.

`LIMIT n_diffs`

LIMIT specifies a number of mismatches *n_diffs* that may occur before COMPFILE is terminated. If *n_diffs* is 0 (zero) or LIMIT is not specified, then there is no limit to the number of differences and COMPFILE only terminates after all records selected for compare have been processed.

`LIST=FMT|TEXT`

The report generated by the compare files utility is available in two formats controlled by the **LIST=FMT|TEXT** option.

LIST=FMT is the default when **COMPFILE** is executed within the **SELCOPYi online environment**. Using this option the report generated is a **structured data file**. This is designed to be (automatically) browsed (not printed) from within a SELCOPYi online session, using a structure definition (SDO) which is also generated automatically during execution of the compare.

LIST=TEXT is the default when **COMPFILE** is executed from JCL as a batch job. Using this option the report generated is a more traditional formatted text document, designed to be printed if necessary.

`REPORT | RPT fileid`
`NOREPORT | NORPT`

REPORT specifies that the COMPFILE report is to be written to the specified sequential data set or PDS/PDSE member *fileid*. NOREPORT specifies that no report output is to be generated (only console messages are displayed).

If *fileid* exists but is uncataloged, then include the required volser as part of the fileid specification in the format *volser:data.set.name*.

The report is a structured data file designed to be browsed (not printed) using a SELCOPY/i structure definition object (SDO), which is also generated by COMPFILE. The associated SDO fileid is constructed simply by adding '.SDO' to the DSN of the sequential or partition dataset name specified by *fileid*. The DSN is therefore restricted to 40 bytes in length. e.g. If *fileid* is ZX1234.SELCOPYI.COMPF.REPORT(XYZ001), the allocated SDO is ZX1234.SELCOPYI.COMPF.REPORT.SDO(XYZ001).

If the report file *fileid* and/or the SDO file do not already exist, then they will automatically be allocated by COMPFILE relying on SMS ACS to select a suitable storage group of eligable DASD volumes. File *fileid* is allocated using DCB geometry RECFM=VB, LRECL=32756, BLKSIZE=0 and a space allocation of TRACKS(150,75). SDO is allocated using DCB geometry RECFM=VB, LRECL=16380, BLKSIZE=0 and a space allocation of TRACKS(2,2).

If REPORT is not specified, *fileid* defaults to "*user*.SELCOPYI.COMPFILE.REPORT" with SDO fileid "*user*.SELCOPYI.COMPFILE.REPORT.SDO".

`NEWPAGE n_lines`

Applicable to formatted compare with LIST=FMT only, the NEWPAGE option will cause a new page to be thrown if *n_lines* lines are not still available on the current page at the point where print output for a new record (or syncronised record pair) is being started.

Note that the page depth is defined by the prevailing Data-Edit **"SET PAGEDEPTH n"** option.

`PRINTREPORT BATCH | INTERACTIVE | ALL | NONE limit` **`--- OBSOLETE ---`**

PRINTREPORT is now obsolete having been superseded by the LIST=TEXT option.

If REPORT is used, PRINTREPORT controls whether or not the generated COMPFILE structured report output is to be printed. Keyword parameters determine the environment(s) in which execution of the COMPFILE command will generate

printed output.

BATCH	Batch only using program SDEAMAIN. (Default)
INTERACTIVE	Interactive only (VTAM, TSO or ISPF)
ALL	Either batch or interactive environments.
NONE	Suppress print for all environments.

The optional *limit* parameter may be specified as a numeric integer literal indicating the maximum number of pages of print output allowed. A limit value of 0 (zero) indicates that there should be no restriction placed on the number of output pages. This is the default.

If print output is required, the following PRINT utility command is executed at end-of-job.

```
PRINT FILE TRUNC LIMIT <limit>
  INDSN(      <ReportFileid>
        USING <ReportFileid>.SDO
        INITCMD ("select zFileId,zRecNo,zRecord,*  from Hit"
                 "VFMT"
                 "PUSH SAVEOPTIONS"
                 "SAVEOPTIONS OFF"
                 "RECLEN OFF"
                 "REFERENCE OFF"
                 "PREFIX OFF"
                 "POP SAVEOPTIONS"
                )
      )
```

Output File Opts

Records from the NEW and OLD files may be copied to different output files based on their status flag as assigned by COMPFILE.

The fileid specified in each of the following options must belong to a sequential data set, PDS/PDSE member or HFS file which is not already assigned an exclusive ENQ. If the output file is non-HFS and is not yet allocated, then COMPFILE will allocate the file as a cataloged data set with space allocation of TRACKS(150,75) and data set geometry modelled on the NEW or OLD file as appropriate.

If the specified output file exists but is uncataloged, then include the required volser as part of the fileid specification in the format *volser*:*data.set.name*.

COMPFILE will fail with error ZZSD027E if the same fileid is specified on more than one of the following output file options.

`WRITECHANGEDNEW|WRITECN` *`cn_fileid`*
: Indicates that records from the NEW file, flagged as having been changed (CN), are to be copied to *cn_fileid*.

`WRITECHANGEDOLD|WRITECO` *`co_fileid`*
: Indicates that records from the OLD file, flagged as having been changed (CO), are to be copied to *co_fileid*.

`WRITEMATCHEDNEW|WRITEMN` *`mn_fileid`*
: Indicates that records from the NEW file, flagged as matching records in the OLD file, are to be copied to *mn_fileid*.

`WRITEMATCHEDOLD|WRITEMO` *`mo_fileid`*
: Indicates that records from the OLD file, flagged as matching records in the NEW file, are to be copied to *mo_fileid*.

`WRITEINSERTEDNEW|WRITEIN` *`in_fileid`*
: Indicates that records from the NEW file, flagged as having been inserted (I), are to be copied to *in_fileid*.

`WRITEDELETEDOLD|WRITEDO` *`do_fileid`*
: Indicates that records from the OLD file, flagged as having been deleted (D), are to be copied to *do_fileid*.

`CASEIGNORE|CASEINSENSITIVE`
: Both CASEIGNORE and CASEINSENSITIVE specify that the compare of record data treats a lower case alpha character as being a match with its equivalent upper case character. For formatted compare, this option applies to character (AN) fields only.

Parameters CASEIGNORE and CASEINSENSITIVE do not affect key synchronisation where alpha characters in the key segments must match exactly.

Examples:

Use of the COMPFILE command may result in long command streams. Therefore, it is recommended that any COMPFILE command you enter (or generate via the Compare File dialog panel) should be entered as text in your HOME command centre (CMX) data set. Doing this will save retyping the command in order to make any required syntax changes before re-execution.

```
<compf  CBL.CBLI310.ASM(WINFIPO0)  CBL.CBLI190.ASM(WINFIPO0)  sync readahead(100 1)
```

Compare two versions of the same Assembler source module to identify record changes, inserts and deletions that have been made between program releases. Read-Ahead synchronisation is used since code insertions/deletions are

expected.

```
<compfile    CBL.SELC300.SZZSMAC(ZZSNAME)                    \
             CBL.SELC300.AZZSMAC(ZZSNAME)                    \
             sync readahead(40 1)                            \
             nstartrec 10      ostartrec 15      for 100     \
             startcol 1   comparelen 72
```

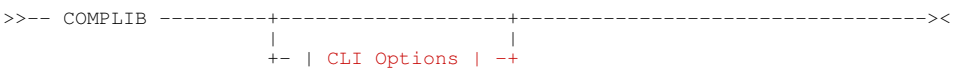
Compare the target and distribution library copies of the SELCOPY Product Suite CBLNAME macro to identify whether differences exist. Macro contains additional 5 lines of comment data in the distribution (OLD file) library so different STARTREC values are required (OSTARTREC/NSTARTREC). Compare excludes sequence numbers that may exist in columns 73-80 using a STARTCOL/COMPARELEN combination.

```
<compfile    NBJ2849.AMCUST.G01647.DA                        \
             NBJ2849.AMCUST.G01646.DA                        \
             using  cobol  MAST649.AM.COB(AMCUST)            \
             view    ADDR_REC                                \
             select  CUSTID, ADDR1, POSTCODE  FROM ADDR_REC  \
             key     CUSTID                   FROM ADDR_REC
```

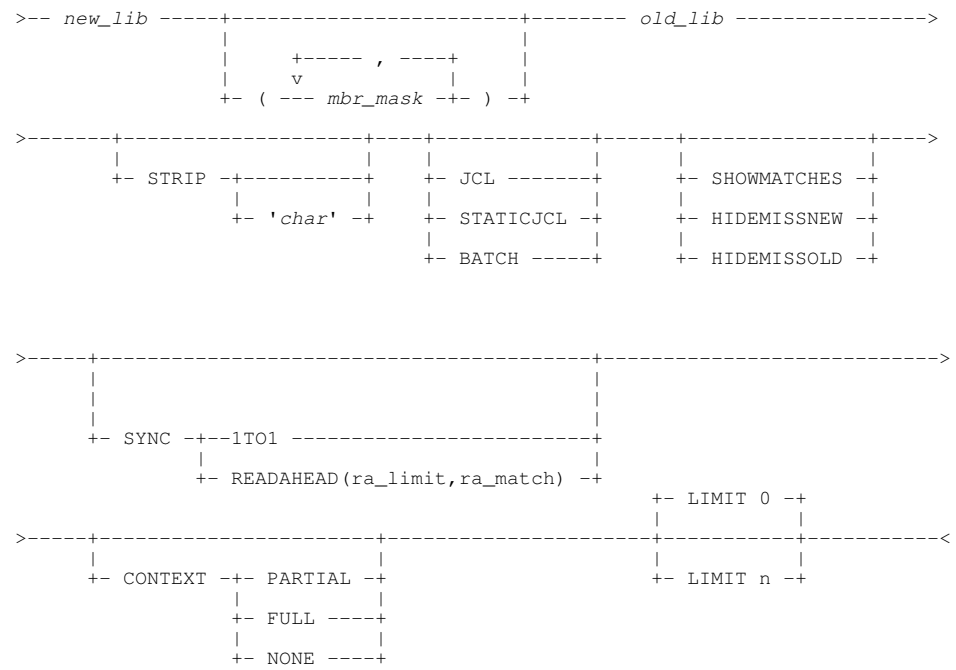
Compare NEW and OLD versions of a customer records file containing records mapped by a cobol copybook record structure (01 level), ADDR_REC. The records are sorted in ascending order on key field, CUSTID, which is used here for re-synchronisation of corresponding record pairs. Only fields CUSTID, ADDR1 and POSTCODE are compared.

COMPLIB

Syntax:



CLI Options:



Description:

The COMPLIB command invokes the Compare Libraries utility to perform a compare of two PDS/PDSE libraries in the foreground, generate a batch job to perform the compare or, if executed with no parameters, open the Compare Libraries Panel.

As well as being run interactively under SELCOPYi, COMPLIB may also be run in batch as an SDEAMAIN command.

COMPLIB executes the SELCOPY program to compare selected library members and generates an output report suitable for edit using the SELCOPY/i (CBLe) text editor.

Parameters:

new_lib
Specifies the source PDS/PDSE library(s) containing the NEW images of library member data to be compared.

For a single library, this should be the fully qualified PDS/PDSE library DSN.

A library search path may be specified by specifying an allocated DD name for the a library concatenation. (e.g. SYSEXEC)

This means that only the first occurrence along the directory path of each member that match the specified **mbr_mask(s)** will be processed.

Members are processed in member name order.

mbr_mask
Specifies a member name mask which is to be used by COMPLIB to select members from the NEW and OLD libraries to be included in the compare operation.

mbr_mask may optionally contain the following wild card characters:

- `*` A single asterisk represents an entire member name or zero or more characters within a member name mask.
- `%` A single percent sign represents exactly one character within a member name mask. Up to 8 percent signs can be specified in each member name mask.

If specified, the member name mask must immediately follow the PDS/PDSE *new_libdsn* and be enclosed in "( )" (parentheses). Multiple member name masks, all specified within the single set of parentheses, must be separated by a "," (comma) with no intervening blanks

Default is to compare **all** members of both the NEW and OLD libraries.

old_lib
Specifies the source PDS/PDSE library(s) containing the OLD images of library member data to be compared.

For a single library, this should be the fully qualified PDS/PDSE library DSN.

A library search path may be specified by specifying an allocated DD name for the a library concatenation. (e.g. SYSEXEC)

This means that only the first occurrence along the directory path of each member that match the specified **mbr_mask(s)** will be processed.

Members are processed in member name order.

STRIP < *char* >
STRIP indicates that trailing characters in the compare data are to be stripped from the longer record, to the length of the shorter record, when the records to be compared are of different lengths. The optional strip character *char* may be specified in quotes (") or apostrophes (').

If STRIP is not specified, then no characters are stripped from the compare data. If STRIP is specified without *char*, the default strip character is blank ' ' (X'40').

JCL
JCL specifies that COMPLIB should generate JCL to execute itself in batch using PGM=SDEAMAIN.

The JCL will be displayed in a temporary Text-Edit window view.

STATICJCL
BATCH specifies that COMPLIB should generate JCL to directly execute PGM=SELCOPY in batch.

Concatenated SYSIN input includes the STRIP option status and strip character, the input control statements (ZZSCOMPL) and a list of selected members reflecting the state of the NEW/OLD libraries at the time the batch job is generated.

For this reason, it is recommended that the JCL parameter is used in preference to STATICJCL, which is supported only for compatiibilty with older versions of the product which did not support the COMPLIB command running directly under PGM=SDEAMAIN.

The JCL will be displayed in a temporary CBLe text edit window view.

BATCH
BATCH is a synonym for STATICJCL supported for compatibilty with older versions.

SHOWMATCHES
Select this option to report members that exist in both NEW and OLD library and whose contents match.

HIDEMISSNEW
Select this option to exclude from the report members that exist in the OLD library but are missing from the NEW library.

Performance may be improved by selecting this option, since once all selected members from the NEW library have been processed then Compare Libraries may terminate without processing any remaining members of the OLD library.

`HIDEMISSOLD`
Select this option to exclude from the report members that exist in the NEW library but are missing from the OLD library.

Performance may be improved by selecting this option, since once all selected members from the OLD library have been processed then Compare Libraries may terminate without processing any remaining members of the NEW library.

`SYNC 1TO1 | READAHEAD(`*`ralimit,ra_match`*`)`
Defines the synchronisation type to be used on the generated CompFile command. Select either **Read-Ahead** or **1-to-1**.

For **Read-Ahead**, you may specify the limit and match values in brackets immediately following provided no blanks are used. *ralimit* and *ra_match* must be separated by a single comma.

`LIMIT` *`n`*
Defines the LIMIT parameter to be used on the generated CompFile command.

Use this option in order terminate the CompFile process as soon as the specified number of record mismatches has been encountered.

A value of zero (default if LIMIT not specified) then no limit is placed, and therefore the whole of each file is processed.

Note that Compare Libraries itself processes each member only to the point required to establish a single mismatch.

`CONTEXT PARTIAL | FULL | NONE`
Defines the CONTEXT parameter to be used on the generated CompFile command.

PARTIAL
Show 10 lines before and after each difference in order to provide additional context without displaying all matching records. Each difference "block" will be followed by 2 "gap" records designed to provide visual separation.

This option corresponds to the COMPFILE parameters "CONTEXT 10" and "GAP 2".

For full control over the number of context and gap records select Use Extended Options below.

FULL
All corresponding records from both NEW and OLD files that match are to be included in the output report file.

This option corresponds to the COMPFILE parameter "INCMATCHED".

NONE
No matching records are to be displayed.

Examples:

```
COMPLIB  NBJ.CBLI310.ASM  NBJ.CBLI31B.ASM
```

Compare all members of PDS library NBJ.CBLI310.ASM with all members of PDS library NBJ.CBLI31B.ASM.

```
COMPLIB  NBJ.SMPEINST.JCL(RX* QS*)  NBJ.SMPEINST.JCL.BKUP  BATCH
```

Generate a batch SELCOPY job to compare all members with names beginning "RX" or "QS" in PDS library NBJ.SMPEINST.JCL with equivalent members of PDS library NBJ.SMPEINST.JCL.BKUP.

COMPTABLE

Syntax:

```
>>--- COMPTABle ------------------------------------------------------------><
```

Description:

COMPTABLE displays the Compare DB2 Base/Result Tables panel.

CRETRIEV

Syntax:

```
>>--- CRETRIEV ------------------------------------------------------------><
```

Description:

CRETRIEV conditionally retrieves a previously executed command.

If the cursor is located within a command input area (command line), then a command is retrieved from the command stack and displayed at the command prompt. Otherwise, the cursor is placed in the first column of the command input area.

CRETRIEV is assigned to **F12** by default. Repeated execution will retrieve commands, one at a time, from the stack of commands, starting at the most recently executed and scrolling backwards chronologically. Following display of the very first command executed in the current SELCOPY/i session, CRETRIEV will cycle back to the most recently executed command.

Parameters:

CRETRIEV has no parameters.

CSVGEN

Syntax:

```
>>- CSVgen -+-------------------------------------------------------------+--><
            |                                                             |
            +- | Input dataset specification | --+-- | Common Options | --+
            |                                    |
            |          +- .ZFIRST -- .ZLAST --+  |
            | (1)      |                      |  |
            +-- * ----+-----------------------+--+
                      |                       |
                      +- .name1 --+----------+
                                  |          |
                                  +- .name2 -+
```

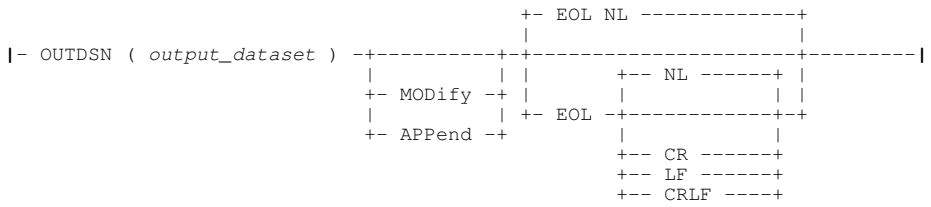
Input dataset specification:

```
 |- INDSN ( input_dataset_specification ) ----------------------------------|
```

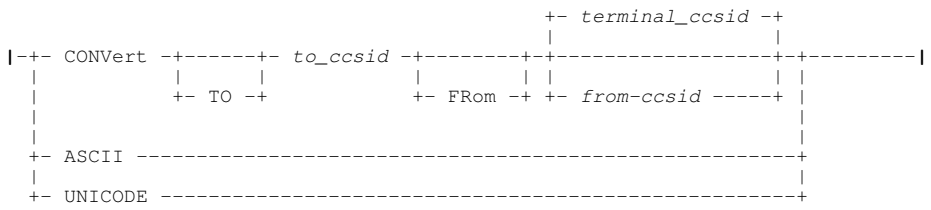
Common Options:

```
 |----------+--------------------------------------+-------------------------->
            |                                      |
            +-| Output dataset specification    |-+

 >----------+--------------------------------------+-------------------------->
            |                                      |
            +-| Character conversion option     |-+


    +- SEParator , --------+  +- QUOTE CHARacter -+  +- NOStrip -+
    |                      |  |                   |  |           |
 >--+----------------------+--+-------------------+--+-----------+------------>
    |                      |  |                   |  |           |
    +- SEParator sep_char -+  +- QUOTE ALL -------+  +- STRIP ---+
                              |                   |
                              +- QUOTE REQuired --+


                                        +- HEADer   -+   +- BROWSE -+
                                        |            |   |          |
 >-+------------------------------------+---+--------+---+----------+-----|
   |                                    |   |        |   |          |
   +- RECord -+--------+- record_type -+   +- NOHEADer -+   +- EDIT ---+
              |        |                                    |          |
              +- TYpe -+                                    +- NOVIEW -+
```

Output dataset specification:



Character conversion option:



Notes:

1. Asterisk (*) is optional if any other parameter is specified.

Description:

CSVGEN is used to open the general CSV Generation panel or to immediately produce an exportable copy of structured data as Comma Separated Variable text. The structured data may belong to a specific structured data set or the current SDE view.

If asterisk (*), or any other parameter other than INDSN, is specified, then data in the current SDE edit or browse view will be used as input for CSV generation. If no parameters are specified, the **CSV Generation** panel is opened.

The CSVGEN command may be executed in the foreground or via SDEIN input to program SDEAMAIN for batch processing.

During foreground execution a progress window is displayed showing input and output record counts, updated every second, which allows the user to interrupt processing before completion using the attention key.

Parameters:

Input dataset specification (INDSN)
Use of a CSVGEN input data set nominates a specific data set from which records are to be selected for CSV generation.

The CSVGEN input dataset may be specified explicitly in the command as the argument of the **INDSN** keyword. If the **INDSN** keyword is not present in the command then the contents of the current SDE structured dataset browse or edit view are used. If there is no current structured dataset the CSV Generation panel is opened.

INDSN (*input_dataset_specification*)
The input dataset specification is in the form of a structured edit BROWSE command (the BROWSE command verb is not required) which must be enclosed in parentheses following the INDSN keyword.

BROWSE keyword options such as **FROM**, **FOR**, **FILTER** and **VIEW** may be specified to limit the records from the input dataset which will be copied to the output dataset.

*
Required only if no other CSVGEN parameters are specified in order to immediately generate CSV for data from the current SDE view (using defaults) as opposed to opening the general purpose CSV Generation panel or the SDE CSV Generation Panel as appropriate.

Start/End Line labels (.name1/.name2)
Applicable only to CSV generation from data in the current SDE view, start and end line labels may be used to select a range of data records to be processed.

.name1
Corresponds to a label name *.name1* that identifies the first line in a range of SDE edit/browse lines. The preceding "." (period/dot) is mandatory.
Default is .ZFIRST.

.name2
Corresponds to a label name *.name2* that identifies the last line of a range of SDE edit/browse lines. The preceding "." (period/dot) is mandatory.
.name2 may occur on a line with a lower line sequence number than *.name1*. This is functionally equivalent to specifying *.name2* before *.name1* on the CSVGEN command.
Default is .ZLAST.

Output dataset specification (OUTDSN)

The CSVGEN output dataset may be specified explicitly in the command as the argument of the **OUTDSN** keyword. If the **OUTDSN** keyword is not present in the command then the value of the INI file variable **SDE.CSVGENOUTDSN** is used if it exists, otherwise a default dataset name **userid.ZZS.CSVGEN** is used.

OUTDSN (*output_dataset*)

The output dataset name. Parentheses around the dataset name are accepted but not required. If this dataset exists its organisation may be sequential, a partitioned dataset member, VSAM (except LDS and KSDS) or HFS (ZFS). If it does not exist and CSVGEN is executing in batch the command terminates with an error message. If it does not exist and CSVGEN is executing interactively the user will be asked to allocate it (unless the name represents an HFS file in which case it will be implicitly defined).

This parameter may also be a DD name. If **output_dataset** consists of 8 or fewer characters and represents an allocated DD name then this allocated dataset is used for output.

MODIFY | **APPEND**

The output will be appended to the dataset if it exists (and is not partitioned). If this keyword is not specified the output will overwrite any existing dataset content.

EOL NL|CR|LF|CRLF

HFS file end of line specification. This parameter is accepted but ignored if the output dataset is not an HFS file. The values here are specified in EBCDIC, but if the output is subject to character conversion, the line end characters will also be converted. Note that ASCII does not have a NL (newline) character so if the output is being converted to a non-EBCDIC CCSID NL is changed to CR.

NL	X'15'	EBCDIC New Line. This is the default for EBCDIC output to an HFS file.
CR	X'0D'	EBCDIC Carriage Return.
LF	X'25'	EBCDIC Line Feed.
CRLF	X'0D25'	EBCDIC Carriage Return Line Feed.

Character conversion option (CONVERT/ASCII/UNICODE)

Since the purpose of CSVGEN is to produce a portable export version of the data in a z/OS mainframe structured data file, and the output is character data, the coded character set identifiers (CCSIDs) of the input, output and of the CSVGEN internal constants themselves are of significance.

Even if the input and output is coded in an EBCDIC CCSID, these may differ, and both may differ from the CCSID of the CSVGEN internal constants. Since some of the special characters used in CSV have different code points in different EBCDIC CCSIDs (for example square brackets) these must be dealt with consistently to produce correct CSV output.

CSVGEN uses the z/OS character conversion support supplied by IBM modules CUNLINFO (for obtaining CCSID information) and CUNLCNV (for character conversion from one CCSID to another).

The internal CSVGEN CCSID (that of the constants used to build the CSV syntax) is CCSID 285 (EBCDIC, SBCS UNITED KINGDOM).

CSVGEN assumes a default CCSID as follows:

Interactive

When executed interactively CSVGEN uses as default input CCSID that of the user's 3270 terminal.

Batch

When executed in batch CSVGEN uses as default input CCSID the value of the INI file variable **SDE.CCSID**. This variable is set automatically to the user's 3270 terminal CCSID (if not already set) during an interactive session. It can also be set using the structured data SET CCSID command.

If no explicit conversion is specified the CSV output dataset is produced using the default CCSID and the input dataset character fields are assumed to be in the same CCSID. The internal CSVGEN constants are converted from internal CCSID 285 to the default CCSID.

CONVERT

Use this keyword to request character CCSID conversion.

to_ccsid

The CCSID of the output CSV text dataset. Internal CSVGEN character literals and input character data fields (and HFS line end characters if used) are converted to this CCSID.

from_ccsid

The input character data fields are converted from this CCSID. If this parameter is not supplied the default input CCSID is used.

ASCII

Convert the output to ASCII. This is equivalent to specifying **CONVERT TO 819**. CCSID 819 is ISO 8859-1 ASCII.

UNICODE

Convert the output to UNICODE (UTF-16). This is equivalent to specifying **CONVERT TO 1200**. CCSID 1200 is the IBM bigendian UTF-16 CCSID which is automatically transformaed to the most recent UTF-16 standard.

Start/End Line names (.ZFIRST/.ZLAST)

If **INDSN** is not specified then input is taken from the current structured dataset (browsed or edited) in which case start and end line names may be specified. Defaults for start and end are **.ZFIRST** and **.ZLAST** respectively.

Separator character option (SEPARATOR)
By default CSVGEN produces **comma** separated variables, but this option allows the user to specify any other character as the variable separator. The argument to the **SEPARATOR** option may be specified as a single quoted or unquoted character literal, or as a hex value using **X'nn'** notation.

Record-Type option (RECORD TYPE)
CSVGEN produces output for a single input record-type as defined by the input copybook/structure, with all input records of other record-types being ignored. Where the structure defines more than one record-type the **RECORD TYPE** option allow the user to choose which record-type to select, the default being the record-type of the first selected input record.

Quoted strings option (QUOTE)
The **QUOTE** option controls when variable values are to be enclosed in double-quotes.

CHARacter	Quote character fields values only (default).
ALL	Quote all field values.
REQuired	Quote only if required i.e. if value contains a double-quote or the separator character.

Strip trailing blanks option (STRIP/NOSTRIP)
The **STRIP** option controls whether trailing blanks are to be stripped from each variable. This option is particularly relevant to fixed length character fields. Default is NOSTRIP.

Output field names record option (HEADER/NOHEADER)
This option controls whether a CSV record containing the original field names is generated as the first output record.

HEADER
Output a field names header record. This is the default.

NOHEADER
Do not output a field names header record.

Output view option (BROWSE/EDIT/NOVIEW)
When CSVGEN is run interactively this option allows the user to request to view the output when the command completes.

BROWSE
Browse the output CSV dataset. This is the default when run interactively.

EDIT
Edit the output CSV dataset using the SELCOPY/i text editor.

NOVIEW
Do not view the output CSV dataset. This option is forced when run in batch.

Examples:

In the following example the COBOL copybook CBL.COB(CSVEXAMP) is used to map the 3 records in the file CBL.CSVEXAMP

The content of the COBOL copybook CBL.COB(CSVEXAMP) is shown as it would appear in text edit:

```
-CBL.COB(CSVEXAMP)     80 F PDSE   Size=10   Alt=0,0;1
       <---+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000000 * * * Top of File * * *
000001       ** CBL.COB(CSVEXAMP) ***     L=001 --- 2013/10/08 14:45:56  (CBL)
000002
000003        01  CSVExample.
000004            05  Employee.
000005                07  FirstName      Pic x(20).
000006                07  LasttName      Pic x(20).
000007            05  Age                Pic s9(3) comp.
000008            05  Salary             Pic s9(7)v99 comp-3.
000009 * * * End of File * * *
```

The content of the file CBL.CSVEXAMP is shown as it would appear in structured data edit with HEX ON in effect to show the content of numeric fields:

```
-Edit   CBL.CSVEXAMP using CBL.COB(CSVEXAMP)          47 F SEQ
00000000 *** Top of Data ***
Record type: CSVEXAMPLE   Fixed(47) Offset=0 Data elements=6

         FIRSTNAME            LASTTNAME               AGE      SALARY
         #3                   #4                       #5          #6
         <---+----1----+----> <---+----1----+----> <---+> <---+----1>
00000001 John                 Doe                      52    33000.00
         D9894444444444444444 C9844444444444444444     03       03000
         1685000000000000000 4650000000000000000      04       0300C
```

```
00000002 Amy                  Johnston                28    41500.00
         C9A44444444444444444 D989AA99444444444444    01       04500
         14800000000000000000 16852365000000000000    0C       0100C

00000003 Freda                Bloggs                  39    27800.00
         C9888444444444444444 C9988A44444444444444    02       02800
         69541000000000000000 23677200000000000000    07       0700C

00000004 *** End of Data ***
```

The following CSVGEN command then produces the output file CBL.CSVGEN:

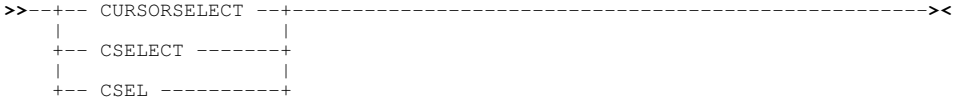
```
     csvgen  indsn(cbl.csvexamp using cobol cbl.cob(csvexamp))
             outdsn(cbl.csvgen)
             nocomment indent 3 edit
```

The CSVGEN output file is edited as a result of the EDIT keyword parameter in the command. Note that each record has an associated record type (level 1) tag CSVEXAMPLE with the SEQ_NUMBER attribute identifying the record number:

```
-CBL.CSVGEN     27990 V SEQ   Size=26   Alt=0,0;0
       <---+----1----+----2----+----3----+----4----+----5----+----6----+--
000000 * * * Top of File * * *
000001 <INPUT FILE="CBL.CSVEXAMP" FORMAT="CBL.COB(CSVEXAMP)">
000002    <CSVEXAMPLE SEQ_NUMBER="1">
000003       <EMPLOYEE>
000004          <FIRSTNAME>John</FIRSTNAME>
000005          <LASTTNAME>Doe</LASTTNAME>
000006       </EMPLOYEE>
000007       <AGE>52</AGE>
000008       <SALARY>33000.00</SALARY>
000009    </CSVEXAMPLE>
000010    <CSVEXAMPLE SEQ_NUMBER="2">
000011       <EMPLOYEE>
000012          <FIRSTNAME>Amy</FIRSTNAME>
000013          <LASTTNAME>Johnston</LASTTNAME>
000014       </EMPLOYEE>
000015       <AGE>28</AGE>
000016       <SALARY>41500.00</SALARY>
000017    </CSVEXAMPLE>
000018    <CSVEXAMPLE SEQ_NUMBER="3">
000019       <EMPLOYEE>
000020          <FIRSTNAME>Freda</FIRSTNAME>
000021          <LASTTNAME>Bloggs</LASTTNAME>
000022       </EMPLOYEE>
000023       <AGE>39</AGE>
000024       <SALARY>27800.00</SALARY>
000025    </CSVEXAMPLE>
000026 </INPUT>
000027 * * * End of File * * *
```

CURSORSELECT

Syntax:



Description:

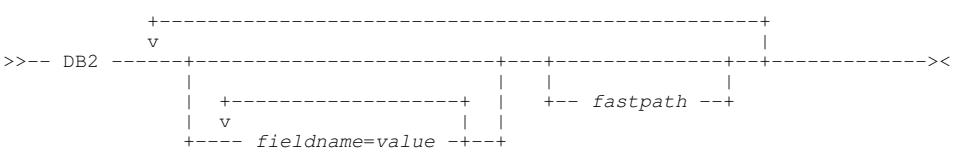
CURSORSELECT is a generic command that performs the default operation for the field at the cursor position.

The default operation depends on the function of the window, the window class and the field in which the cursor is positioned. The default operation for a particular window/field may be found in the documentation for the relevant window type.

CURSORSELECT is intended to be assigned to a function key to execute the same operation as that performed when the >Enter< key is hit.

DB2

Syntax:



Description:

Display the SELCOPY/i DB2 primary option menu panel and, optionally, nested sub-panels. If a fast path is specified or, if no fast path is specified but a value is specified for field SSN (SSN=*xxxx*), then a DB2 connection is performed automatically. In all other cases, no DB2 connection is attempted until the user hits <Enter> (or any PFKey) to accept the DB2 SSN and SQLID displayed in the DB2 primary option menu panel.

Parameters:

fastpath
: Each SELCOPY/i DB2 panel may be accessed via one or more levels of nesting into menu panels. Fast path menu item selection allows immediate nesting into panel menus to open the required SELCOPY/i DB2 panel without the need to select menu items individually.

  Note that each nested panel in the fast path will be displayed. Furthermore, a connection to the specified subsystem (SSN=*xxxx*) or the last accessed or default DB2 subsystem, will occur automatically with SQLID set to the corresponding value in the DB2 primary option menu panel.

  fastpath is an integer value representing a menu choice. If no update of input fields (*fieldname=value*) is to be performed between successive *fastpath* values, then successive *fastpath* values may be separated by a single dot/period (.) without intervening blanks. e.g. DB2 7.1

fieldname=value
: *fieldname* nominates a field name within the current nested DB2 panel, into which the corresponding *value* will be inserted. If *value* contains special characters or blanks, then it must be contained within apostrophes (') or quotation marks (").

  A separate *fieldname=value* combination may be specified for each input field in the resulting panel.

  Beware that specifying a fast path in addition to a *fieldname=value* combination will immediately execute the command assigned to the resultant panel. e.g. CREATE STOGROUP, DROP TABLE.

Examples:

DB2 5
: Open the CREATE Object option panel.

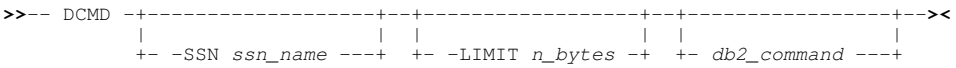
DB2 SQLID=NBJ002
: Open the DB2 primary option menu panel and insert "NBJ002" in the SQLID field. Connection will **not** occur when the panel is opened until <Enter> or a PFKey is hit.

DB2 SSN=DB2B 1 DB2CMD="-DISPLAY BUFFERPOOL(BP1)"
: Open the DB2 primary option menu panel, insert "DB2B" in the SSN field and perform DB2 connect. Then open the DB2 Command panel, insert "-DISPLAY BUFFERPOOL(BP1)" in the DB2CMD field, and immediately execute the command to display a list of DB2 buffer pools.

DB2 SSN=DB2C 7.1 NAME=DSN%
: Open the DB2 primary option menu panel, insert "DB2C" in the SSN field and perform DB2 connect. Then open the List Storage Groups panel, insert "DSN%" in the NAME field, and immediately execute to display a list of storage groups with name beginning "DSN".

DCMD

Syntax:



Description:

Use the DCMD command to open the Execute DB2 Commands (ZZS2XDB2) panel.

If the user has already opened a DB2 Primary Option Menu in the current SELCOPY/i session and so a DB2 connection already exists for the specified DB2 subsystem, then no new connection is made. Otherwise, a connection is made to this DB2 subsystem when the panel is opened.

Note: Not implemented for CMS or VSE.

Parameters:

`-SSN ssn_name`
: The DB2 subsystem in which the specified DB2 command will be executed.

  Default is the last DB2 subsystem to which the user connected in this or previous SELCOPY/i sessions. If no previous DB2 connection has been made using SELCOPY/i, the default is the value of the SELCOPY/i INI option, DB2.SSN, or else the subsystem name specified in the DB2SubSys field of the CBLNAME load module.

`-LIMIT n_bytes`
: Limits the number of bytes used for the DB2 command output data buffer.

  Where the length of data returned by the command exceeds the output buffer size, then error message ZZSX016W is returned indicating the number of bytes of output data returned, and number of bytes not returned by the command.

  The *n_bytes* value is placed in the "**Byte Limit>**" field of the Execute DB2 Command panel and the default limit is 0 (no limit).

db2_command
: A valid DB2 command to be executed when the Execute DB2 Command panel is opened.

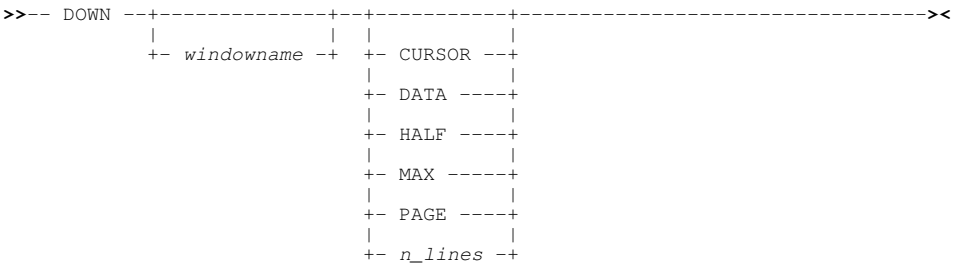
  The *db2_command* string is placed in the "Command>" field of the Execute DB2 Command panel and the default is a null string.

Examples:

`DCMD -SSN DB8G -DISPLAY BUFFERPOOL(BP0)`
: Display the status of buffer pool BP0 in subsystem DB8G.

DOWN

Syntax:



Description:

Scroll the view of the data within the specified window downwards towards the bottom of the displayable data.

The extent by which data is scrolled is determined by the CURSOR, DATA, HALF, PAGE, MAX or *n_lines* parameter which may be specified using any one of three methods determined in the following order of precedence:

1. The scrolling command verb, DOWN, and one of these scrolling parameters is explicitly specified on the command line.
2. The scrolling parameter is specified on the command line and a PFKey assigned to DOWN is actioned.
   Note that the contents of a command line are appended to the command stream assigned to a PFKey when that PFKey is actioned.
3. No scrolling parameter is specified, so the current value of the "Scroll>" field is used.
4. No scrolling parameter is specified and no "Scroll>" field is present, so a defualt of one line is used.

By default this command is assigned to function key **PF8**.

Parameters:

windowname
: The window name of the window in which the display is to be scrolled. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

CURSOR
: The line on which the cursor is positioned becomes the first line of the scrolled display.
If the cursor is positioned outside the display area or on the first line within the display area, then DOWN PAGE is executed instead.

DATA
: Scroll down to display one page (display window depth) less one line of data.
The last line in the current display area becomes the first line of the scrolled display.

HALF
: Scroll down half a page of data.
The line that is half way down the page of data in the current display area becomes the first record of the scrolled display.

MAX
: Scroll down to display the last page of data.
Where more than one page of data exists, the last displayable line becomes the last line of the scrolled display.
Otherwise, the first line of data becomes the first line of the scrolled display.

PAGE
: Scroll down to display the next whole page of data.
The line following the last line of the current display area becomes the first line of the scrolled display.

n_lines
: Scroll down a specified number of lines.
The line that is *n_lines* below the current line becomes the first line of the scrolled display.

DRAGBORDERMINUS

Syntax:

```
>>--- DRAGBORDERMINUS -----------------------------------------------------><
```

Description:

Window resizing command, DRAGBORDERMINUS is intended to be assigned to PFKeys at the "Border" level (see CLI command KEYS).

DRAGBORDERMINUS moves the window's border closer to the top left corner of the 3270 display area.

When the cursor is positioned on a horizontal border, the border is dragged one row upwards; when on a vertical border, the border is dragged one column to the left; and when on a corner, the border is dragged both one row upwards and one column to the left.

The border will not be positioned outside the display area in which it is defined. i.e. MDI child window borders cannot be dragged outside its parent's client area and all other windows cannot be dragged outside the 3270 display area.

Similarly, SELCOPY/i will not allow the window borders to be dragged so that the window is smaller than 3 rows x 8 columns.

By default, DRAGBORDERMINUS is assigned to Border PFKeys F7 and F10.

DRAGBORDERPLUS

Syntax:

```
>>--- DRAGBORDERPLUS ------------------------------------------------------><
```

Description:

Window resizing command, DRAGBORDERPLUS is intended to be assigned to PFKeys at the "Border" level (see CLI command KEYS).

DRAGBORDERPLUS moves the window's border away from the top left corner of the 3270 display area.

When the cursor is positioned on a horizontal border, the border is dragged one row downwards; when on a vertical border, the border is dragged one column to the right; and when on a corner, the border is dragged both one row downwards and one column to the right.

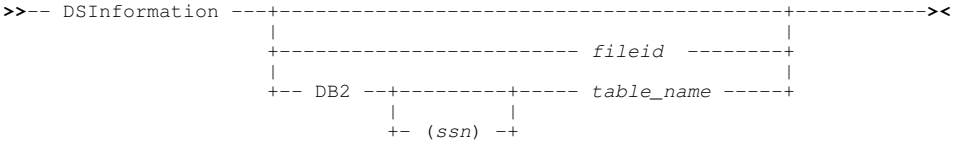
The border will not be positioned outside the display area in which it is defined. i.e. MDI child window borders cannot be dragged outside its parent's client area and all other windows cannot be dragged outside the 3270 display area.

Similarly, SELCOPY/i will not allow the window borders to be dragged so that the window is smaller than 3 rows x 8 columns.

By default, DRAGBORDERPLUS is assigned to Border PFKeys F8 and F11.

DSINFORMATION

Syntax:



Description:

Display the appropriate Data Sets Information panel for the specified *fileid* or display formatted report output for a DB2 *table_name*.

fileid may be the DSN of a sequential, VSAM, GDG, PDS or PDSE library (with or without a member name) or the name of a GDG Base catalog entry. *table_name* may be the name of an existing DB2 table or view in the DB2 server with sub-system name *ssn*.

The Data set information panel or DB2 table report may also be opened by entering the "I" prefix command against an entry in a file or DB2 table list.

If DSINFORMATION is executed with no parameters, the *fileid* or *table_name* of the file/DB2 table in the current text or data edit/browse view is used.

Parameters:

fileid
: Identifies the file object for which information is to be displayed. Error ZZSI027E will be returned if the *fileid* data set does not exist.

ssn
: Identifies the DB2 server sub-system name in which *table_name* is located. The SELCOPY/i DB2 plan (CBLPLAN0) must be bound to this DB2 sub-system for successful operation.

  Default sub-system name is that set by the DB2 Primary Option Menu.

table_name
: Identifies the DB2 table or view for which information is to be displayed. Error ZZSD340E will be returned if the *table_name* does not exist in the specified DB2 server.

  table_name may be specified with 1, 2 or 3 qualifiers representing *name*, *schema.name* or *location.schema.name* respectively. Default location is the local DB2 server and the default schema is the value assigned to special register CURRENT SCHEMA (initially set to the user's SQLID).

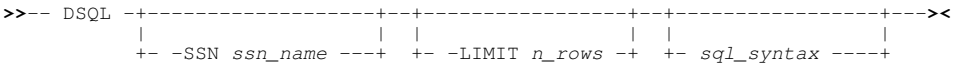
Examples:

`DSINFO OEM.SELC320.CBL13295.CSI`
Display information for the SELCOPY Product Suite SMP/E CSI data set.

`DSINFO DB2(CBLA) SYSIBM.SYSTABLES`
Display information for DB2 SYSTABLES catalog table in sub-system CBLA.

DSQL

Syntax:



Description:

Use the DSQL command to open the Execute SQL Statements (ZZS2XSQL) panel.

If the user has already opened a DB2 Primary Option Menu in the current SELCOPY/i session and so a DB2 connection already exists for the specified DB2 subsystem, then no new connection is made. Otherwise, a connection is made to this DB2 subsystem when the panel is opened.

Note: Not implemented for CMS or VSE.

Parameters:

`-SSN ssn_name`
The DB2 subsystem in which the specified SQL statement will be executed.

Default is the last DB2 subsystem to which the user connected in this or previous SELCOPY/i sessions. If no previous DB2 connection has been made using SELCOPY/i, the default is the value of the SELCOPY/i INI option, DB2.SSN, or else the subsystem name specified in the DB2SubSys field of the CBLNAME load module.

`-LIMIT n_rows`
Limit the number of rows to be displayed in the display area of the Execute SQL Statements panel following a SELECT transaction. Once the limit threshold has been reached, no further attempt is made to retrieve selected rows of data.

The *n_rows* value is placed in the "**Row Limit>**" field of the Execute SQL panel and the default limit is 0 (no limit).

sql_syntax
Valid SQL syntax to be executed when the Execute SQL panel is opened.

The *sql_syntax* string is placed in the "Statement>" field of the Execute SQL panel and the default is the last SQL statement executed from this panel. If the default *sql_syntax* string is used, its execution is delayed.

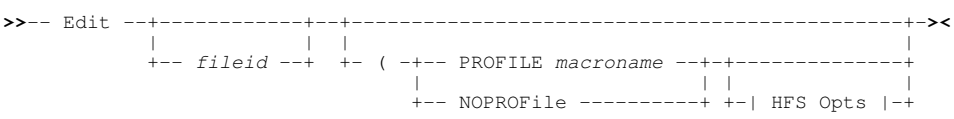
Examples:

`DSQL -SSN DB8G -LIMIT 200 SELECT * FROM DSN810.EMP`
Display the first 200 entries in the table DSN810.EMP of DB2 subsystem DB8G.

`DSQL -SSN CBLA GRANT EXECUTE ON PLAN CBLPLAN1 TO USER002`
Execute an SQL GRANT statement in subsystem CBLA.

EDIT

Syntax:



Description:

Use the EDIT command to open a CBLe text edit window view to edit a file (read/write).

If the CBLe text editor main window has been stopped, EDIT will start the CBLe main window before opening the edit view of the requested file.

If the file has a VSE LIBR LOCK applied or an exclusive MVS SPFEDIT ENQ to a different job id, then the **Enqueue Failed** pop-up window is displayed containing message ZZSE015E prompting the user to edit the file in read-only mode.

Read-only edit may be invoked directly using the VIEW command thus avoiding the Enqueue Failed window.

Figure 315. Enqueue Failed Window.

Parameters:

fileid
The fileid of the file to be edited.

For MVS, *fileid* may be any of the following:

- ◊ The DSN of a physical sequential data set.
- ◊ The DSN of a VSAM (KSDS, ESDS, RRDS, VRDS or LDS) data set.
- ◊ The library DSN and parenthesised member name of a PDS or PDSE library member.
- ◊ The library DSN, parenthesised member name and absolute or relative number of a PDSE library member generation as described under z/OS PDSE Library Member Generations. (PDSE version 2 with MAXGENS only.)
- ◊ The name of a member to be edited from the same PDS or PDSE library as the member displayed in the current text edit window view.
- ◊ The name of an HFS/ZFS file system fileid.

For VSE, *fileid* is the member name of a LIBR library in lib.sublib.mn.mt format.

For CMS, *fileid* is a CMS fileid.

Note that if **fileid** is not specified, then the SELCOPY/i INI variable System.CmdTEXT is used. If System.CmdTEXT has not been set, then no action is taken.

PROFILE *macroname*
The REXX edit macro to be executed as the profile when editing the file.

This macro overrides use of the default profile macro defined by the SELCOPY/i INI option Edit.DefProfile=macroname and/or the CBLe command SET DEFPROFILE (default PROFILE.)

The macro name must exist in a library within the CBLe macro path.

The PROFILE option only affects the profile for the file currently being added to the ring, and does not affect the profile to be used when additional files are added to the edit ring later in your edit session.

NOPROFILE
Suppresses use of a profile macro when editing the file.

The NOPROFILE option only affects the file currently being added to the ring, and does not affect the profile to be used when additional files are added to the ring later in your edit session.

HFS Opts
See CBLe CLI command EDIT for supported HFS parameters.

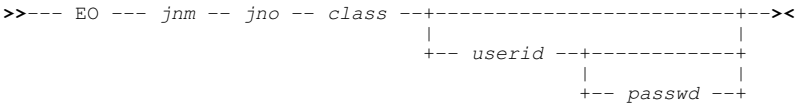
Examples:

```
EDIT     NBJ.DATA.SET    (PROFILE  PROFTEMP
```
Edit NBJ.DATA.SET using macro PROFTEMP as a profile.

```
EDIT     NBJ.JCL(CBLINS01)    (NOPROF
```
Edit PDS member NBJ.JCL(CBLINS01) without a profile macro.

EO

Syntax:



Description:

Use the EO command to Edit (read only) an Output listing from the system's output queues. A new CBLe text editor window is opened if SELCOPY/i INI variable Edit.Instance=Multiple or if no CBLe window is already open.

A temporary fileid is used to edit the file. For VSE, the fileid is **SYSLST.class.jnm.jno**.

The SELCOPY/i command, EO, may be used in conjunction with the CBLe command, SUBMIT. A job may be submitted to batch from within CBLe and the output retrieved via EO.

In VSE, Basic Security Manager (BSM) does not impose security on the VSE POWER queues. Therefore, in order to impose access restrictions on LST queue output when SELCOPY/i INI variable System.VSESMLogon=Yes, the following restrictions apply:

1. If an entry is **not** password protected, then it may only be edited if the TO or FROM attributes match the user's userid.
2. If an entry is password protected, then it may be edited by any user so long as the password is supplied.

If SELCOPY/i INI variable System.VSESMLogon=No (i.e. no Security Manager is active), then EO is only successful if the entry is password protected and the password is specified as a parameter to EO.

Note: Not yet implemented for MVS.

Parameters:

jnm
The required Job Name.

jno
The required Job Number.
Note that, when a job is submitted using the CBLe SUBMIT command, the job number is displayed in the job submitted confirmation message.

class
The required List Class.

userid
The userid of the user that owns the job. For VSE, this must be the userid on either the TO or FROM LST queue attributes.
Default is the current user's userid.

passwd
The password to be used when editing a password protected queue entry.
If the entry is password protected and no password is specified, then EO will fail.

Examples:

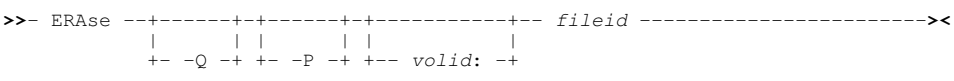
`eo LIBRDEL 1551 S`
View list output for job LIBRDEL having job number 1551 and belonging to list class S. For successful operation, the job must have TO or FROM LST queue attributes equal to the userid of the current user and SELCOPY/i INI variable System.VSESMLogon=Yes.

`eo CICSICCF 201 A SYSA SECRET`
View list output for job CICSICCF having job number 201, a TO or FROM attribute of SYSA and belonging to list class A. The queue entry is password protected with password "SECRET".

ERASE

Syntax:

Erase an MVS data set or HFS file:



Erase PDS(E) member(s) or PDSE member generations:

```
            |      | |      | |           |           |                    |
            +- -Q -+ +- -P -+ +-- volid: -+           +- mbrmask.genmask -+
```

Erase a CMS file on an accessed minidisk:

```
>>- ERAse ---------------------------------- fileid ------------------------><
```

Erase a VSE sequential or VSAM file:

```
>>- ERAse --------------------+-----------+-- fileid ------------------------><
                              |           |
                              +-- volid: -+
                              |           |
                              +- catdsn: -+
```

Description:

Erase (delete) a single sequential DASD dataset, VSAM dataset or HFS(ZFS) file. Alternatively, erase one or more PDS/PDSE library member or PDSE member generation.

To successfully erase a file object, the user must have sufficient access authority for the file and no exclusive ENQ or LOCK should already exist for the file.

For VSE, sequential files may only be erased if the CBL software product **CBLVCAT** is licensed. SELCOPYi uses CBLVCAT's DEL operation to perform the erase.

Parameters:

volid
: For z/OS uncataloged data sets and z/VSE sequential disk files, this is the volume serial number of the DASD volume on which the file resides.

-Q
: Execute the erase quietly so that no message confirming deletion is produced.

-P
: Prompt the user for erase confirmation. This is the default if multiple library members are specified e.g. member mask includes wildcards

catdsn
: For z/VSE VSAM files only, this is the complete fileid of the VSAM catalog to which the VSAM managed file belongs.

fileid
: The complete fileid (e.g. dataset name or HFS/ZFS file path) of the file object to be erased.

  For z/OS, specification of a leading "." (dot/period) or "/" (slash) identifies **fileid** as being an absolute or relative HFS path name. Erasing an HFS path name performs a USS UNLINK operation for the individual HFS path name and so alternate path names to the same data are unaffected.

  If *fileid* is a defined ALIAS for a non-VSAM data set, the ALIAS will be deleted, **not** the related data set.

libname
: The complete dataset name of a PDS/PDSE library from which one or more members or member generations are to be erased.

mbrmask
: Following a *libname* PDS/PDSE library specification, *mbrmask* identifies the member name of one or more members to be erased from that library. Multiple member name masks may be specified to provide alternate member selection masks. e.g.

```
ERASE  DEV.OEM.CBL202.CBLI.HELP.HTML(S*AN%  WIN*, *R)
```

genmask
: Applicable only to PDSE version 2 libraries allocated a MAXGENS value, *genmask* may be specified following a *mbrmask* to identify a related member generation mask.

  Erase will only occur for member generations that match both the member name mask and its related generation mask. e.g.

```
ERASE  NBJ.PDSE2G10.JCLLIB(ZZS*.<=-6   SQ%%%%%A.<0)
```

Examples:

`ERASE  TEST.EXEC.A`
: Erase CMS file "TEST EXEC A".

`ERASE  CBL.CBLI.TEST.FILE`

Erase z/OS cataloged data set CBL.CBLI.TEST.FILE.

```
ERASE  CBL.CBLI.TESTLIB(EXAMPLE1)
```
Erase z/OS PDS member CBL.CBLI.TESTLIB(EXAMPLE1).

```
ERASE  CBL.SELCOPYI.INIT.JCLLIB(INS*.<-1)
```
Erase z/OS PDSE V2 library member generations below relative generation value -1 whose member names begin with "INS".

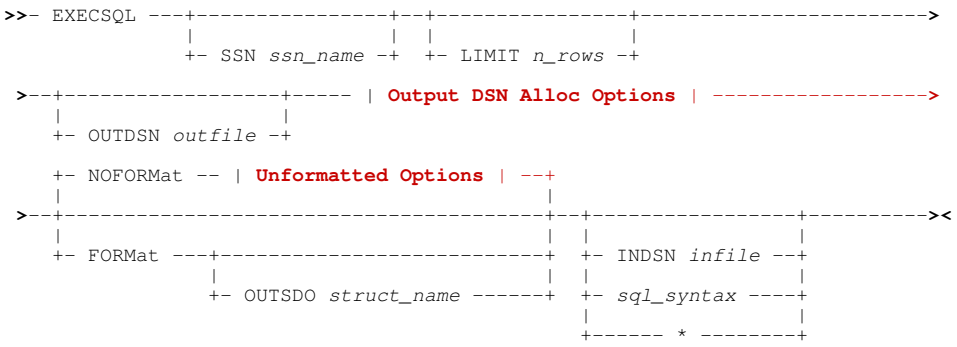
```
ERASE  OEM001:CBL.CBLI.TEST.FILE
```
Erase z/OS uncataloged data set CBL.CBLI.TEST.FILE from DASD volume OEM001.

```
ERASE  SYSWK1:CBL.SELCOPY.NAM
```
Erase z/VSE sequential file CBL.SELCOPY.NAM on SYSWK1. (CBLVCAT must be licensed.)

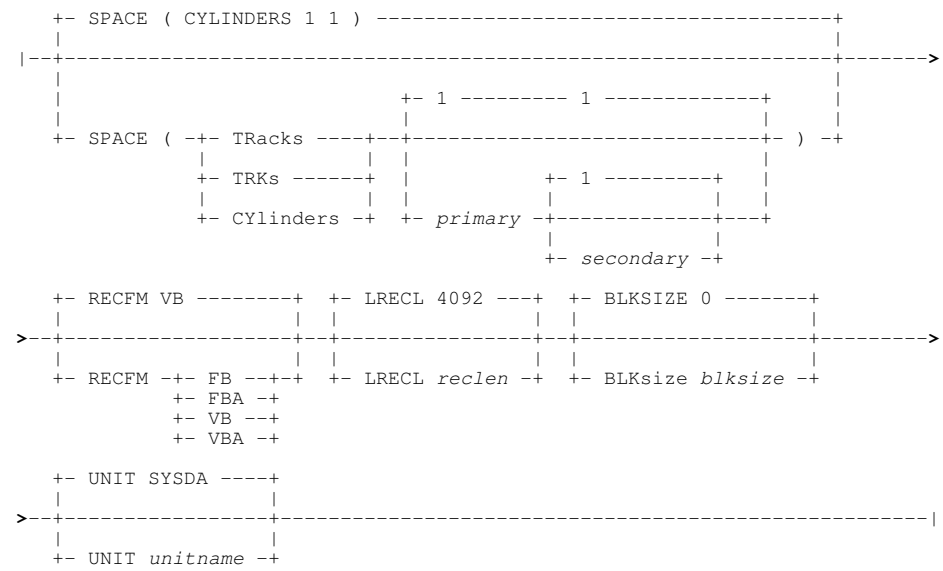
```
ERASE  VSESP.USER.CATALOG:CBL.TEST.KSDS
```
Erase z/VSE VSAM managed data set CBL.TEST.KSDS cataloged in the VSAM catalog, VSESP.USER.CATALOG.

EXECSQL

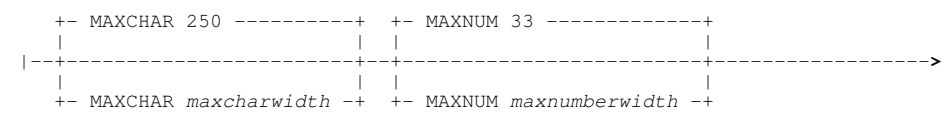
Syntax:



Output DSN Alloc Options:



Unformatted Options:



Description:

The EXECSQL primary command is the command line interface to the SELCOPYi SQL Execution facility. If EXECSQL is executed with no parameters, the ExecSQL utility panel is opened.

EXECSQL has all the features of the IBM SPUFI utility but with additional support for the following:

- SQL source input file and report output file may be a sequential data set or a PDS or PDSE library member and is not restricted by **RECFM/LRECL.** (SPUFI input must be RECFM=F LRECL=80).
- SQL sourced from an input file is not required. SQL may be provided via a text edit or data edit view of temporary data, or as parameters to the EXECSQL command.
- The report output file may be an as yet unallocated sequential data set. EXECSQL will allocate and catalog the new data set using values that may be specified by the EXECSQL command syntax.
- The report output may optionally be formatted and displayed in a data edit window view using a generated SDO structure.
- EXECSQL may be executed in batch using program SDEAMAIN.

If EXECSQL is executed with no parameters, SQL is sourced from text in the focus text or data edit window.

When run interactively, the output report file is browsed automatically on completion.

Parameters:

SSN *ssn_name*
Name of the DB2 sub-system (server) to which the SQL will be directed. SELCOPYi will perform a CONNECT to this sub-system before executing the SQL syntax.

Default is the SSN value assigned to the SELCOPYi INI option DB2.SSN. Note that the value assigned to this option may be updated via the DB2 Settings (=0.7) or DB2 Primary Option Menu (=12) panels.

LIMIT *n_rows*
Limit the number of rows fetched by an SQL query (SELECT) statement.

This limit is imposed by SELCOPY/i on all SELECT statements as rows are fetched. If a *fetch-first-clause* is specified on the SELECT, then a number of rows will be fetched which is the lesser of the limit value and the *fetch-first-clause* value.

Note that, if all selected rows are fetched then SQLCODE 100 is returned indicating that a FETCH statement was executed when the cursor was positioned after the last row of the result table. If the limit threshold is reached before this occurs, then the execution of SELECT will end with SQLCODE 0.

OUTDSN *outfile*
Specifies *outfile*, the name of the report output file. This file may be a new or existing sequential data set, a PDS or PDSE library member or, for unformatted output only, an HFS file path. Alternatively, it may be a DD name which is allocated to an existing output file.

If *outfile* is the DSN of a new, as yet unallocated sequential dataset, allocation parameters specified by **Output DSN Alloc Options** will be used to allocate and catalog the data set.

If OUTDSN is not specified, *outfile* will be the default name specified for Output Dataset in the Execute SQL Settings (=0.7.2) panel. If this value is blank, the generated output file will be a new sequential data set with DSN *prefix*.ZZS2XSQL.D*yyyyddd*.T*hhmmssu*.LST where *prefix* is the defined user DSN prefix.

Output DSN Alloc Options
Options used to allocate a new sequential data set output report file (*outfile*). If *outfile* already exists, these parameters are ignored.

Allocation values apply to both formatted and unformatted output report data sets.

SPACE(TRACKS|TRKS|CYLINDERS *primary secondary*)
The data set primary and secondary SPACE allocation values.

Default *secondary* allocation value is 1, default *primary* allocation value is 1 and default allocation unit is CYLINDERS.

RECFM FB|FBA|VB|VBA
The data set record format. Default is variable blocked (VB).

`LRECL reclen`
: The data set maximum record length. Default is 4092.

`BLKSIZE blksize`
: The data set block size. Default is 0 implying SMS system determined blocksize.

`UNIT unitname`
: The data set UNIT device type. Default is the esoteric group of devices, SYSDA.

`FORMAT | NOFORMAT`
: Specifies whether the output report is to contain formatted data (FORMAT), for display in the SDE data editor with associated SDO structure, or is unformatted plain text (NOFORMAT).

Default is NOFORMAT.

Unformatted Options
: Applicable only to unformatted output, these options control the display of result table row data fetched by an SQL query SELECT statement.

`MAXCHAR maxcharwidth`
: Specifies the maximum width of any character data type value in the selected columns. Default is 250.

`MAXNUM maxnumberwidth`
: Specifies the maximum width of any numeric data type value in the selected columns. Default is 33.

`OUTSDO struct_name`
: Applicable only to formatted output, specifies *struct_name* the name of the generated SDO structure. This may be a new or existing data set or library member, or a DD name which is allocated to one of these types of files.

If OUTSDO is not specified, *struct_name* will be the default name specified for Output Structure in the Execute SQL Settings (=0.7.2) panel. If this value is blank, the generated output file will be a new sequential data set with DSN *prefix*.ZZS2XSQL.D*yyyyddd*.T*hhmmssu*.SDO where *prefix* is the defined user DSN prefix.

`INDSN infile`
: Specifies *infile*, the name of the SQL source input file containing the text of one or more SQL statements. Note that the ';' (semi-colon) character is used to separate SQL statements and standard sequence numbers are ignored.

For compatiblity with **SPUFI,** any lines starting with **"--"** are treated as comment.

In addition any text enclosed by **'/*'** and **'*/'** characters is also ignored, allowing comments to span several consecutive lines.

The input file may be an existing sequential data set, library member or DD name which has been allocated to one of these files. Note that, unlike SPUFI input which must be RECFM=F LRECL=80, an EXEXCSQL input file is not restricted by **RECFM** or **LRECL**.

INDSN *infile*, *sql_syntax* and * (asterisk) are mutually exclusive parameters.

`sql_syntax`
: One or more SQL statements (*sql_syntax*) may be specified in-line following the last EXECSQL keyword parameter.

INDSN *infile*, *sql_syntax* and * (asterisk) are mutually exclusive parameters.

`*`
: Asterisk (*) indicates that SQL source is provided via the focus text edit or data edit/browse view. If INDSN and *sql_syntax* are omitted but other parameters are specified, then * is default.

INDSN *infile*, *sql_syntax* and * (asterisk) are mutually exclusive parameters.

Examples:

`execsql outdsn /u/cbl/nbj/lst/execsql_lst`
: Execute SQL statements in the focus text edit view and write unformatted report output to an HFS file.

`execsql outdsn NBJ.EXECSQL.LST(D2013316) indsn NBJ.SQL(D2013316)`
: Execute SQL statements in a PDS member and write the unformatted report output to a member of another PDS library.

`execsql indsn NBJ.SOURCE.SQL00233 format`
: Execute SQL statements in a sequential data set and write the formatted report output and structure to the default data sets.

FAV

Syntax:

```
>>---- FAV ------------------------------------------------------------------><
```

Description:

The FAV command may be used to open a Favourites Datasets/Commands window to easily access commonly used files and commands.

The dialog window will be opened with fields populated with parameters entered by the user during the last invocation of the window.

Parameters:

FAV has no parameters.

FCOPY

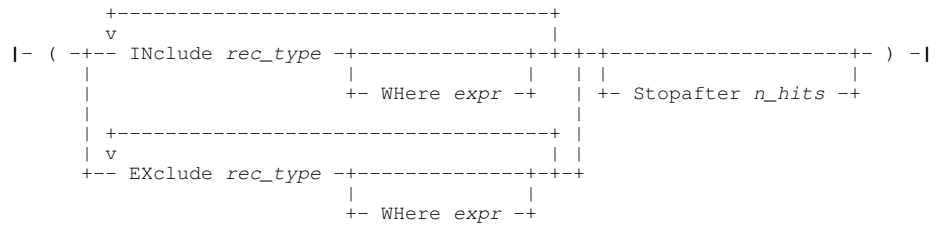
Syntax:

```
>>-- FCopy -----------+--------------------+---------------------------------><
                      |                    |
                      +--| CLI Options |--+
```

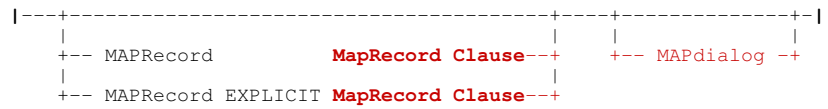
CLI Options:

```
 |-- from_fileid -+-------------------------------------+------------------>
                  |                                     |
                  |         +- SDO ---+                 |
                  |         |         |                 |
                  +- USING -+---------+-- in_struct ---+
                            +- COBOL -+
                            +- PL1 ---+
                            +- ADAta -+

 >-- to_fileid ---+-----------------------------------------------------+-->
                  |                                                     |
                  |         +- SDO ---+                                 |
                  |         |         |                                 |
                  +- USING -+---------+-- out_struct --+---------------+
                            +- COBOL -+                |               |
                            +- PL1 ---+                +- MAP Options -+
                            +- ADAta -+

 >--+-------------------------+-+----------------------+-+-----------------+->
    |                           | |                      | |                 |
    +- STARTKEY -- start_key -+ +- STOPAFT -+- n_recs -+ +- FILL -+- char -+
    |                           | |          |           |         |
    +- STARTRBA -- start_rba -+ +--FOR -----+           +- PAD --+
    |                         |                       + STRIP -+
    +- STARTREC +- start_rec -+
    |           |
    +- FROM ----+

 >--+----------+--+---------+--+----------+--+---------+--+-------+---------->
    |          |  |         |  |          |  |         |  |       |
    +- APPend -+  +- JCL ---+  +- REPlace +  +- Quiet -+  +- NEW -+
                  |         |
                  +- BATch -+

    +- MEMBERDLM --- 0 ----+
    |                      |
 >--+----------------------+---+--------+------------------------------------>
    |                      |   |        |
    +- MEMBERDLM -+- 0 --+-+   +- MOVE -+
                  +- 1 --+
                  +- 2 --+

 >-+-----------------------------------+--------------------------------------|
   |                                   |
   +-- FILTer --+--- filter_fileid ---+
                |                     |
                +--| Filter Clause |--+
```

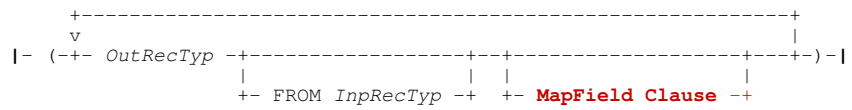
Filter Clause:



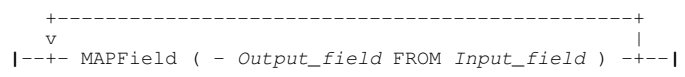
MAP Options:



MAPRecord Clause:



MAPField Clause:



Description:

The FCOPY command invokes the File Copy utility to copy a file or library members in the SELCOPY/i foreground, generate JCL to perform the copy in batch or, if executed with no parameters other than a source fileid, open the File Copy Panel.

FCOPY supports any combination of HFS files; cataloged sequential, GDG or VSAM data sets; cataloged PDS/PDSE libraries, library members and library member generations whose names match the *from_fileid* mask. Output *to_fileid* must be a single data file of any of these input types other than a PDSE library member generation.

The source and target file may be of different data set organisation and geometry (RECFM, LRECL, BLKSIZE) with truncation or padding of records being performed as required. e.g. Multiple members of a PDS/PDSE library may be copied to a sequential data set. VSAM KSDS records may be copied to a sequential data set or PDS/PDSE library member or HFS file etc.

Library copy is performed if the source file is a PDS/PDSE library, specified with or without a member mask, and the target file is a PDS/PDSE library with no member name specified. Note that a target PDS/PDSE library DSN with no member name is valid only for library copy.

Where possible, the File Copy utility invokes IEBCOPY to perform library copy. For file copy or where library copy performs copy functions not supported by IEBCOPY (e.g. libraries are of different geometry or the parameter STARTREC or FOR has been specified), then the SELCOPY/i File Search/Update/Copy/Remap Utility (FSU) is invoked.

Parameters:

from_fileid
: The full fielid of the source file or library to be copied.

  For PDS/PDSE libraries, multiple blank and/or comma separated may be member masks may be specified in parentheses "( )" following the library DSN. Each member mask may contain wildcard characters, "*" (representing zero or more characters) and/or "%" (representing exactly one character.) If a library DSN is specified without a member mask, then a member mask of (*) is assumed. i.e. all members.

  For a PDSE version 2 library supporting member generations (a MAXGENS value > 0), then each member mask may include a generation mask in order to select only specific generations of members whose name matches the member mask. Note that member generations are copied in ascending order of generation number, thus maintaining the generation hierarchy in the target library. If the target library does not support PDSE member generations, then the base generation from the input library will prevail.

A PDS/PDSE library may be specified as an ampersand (&) prefixed DDname which is allocated to one or more (concatenated) library DSNs (e.g. &SYSEXEC(MYREXX)). This means that the first occurrence of member MYREXX found within the path of concatenated libraries will be used as the source of the copy operation.

`USING SDO|COBOL|PL1|ADATA in_struct`

Specifies *in_struct*, the name of an SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file to be used to map record data fields in *from_fileid* for use in a SELCOPY/i FSU Formatted File Remap operation. Therefore, specification of an input structure *in_struct* is redundant if no output structure *out_struct* is specified for *to_fileid*.

All input records are treated as comprising a number of data fields of pre-determined lengths and of various data types, each field being identified via a field name.

If a COBOL copybook, PL1 include file or an ADATA file generated from a COBOL or PL1 compilation is specified, then this file will be used to generate a temporary SDO before proceeding with record formatting. This is an overhead which may be overcome by generating a non-temporary SELCOPY/i SDO file and referencing that instead. Note that a non-temporary SDO may be generated from the COBOL/PL1/ADATA file using the SDE command, CREATE STRUCTURE.

When *from_fileid* is read, each input record is assigned a record type (RTO), as defined in the specified or generated SDO, and the field definitions defined by that RTO are used to map the data within the record. SDE determines the record type to be assigned to each record based on any USE WHEN conditions saved in the SDO and the individual record's length. See *"Record Type Assignment"* in the *"SELCOPY/i Structured Data Editor (SDE)"* publication.

Data within the input record fields is remapped as described below under USING *out_struct* .

SDO, COBOL, PL1 or ADATA identifies the source format of the input structure file.
Default is SDO.

SDO	A SELCOPY/i Structured Data Object. This is the format required by SELCOPY/i to format structured data.
COBOL	A COBOL copy book. SELCOPY/i will use the COBOL compiler to compile the file in order to generate a temporary SDO.
PL1	A PL1 include file. SELCOPY/i will use the PL1 compiler to compile the file in order to generate a temporary SDO.
ADATA	An ADATA file created by a previous COBOL or PL1 compilation of a copybook/include file. SELCOPY/i will use the ADATA file to generate a temporary SDO.

to_fileid

The full fileid of the target file or library.

If *from_fileid* is a library (with or without member/generation masks) and *to_fileid* is not a library DSN, then records from all selected members will be copied to the single target data set. See parameter MEMBERDLM for optional member data delimitation.

Note that specification of a target library member mask for *to_fileid* is invalid. FCOPY does not support rename of multiple source members on copy to a target library (library copy).

If option NEW is not specified (for library copy only) and the *to_fileid* dataset does not already exist, then, provided FCOPY is not being executed in batch (via SDEAMAIN), the Allocate NonVSAM dialog window will be opened automatically.

`USING SDO|COBOL|PL1|ADATA out_struct`

Specifies *out_struct*, the name of an SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file to be used to map record data fields in *to_fileid* for use in a SELCOPY/i FSU Formatted File Remap operation. Therefore, specification of an output structure *out_struct* is redundant if no inpput structure *in_struct* is specified for *from_fileid*.

During the remap process, the following will occur:

1. Input records of record types not defined in the output structure are copied without field remap.
2. Output structure record types not defined in the input structure are redundant and so are ignored.
3. Record data in input fields are copied to output fields of the same name belonging to record types of the same name by default. To "match-up" record-types and fields with non-identical names use the **MAP Options** (**MAPDIALOG, MAPRECORD** and **MAPFIELD**) described below
4. The input field data will be reformatted to the data type of the output field and will be moved to the output field's position within the record map.
5. Any input fields that are not matched to an output field will not be included in the output record.
6. Any output fields that are not matced to an input field are initialised to their default values.

See the input structure USING *in_struct* field for description of output USING field sub-parameters and implementation of a structure on record data.

`STARTKEY start_key`

If *from_fileid* is a VSAM KSDS, VRDS file or PATH, STARTKEY may be used to specify a full or partial key *start_key* used to identify the first source record to be copied. All records occurring before *start_key* are bypassed.

start_key may be specified as a character or hex string using the standard notations (e.g. abc, 'abc', C'abc' or X'818283'). Note that upper casing of *start_key* will occur if specified as a character string without the "C" (or "c") prefix.

The record selected by *start_key* will be the first record with key field data which is greater than or equal to *start_key*.

`STARTRBA start_rba`
If *from_fileid* is a VSAM ESDS, STARTRBA may be used to specify a relative byte address *start_rba* used to identify the first source record to be copied. All records occurring before *start_rba* are bypassed.

start_rba may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**.

The record selected by *start_rba* will be the first record with a relative byte address which is greater than or equal to *start_rba*.

`STARTREC start_rec`
`FROM`
STARTREC (or FROM) specifies the record number *start_rec* of the first source record to be copied from the *from_fileid* file or library member(s). All records occurring before *start_rec* are bypassed.

start_rec may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**.

If STARTREC is not specified, *start_rec* defaults to 1.

`STOPAFT n_recs`
`FOR`
STOPAFT (or FOR) specifies the maximum number of records *n_recs* to be copied from the *from_fileid* file or library member(s)

If STOPAFT is not specified, *n_recs* is unlimited.

`FILL|PAD|STRIP char`
Indicates that:

1. When copying fixed length records to variable length, contiguous trailing characters at the end of each record that match the specified character *char* are to be stripped.
   Default is not to strip trailing characters.

2. When copying variable length records to fixed length, records are to be padded with the specified character *char*.
   Default is to pad with the blank character (X'40').

The strip/pad character *char* may be specified in character, hexadecimal or binary string notation of length 1 byte. (e.g. 'A', C'a', X'40', B'11110001'.)

`APPEND`
Applicable to file copy or remap only where *to_fileid* is not a library member, APPEND specifies that records written to the target file are to be appended to records that already exist in the target file.

Beware that, if the DSORG of the target file is a reuseable VSAM data set (IDCAMS DEFINE REUSE), then selecting NO will overwrite **all** existing records. An attempt to overwrite an existing record will fail if the VSAM data set is defined with NOREUSE.

Selecting YES will append records to the end of the target file unless it is a VSAM KSDS file. Copy to a VSAM KSDS file will write records to their correct key positions within the data set. If a record is not in key sequence or contains a duplicate key, then that record will not be copied and the copy operation continues at the next source file record.

If this option is not selected, then the existing records will be overwritten.

`MEMBERDLM n`
Applicable only when copying members from a PDS/PDSE library to a single output file e.g. an HFS, VSAM or sequential dataset. This option causes a delimiter record to be written before the data belonging to each member copied.

The value of *n* may be 0, 1 or 2 and indicates the format of the delimiter record to be inserted as described in the table below.

Option	Delimiter Record Format
0	No delimiter record (Default).
1	"./ ADD NAME=<member>"
2	"*>>>>>> DSN=lib.name(member) <<<<<<*"

`MOVE`
Applicable only when copying members from a PDS/PDSE library to another PDS/PDSE library or single output file e.g. an HFS file, library member, VSAM or sequential dataset. This option causes a source library members to be deleted upon successful copy to the target library or file.

`JCL`
`BATCH`
Do not execute the copy immediately but instead generate JCL in a temporary edit view that executes program SDEAMAIN to run the FCOPY command in batch.

`REPLACE`
Applicable to library copy or remap only, indicates that members that exist in *to_fileid* will be replaced by input members of the same name.

If this option is not selected, then existing members will not be replaced.

QUIET

For file copy and remap, no output report is generated so QUIET is redundant.

For library copy or remap, on successful completion of a copy of one or more library members, QUIET suppresses display of the IEBCOPY execution report or the FCOPY PDS Copy Statistics list window. In this case, only FCOPY messages will be displayed.
Default is to display any IEBCOPY report or FCOPY statistics.

NEW

Applicable to library copy or remap only, indicates that, if the *to_fileid* library is not yet allocated, then it should be automatically allocated using the same DCB geometry and SPACE attributes as the *from_fileid* library.

If this option is not selected and FCOPY is not being executed in batch (via SDEAMAIN), then the Allocate NonVSAM dialog window will be opened automatically.

FILTER *filter_fileid* | **Filter Clause**

FILTER specifies additional record filtering criteria to be applied to records that have been selected using the FROM, STARTKEY, STARTRBA and/or FOR parameters, otherwise it applies to all records of the input files.

FILTER parameters are specified via a filter clause which may be supplied as part of the EDIT command or referenced via *filter_fileid*, a separate sequential data set, PDS/PDSE member or HFS file. *filter_fileid* must contain the keyword FILTER followed by a valid filter clause.

Filter Clause

A filter clause must be specified in "( )" (parentheses) and may contain comment data enclosed by "/*" and "*/". If filter clause is specified via *filter_fileid*, then comment data may also occur before and after the filter clause.

The following options are supported by the filter clause.

INCLUDE *record_type*

Include only input records that are assigned the specified record type *record_type*. This parameter may be specified repeatedly to include a number of record types or to perform alternative WHERE *expr* filters for the same record type. If INCLUDE is specified, then all record types that are not referenced by an INCLUDE parameter will be excluded by default.

record_type "Record" (with field name "UnMapped") may be used to perform a filter on the unformatted record data whether or not a structure (USING *struct_name*) has been specified. In this way, a filter may test **all** records regardless of their assigned record type.

INCLUDE and EXCLUDE parameters are mutually exclusive.

EXCLUDE *record_type*

Exclude only input records that are assigned the specified record type *record_type*. This parameter may be specified repeatedly to exclude a number of record types or to perform alternative WHERE *expr* filters for the same record type. If EXCLUDE is specified, then all record types that are not referenced by an EXCLUDE parameter will be included by default.

record_type "Record" (with field name "UnMapped") may be used to perform a filter on the unformatted record data whether or not a structure (USING *struct_name*) has been specified. In this way, a filter may test **all** records regardless of their assigned record type.

INCLUDE and EXCLUDE parameters are mutually exclusive.

WHERE *expr*

WHERE applies further filter conditions to input records assigned to the record type specified by the last INCLUDE *record_type* or EXCLUDE *record_type* parameter processed.

expr is a valid SDE expression which supports function calls, *record_type* field names and references, sub-expressions, arithmetic, relational and logical operators. The result of the WHERE expression must be numeric and is treated as being Boolean in nature with a zero value indicating a "false" condition and any non-zero value indicating a "true" condition.

The WHERE expression is applied to each record assigned the record type *record_type* and, if the result is "true", the record is selected for include or exclude as indicated by the prevailing INCLUDE or EXCLUDE filter. If multiple INCLUDE/EXCLUDE *record_type* WHERE expressions exist for the same record type, then a logical OR is implied for all the expressions relating to that record type.

STOPAFTER *n_hits*

When the number of records selected by the INCLUDE or EXCLUDE filter reaches the value specified by STOPAFTER *n_hits*, then no further filter testing occurs.

If an INCLUDE filter, then all remaining untested records are excluded. If an EXCLUDE filter, then all remaining untested records are included.

MAPDIALOG

MAPDIALOG (MAP) causes a dialog to be displayed allowing the user to interactively "match-up" input to output record-types, and fields within those record-types.

Record-type and Field names that are identical in both the input and output structure are automatically matched. See the **EXPLICIT** option for the **MAPRECORD** parameter which may be used to prevent this action.

`MAPRECORD/MAPFIELD`

Use the MAPRECORD (MAPR) clause to match up one or more input record-types to their corresponding output record-type when reformatting record data during the copy procedure.

It's only necessary to explicitly define the match if the input record-type name is not identical to the output record-type name.

Each record-type match up clause may be immediately followed by a bracketed list of MAPFIELD clauses to match up field names within that particular record-type.

Again, it's only necessary to explicitly define the match if the input field name is not identical to the output field name.

To prevent automatic match up of identical record-type and field names, just add the **EXPLICIT** keyword immediately following **MAPRECORD**.

e.g.

```
MapRecord explicit
 (              ZARTIST              from ARTIST  /* Record-type */
   ( MapField( ZRT                  from RT                        )
     MapField( ZARTIST              from ARTIST.ARTIST             )
   )
              ZALBUM                from ALBUM   /* Record-type */
   ( MapField( ZRT                  from RT                        )
     MapField( ZALBUM               from ALBUM.ALBUM               )
   )
              ZTRACK                from TRACK   /* Record-type */
   ( MapField( ZRT                  from RT                        )
     MapField( ZRELEASE-DATE.YYYY   from RELEASE_DATE.RELEASE_YYYY )
     MapField( ZRELEASE-DATE.MM     from RELEASE_DATE.RELEASE_MM   )
     MapField( ZRELEASE-DATE.DD     from RELEASE_DATE.RELEASE_DD   )
     MapField( ZPERSISTENT-ID       from PERSISTENT_ID             )
     MapField( ZNAME                from NAME                      )
     MapField( ZFILE-SIZE           from FILE_SIZE                 )
   )
 )
```

Examples:

`FCOPY CBL.SEV.X628263.REPORT`

Open the File Copy dialog window and populate the "From DSN" field with CBL.SEV.X628263.REPORT.

`FCOPY NBJ.JCL(ZZS* ZZI*) NBJ.COPY.JCL NEW`

Create a copy of a JCL library containing only members with names starting "ZZS" or "ZZI".

`FCOPY NBJ.JCL(ZZS* ZZI*) NBJ.COPY.JCL REPLACE`

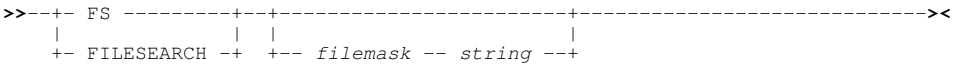
Following execution of the previous example, take up-to-date copies and add any additional members with names starting "ZZS" or "ZZI".

`FCOPY /u/smpnts/X0000012/S0005.CBL.PROD.SERVICE.SVCRNTS NBJ.JCL(SVCRNTS) JCL`

Create a batch job to copy an HFS file to a PDS library member.

FS

Syntax:



Description:

Use the FS command to open the File Search Window and optionally perform a file search.

The File Search window may also be opened via the Utilities menu of the SELCOPY/i main window menu bar.

Parameters:

filemask

The file mask of the MVS PDS and member, the VSE LIBR lib.sublib and member or the CMS file name type and mode to be searched.

This parameter is placed in the Dataset field of the File Search window.

string

The search string.

This parameter is placed in the Search String field of the File Search window.

Examples:

`FS`
Open the File Search window with both the Dataset and Search String fields left blank.

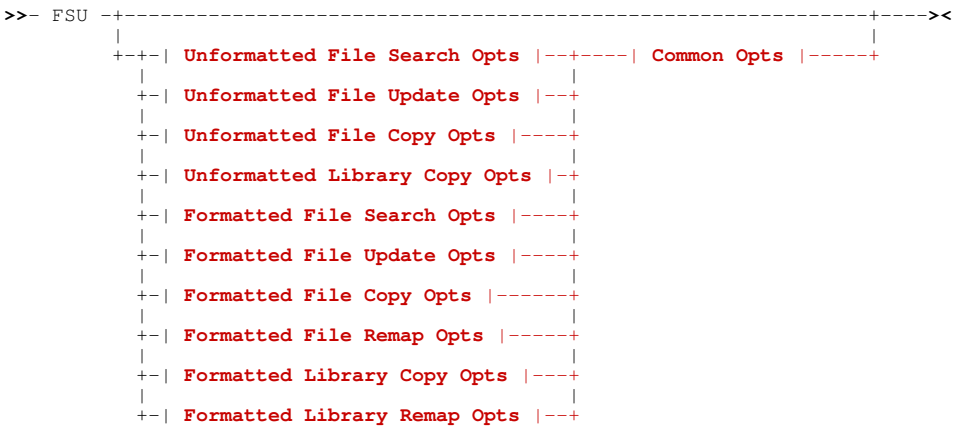
`FS 'CBL.Q6930.JCL(VV*)' 'PGM=CBLV'`
Search PDS members beginning 'VV' for string 'PGM=CBLV'.

`FS 'PRD2.CBL.*.htmlL' '<H1>'`
Search VSE PRD2.CBL library members of type 'HTML' for string '<H1>'.

`FS '* EXEC A' 'SELCOPY'`
Search CMS EXEC files on the A minidisk for string 'SELCOPY'.

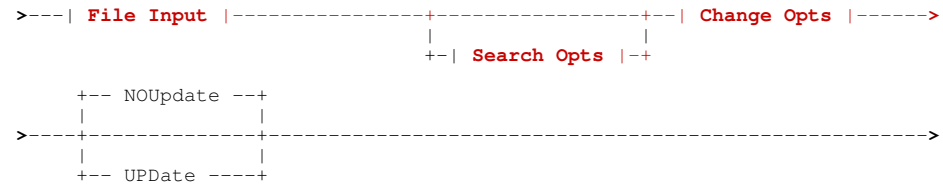
FSU

Syntax:

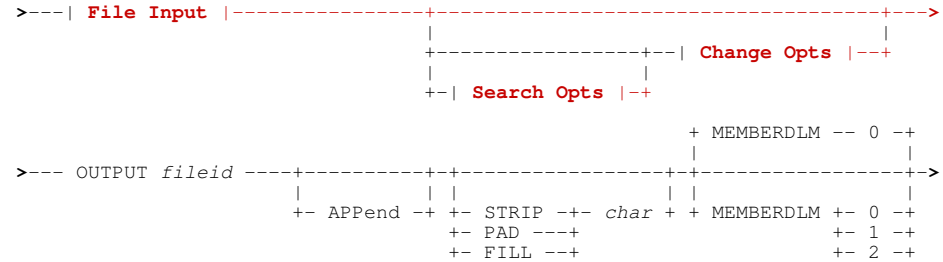


Unformatted File Search Opts:

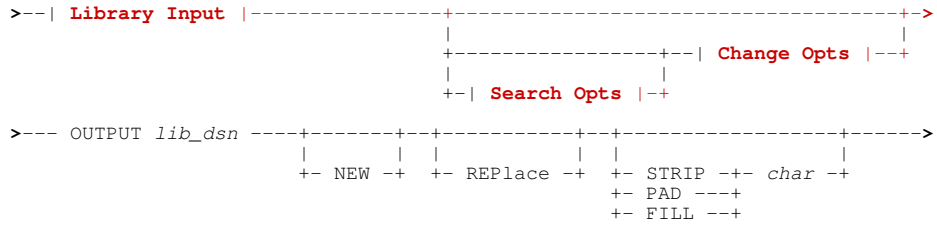
Unformatted File Update Opts:



Unformatted File Copy Opts:

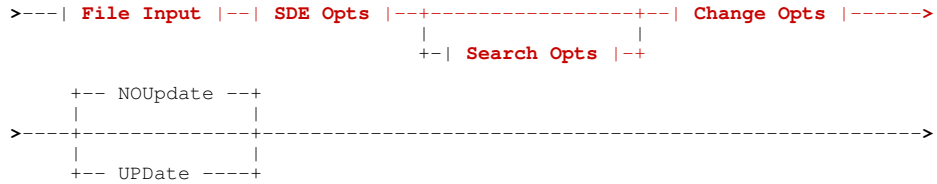


Unformatted Library Copy Opts:

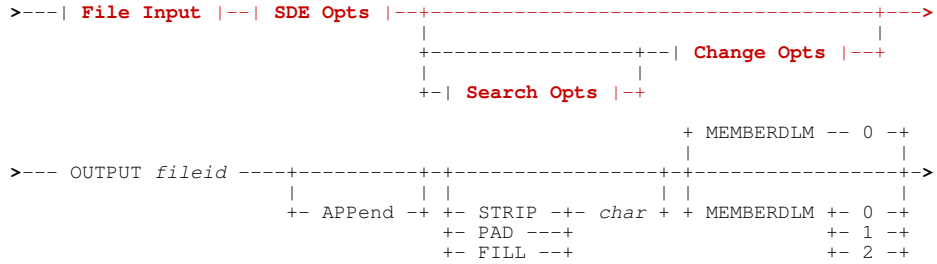


Formatted File Search Opts:

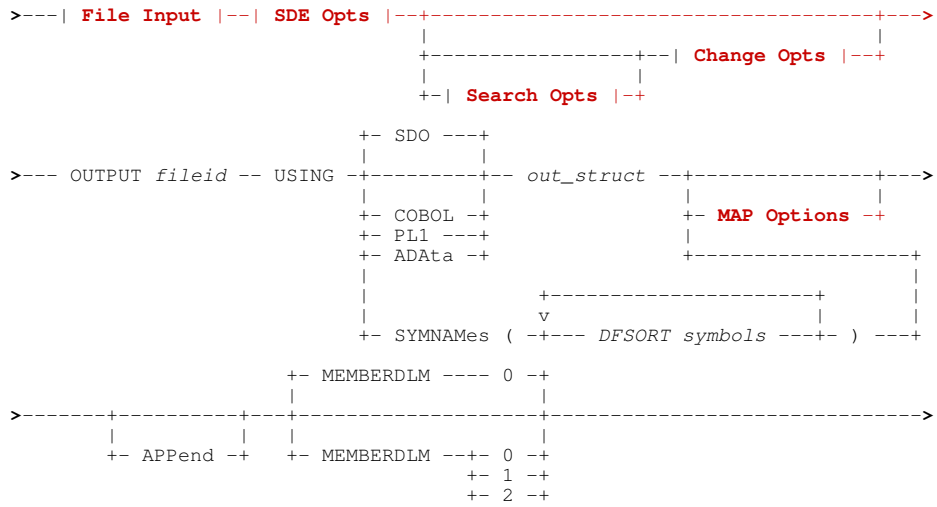
Formatted File Update Opts:



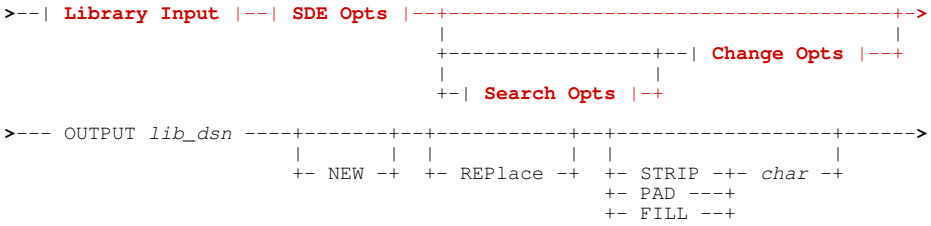
Formatted File Copy Opts:



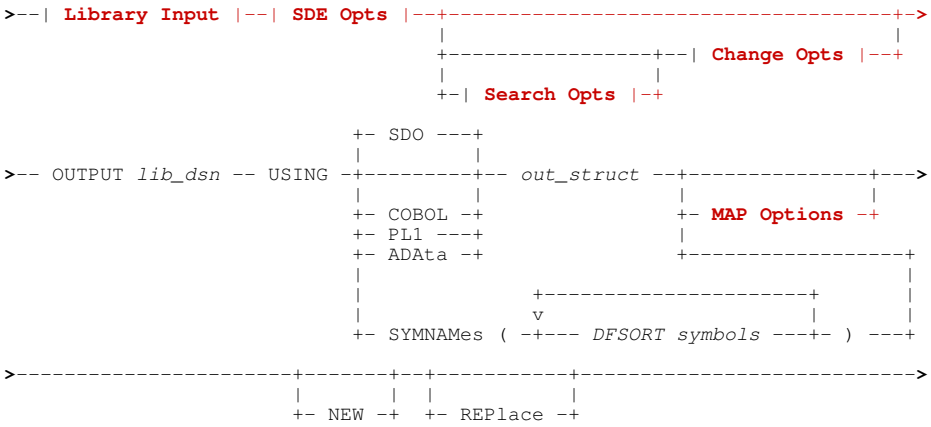
Formatted File Remap Opts:



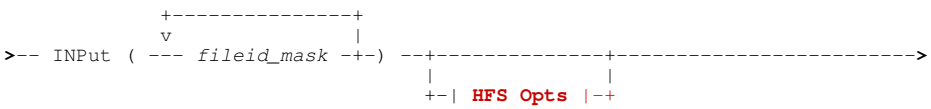
Formatted Library Copy Opts:



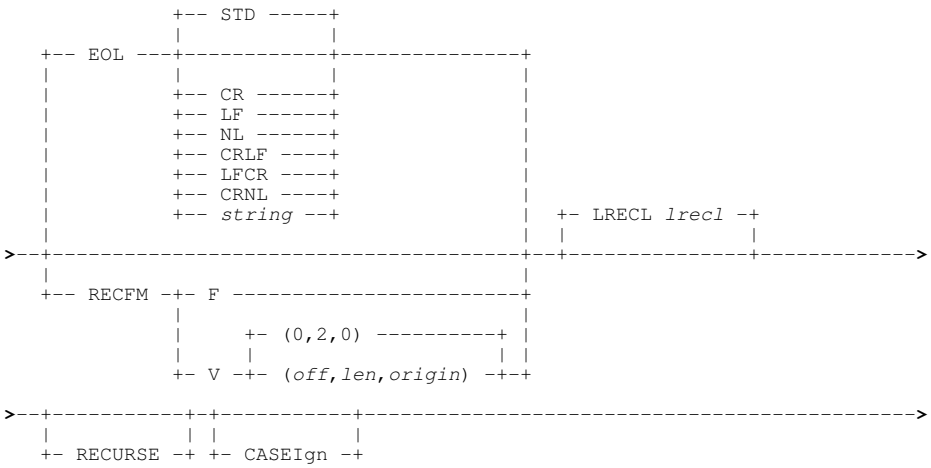
Formatted Library Remap Opts:



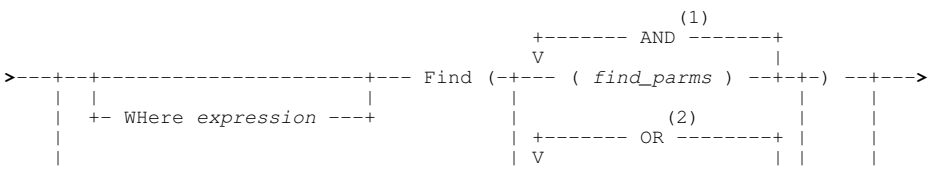
File Input:

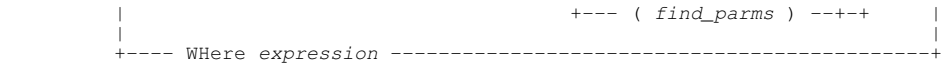


HFS Opts:

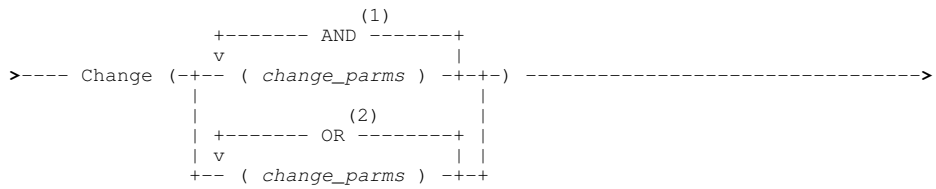


Search Opts:

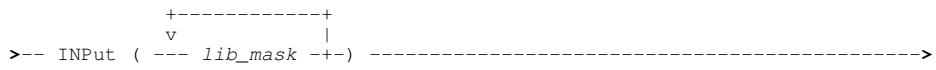




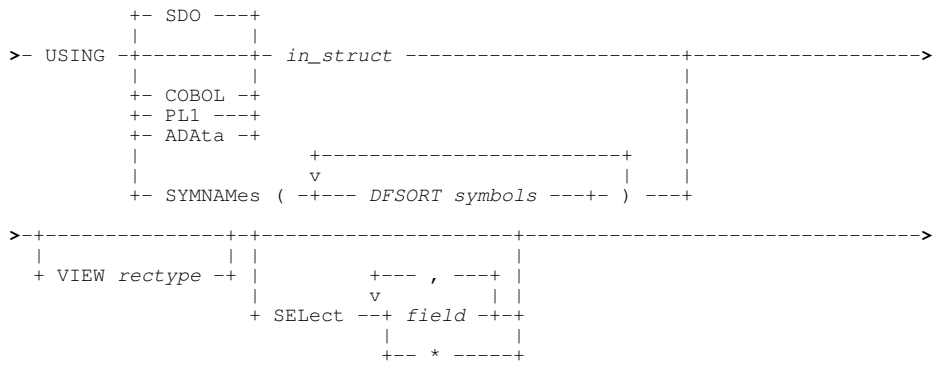
Change Opts:



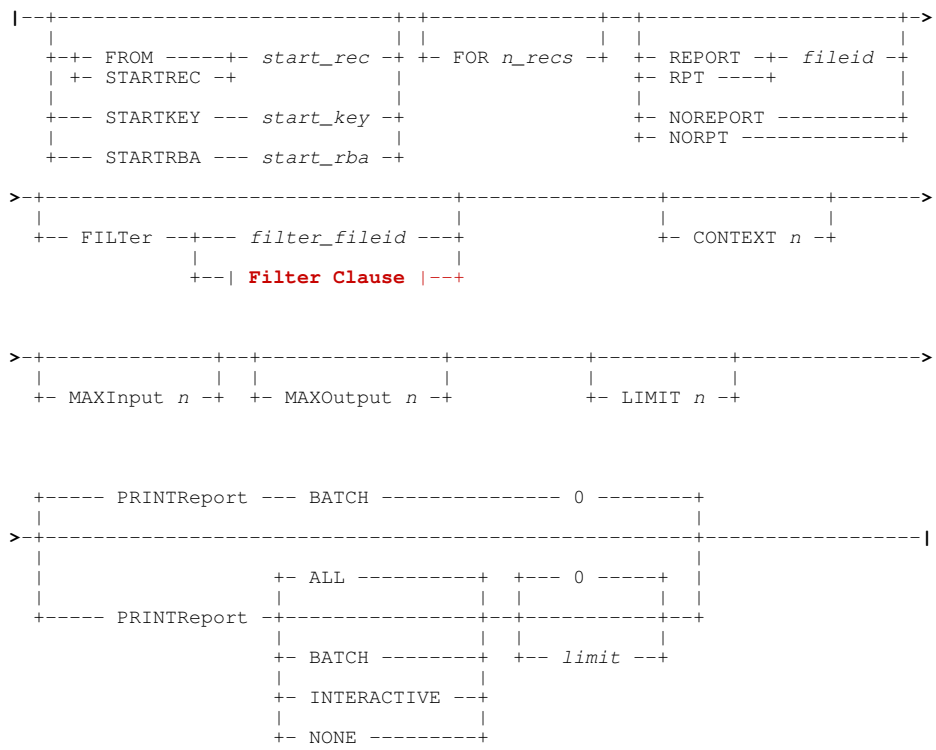
Library Input:



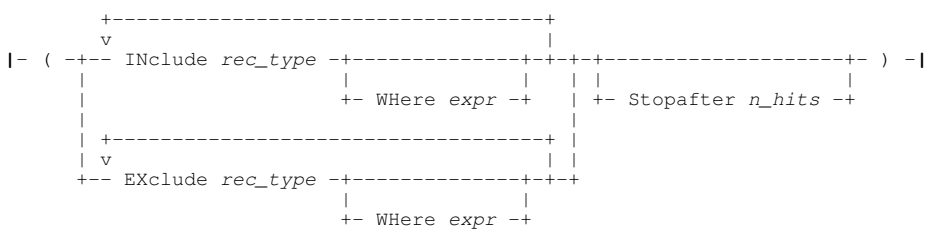
SDE Opts:



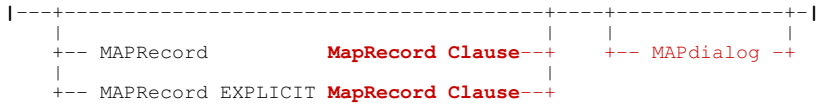
Common Opts:



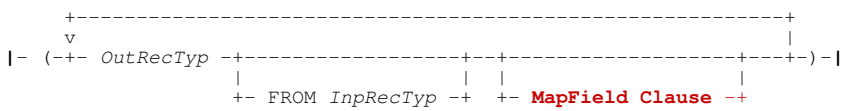
Filter Clause:



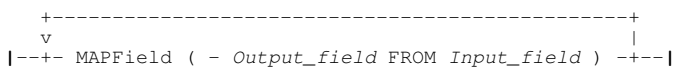
MAP Options:



MAPRecord Clause:



MAPField Clause:



Notes:

1. **AND** logical operator literal or its equivalent symbol: "&" (ampersand).
2. **OR** logical operator literal or its equivalent symbols: "¦" (broken bar) or "|" (vertical line).

Description:

The FSU command is the command line interface to the File Search/Update/Copy/Remap Utility.

If FSU is executed with no parameters, the File Search/Update/Copy/Remap Panel is opened.

Parameters specified on the FSU command govern the type of operation performed by the utility. The following table illustrates the operation performed when FSU parameters, denoted by "*" (asterisk), are provided.

	FIND	WHERE	CHANGE	USING	OUTPUT	OUTPUT USING
Unformatted Search	*	(1)	-	-	-	-
Unformatted Search	(1)	*	-	-	-	-
Unformatted Update	(1)	(1)	*	-	-	-
Unformatted Copy	(1)	(1)	(1)	-	*	-
Formatted Search	*	(1)	-	*	-	-
Formatted Search	(1)	*	-	*	-	-
Formatted Update	(1)	(1)	*	*	-	-
Formatted Copy	(1)	(1)	(1)	(2)	(3)	-
Formatted Remap	(1)	(1)	(1)	*	(3)	*

Notes:

1. At least one of FIND, WHERE, CHANGE or OUTPUT must be specified.

2. USING is only required if a formatted FIND, WHERE and/or CHANGE operation is to be applied to copied records.
3. If the fileid specified on OUTPUT is a PDS/PDSE library DSN with no member name, then **Library copy/remap** is performed so that input library members are copied to members of the same name in the output library. Otherwise, File copy/remap occurs where all input records are copied to a single output file.

Unless restricted by STARTREC/STARTKEY/STARTRBA and/or FOR parameters, **all** records of a file (sequential, VSAM, PDS/PDSE and HFS) whose DSN/fileid matches the fileid mask(s) provided by the INPUT parameter, are included in the operation. If a structure, provided by the USING parameter, is applied to the input records, then only those records assigned the default record type (RTO), specified by the VIEW parameter, are eligible for search and change operations.

Any section or sections of the FSU command stream may be commented out using REXX style comment delimiters. (i.e. enclose areas of the command stream text between "/*" and "*/".) This is particularly useful when FSU commands are entered in the user's HOME (CMX) command centre file, where the command may span a number of (continued) lines. e.g. To temporarily omit the CHANGE parameter, thus allowing the user to identify those records that would be selected for change...

```
<sd fsu                                                \
   INPUT (                                             \
           CBL*:JGE*.CBLI.SDE.SAMP.VAR(DATS*)          \
              %user%.CBLI.SDE.SAMP.VAR.DATS*.ESDS      \
         )                                             \
   WHERE (    CUST-ID > 5000                           \
         )                                             \
/*                                                     \
   CHANGE( (c'Aqua' c'AQUA'  (COMPANY)     )   AND     \
           (c'Jim'  c'James' (NAME,#9:#11) )           \
         )                                             \
      NOUPDATE                                         \
*/                                                     \
   USING      %user%.CBLI.SDO(CobSALES)                \
   VIEW       REC-CARD
```

The FSU command may be executed in the foreground or via SDEIN input to program SDEAMAIN for batch processing.

During foreground execution, the FSU output report is displayed and automatically updated in an SDE window view. A progress window is also displayed which allows the user to interrupt processing before completion using the Attention key. See File Search/Update/Copy/Remap Output.

Parameters:

`INPUT (`*`fileid_mask`*`< ...> | `*`lib_mask`*`< ...>)`

Specifies one or more file masks, *fileid_mask*, used to identify files to be searched, updated, copied or remapped. Multiple instances of *fileid_mask* must be specified with one or more intervening blanks.

All HFS files; cataloged sequential, GDG or VSAM data sets; cataloged PDS/PDSE libraries, library members and library member generations whose names match a *fileid_mask*, will be selected for processing.

If, however, *fileid_mask* includes a volume mask, then both cataloged and uncataloged data sets whose names match the *fileid_mask* **and** reside on the specified volume will be selected for processing.

lib_mask is a sub-category of *fileid_mask* and may be specified as a PDS/PDSE library DSN mask with or without accompanying member masks or member generation masks. Alternatively, *lib_mask* may be specified a concatenation of one or more PDS/PDSE libraries allocated to a DDname. A differentiation is made between *lib_mask* and *fileid_mask* when performing **Library Copy/Remap** where only input PDS/PDSE libraries are processed. Any other file type selected for input to Library Copy/Remap will not be copied.

A single *fileid_mask* may be in one of the following formats:

1. An ampersand (&) prefixed allocated DDNAME which represents one or more (concatenated) PDS/PDSE libraries that form a library directory path. (e.g. &SYSEXEC)

   This means that only the first occurrence along the directory path of each member that matches the specified **Member Mask>** will be processed.

   Members are processed in member name order.

2. A pre-allocated DDNAME (non-HFS) which may represent one or more (concatenated) data sets and/or libraries. (e.g. SYSEXEC)

3. An absolute or relative HFS path name.

   Wild card characters "%" (percent), representing a single characters, and "*" (asterisk), representing zero or more characters, are supported in the name portion of the HFS path. The name portion of the HFS path is the character string at the end of the path that follows the last "/" (slash) of the fileid, or is the entire path name if "/" is not specified.

   Beacuse FSU supports comment specification (text enclosed between "/*" and "*/"). Where the HFS file name wild card "*" (asterisk) is to be used following a directory separator "/" (slash), the HFS path must me enclosed in single quotes (apostrophes) or double quotes. e.g.

   ```
   '/u/ibmuser/tmp/*'    /* Search all files in this directory. */
   ```

4. A DSN Mask and optionally a volume Mask and/or multiple PDS/PDSE member masks in the following format:

```
{volmask:}data.set.name.mask{(membmask{,membmask, ...} )}
```

fileid_mask must **not** be enclosed in quotes (TSO prefix is not used.)

Wild card characters "%" (percent), "*" (asterisk) and "**" (double asterisk) are supported. (See File Search/Update/Copy/Remap Panel documentation for use of wild cards in the volume, DSN and member masks.) Similarly, one or more member masks may be specified between a single pair of "( )" (parentheses). Multiple PDS/PDSE member masks must be separated by a "," (commma) and/or one or more intervening blanks.
e.g. `DEV88%.CBL*.**(SELC*, *MAN, XM*J*)`

All sequential, VSAM and PDS/PDSE data sets that match a *fileid_mask* are selected for input. If one of these data sets is a PDS/PDSE library then all members of that library will be processed. In order to restrict the search to a single PDS/PDSE library and so exclude any non-PDS data set that matches *fileid_mask*, then a member mask should be specified.
e.g. `SYS7.OEM.CBL202.CBLI.CBLE(*)`

If the DSN Mask identifies a PDSE version 2 library supporting member generations (a MAXGENS value > 0), then each member mask may include a generation mask to select specific generations of members whose name matches the member mask.
e.g. `X12345.JCL(SSOPEN22.*, SSCLOSE*.>=-10)`

fileid_mask may be prefixed by a volume serial mask in order to restrict the search to only those cataloged and uncataloged data sets that match the specified *fileid_mask* **and** for which extents exist on the specified volume(s). The volume serial mask must be destinguishable from the rest of the fileid mask by an intervening ":" (colon) with no embedded blanks.
e.g. `CBLM04:SYS7.**.DZ3*.**`

Note that, if this criteria is satisfied, then **all** records of a **cataloged**, multi-volume data set will be searched. However, only extents that exist on the specified volumes will be searched if the multi-volume data set is **uncataloged**.

HFS Opts

Applicable to **all** HFS files that match the specified *fileid_mask*, the following options may be specified to determine how HFS data is processed by the utility.

For non-HFS files that match an INPUT *fileid_mask*, HFS options are ignored.

`EOL=STD|NL|CR|LF|CRLF|LFCR|CRNL|`*`string`*

Specify the EOLIN (input end-of-line) delimiter used to identify the end of each record for unformatted (non-RECFM F or V) HFS file input. EOLIN delimiters are not included in the edited record data or record length. EOL parameter elements are as follow:

STD	-	Any standard line-end.
NL	X'15'	New Line.
CR	X'0D'	Carriage Return.
LF	X'0A'	Line Feed.
string	-	A 2-byte user specified character or hex string.

STD is default so that the EDIT operation scans the input data for any of the standard EOL combinations (not *string*), stopping when one is found. This EOL combination is used as EOLIN for the file.

`RECFM F | V (`*`off`*`,`*`len`*`,`*`origin`*`)`

Specifies that the data is to be treated as containing Fixed or Variable length format records.

RECFM F indicates that all records are of a fixed length as defined by the LRECL argument.

RECFM V allows the user to specify the location of the record length fields within the data as follows:

off	Offset of the record length field from the start of the record.
len	Length of the record length field.
origin	The start of the record data at which the record length is applied.

Default is (0,2,0) which describes standard RECFM V organisation data sets.
The length field will be included as part of the input record data, so, if a CHANGE operation is specified, care must be taken not to corrupt the length field.

`LRECL `*`lrecl`*

Specifies the maximum record length of input HFS file records.

For RECFM F HFS files, *lrecl* is the fixed length of the records processed. If the HFS file size is not a multiple of *lrecl* value, then the last record of the file will be short.
Default *lrecl* for this types of file is 80.

For RECFM V and unformatted (EOL delimitted records) HFS files, if a record length exceeds *lrecl*, processing is stopped for that particular file.
Default *lrecl* for these types of files is 32752.

`RECURSE`

For an HFS *fileid_mask* containing wild card characters, recursively search files within all directories and sub-directories identified by the mask.
Default is not to search files in HFS sub-directories.

`CASEIGN`

Bypass case sensitivity for the **name** portion of all specified HFS path fileid masks. The name portion of the HFS path is the character string at the end of the path that follows the last "/" (slash) of the fileid, or is the entire path name if "/" is not specified.
Default is to respect the character case of the HFS file name.

SDE Opts

Specification of SDE parameters are required for, and applicable only to, **Formatted File or Library Search/Update/Copy/Remap** processing.

If SDE option parameters are not specified, **Unformatted File or Library Search/Update/Copy** processing is performed.

`USING SDO|COBOL|PL1|ADATA` *`in_struct`*

Specifies *in_struct*, the name of an SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file to be used to map input record data fields for use in a Formatted File Search, Formatted File Update, Formatted File Copy or Formatted File Remap operation.

All input records are treated as comprising a number of data fields of pre-determined lengths and of various data types. Each field within the record may be referenced independently (by field name or field reference number) allowing the user to be more descriminate when selecting records, and fields for WHERE, FIND and CHANGE operations.

If a COBOL copybook, PL1 include file or an ADATA file generated from a COBOL or PL1 compilation is specified, then this file will be used to generate a temporary SDO before proceeding with record formatting. This is an overhead which may be overcome by generating a non-temporary SELCOPY/i SDO file and referencing that instead. Note that a non-temporary SDO may be generated from the COBOL/PL1/ADATA file using the SDE command, CREATE STRUCTURE.

Each input record is assigned a record type (RTO) defined in the specified or generated SDO and the field definitions defined by that RTO are used to map the data within the record. SDE determines the record type to be assigned to each record based on any USE WHEN conditions saved in the SDO and the individual record's length. See *"Record Type Assignment"* in the *"SELCOPY/i Structured Data Editor (SDE)"* publication.

SDO, COBOL, PL1 or ADATA identifies the source format of the input structure file.
Default is SDO.

SDO	A SELCOPY/i Structured Data Object. This is the format required by SELCOPY/i to format structured data.
COBOL	A COBOL copy book. SELCOPY/i will use the COBOL compiler to compile the file in order to generate a temporary SDO.
PL1	A PL1 include file. SELCOPY/i will use the PL1 compiler to compile the file in order to generate a temporary SDO.
ADATA	An ADATA file created by a previous COBOL or PL1 compilation of a copybook/include file. SELCOPY/i will use the ADATA file to generate a temporary SDO.

`USING SYMNAMES ( DFSORT symbols )`

Specifies DFSORT SYMNAME symbol definitions that are to be used to format the input data records. The order in which symbol definitions are supplied dictate the order in which the fields will occur in the record type definition.

The symbol name definitions within the SYMNAMES parentheses may be supplied directly in-line and/or via input data sets/library members.

```
SYMNAMES( Card,06,04,CH  Dept,46,03,CH  Amount,49,06,PD  )
SYMNAMES( SYS1.MACLIB(EDGSMFSY)  SYS1.MACLIB(EDGSRCSY) )
SYMNAMES( CBL.DFSORT.SYM(CBLATRAC)  TCB,*,4,BI )
```

`VIEW` *`rectype`*

Applicable to **formatted** records only, *rectype* specifies the default record type against which all SELECT, WHERE, FIND and CHANGE operations are performed. Records not assigned this record type are not processed by these operations.

The named record type must match one defined within the specified SDO structure or within a temporary SDO structure generated from a COBOL, PL1, ADATA file referenced by USING.

Although not mandatory, it is recommended that VIEW *rectype* is included when performing formatted data processing so avoiding confusion over which records are to be processed. This is particularly true when performing CHANGE operations.
Default is the first record type defined in the structure (SDO).

`SELECT field<, ...><, *>`

Applicable to **formatted** records only, *field* identifies an individual field (by name or reference number) in records assigned the default record type, to be selected for FIND and/or CHANGE processing.

Multiple, comma separated *field* arguments may be specified, not only defining a list of fields, but also the order in which they are to be processed. "*" (asterisk) may be specified to represent all remaining fields in the default record type that have so far not been selected.

This is most useful where FIND and CHANGE operations are allowed to default to searching all fields. If CHANGE and/or FIND operations are specified to search specific fields that are not identified by SELECT, then processing stops and error ZZSD179E is returned.

Default is to search all fields in the record in their order of occurrence within the record type definition.

`WHERE expression`

Specifies an SDE expression to be applied by an SDE WHERE record filtering operation.

The WHERE expression defines search criteria for Unformatted File Search or Formatted File Search, or is used for record filtering before executing a FIND or CHANGE operation.

For **formatted** records, the WHERE operation is performed only on those input records that are assigned the default record type as specified by VIEW. Any field within the formatted record may be referenced regardless of whether it has been included by the SELECT.
e.g.

```
FSU INPUT( DEV.USER01.JCL(*) )
    WHERE( (#3 >= 22)  AND  ((EmpName = 'Smith') OR (Dept >> 'E1')) )
```

For **unformatted** records, the WHERE operation is performed on all records. The WHERE expression may only include reference to a single field (field reference #1, field name "Record") which evaluates to all data in the focus record. This field has a data type of CHAR and length equal to the file's maximum record length.
e.g.

```
FSU INPUT( DEV.USER01.JCL(*) )    WHERE( #1 >> '//'  AND  #1 << 'EXEC' )
```

If no CHANGE operation is specified, then a simple file search is performed so that the report identifies all records that satisfy the WHERE operation and any subsequently executed FIND operation. If a CHANGE operation is specified, only these records are eligible to be changed and only records that have been changed are identified in the output report.
e.g.

```
FSU INPUT( DEV.TEST.DATAX3. )  USING( DEV.TEST.SDO(SDDATAX3) )
 WHERE( JOBTITLE = 'MANAGER' OR SALARY > 38000 )
 CHANGE( ('Jo W Smith' c'JWS' (#20:#22)) OR ('J W Smith' c'JWS' (#20:#22)) )
```

`FIND (find_parms) | ((find_parms) op (find_parms) <op ...>)`

Specifies *find_parms* which corresponds to a search string and other supported parameters to be executed by an SDE FIND operation.

A FIND *find_parms* operation defines a search criterion for Unformatted File Search or Formatted File Search, or is used for record filtering before executing a CHANGE operation.

The FIND operation is performed on those input records that first satisfy any supplied WHERE expression.

Unlike SDE file edit, *find_parms* are applied at the record level, not at the file level. Therefore for FIND (but not CHANGE), parameters ALL/NEXT/FIRST/PREV/LAST have no real effect and so are redundant.

Furthermore, for **formatted** records, only records assigned a record type that matches the VIEW default record type are searched. The FIND operation may be further restricted to search only selected input fields as specified by the SELECT parameter. If *find_parms* includes field references and SELECT is specified, then all fields identified by *find_parms* must also be referenced by SELECT, otherwise processing stops and error ZZSD179E is returned.

For **unformatted** records, if no WHERE expression is specified, FIND will be performed on all input records.

If multiple *find_parms* combinations are specified, each *find_parms* group must be enclosed in "( )" (parentheses) with intervening operator *op*, (either logical AND or logical OR).

- ◊ If logical **AND** is specified then a record must satisfy **all** *find_parms* operations in order to be selected.
- ◊ If logical **OR** is specified then a record may satisfy **any** one of *find_parms* operations in order to be selected.
- ◊ A combination of both AND and OR logical operators is not supported and will cause an error.

e.g.

```
FSU INPUT( DEV.USER01.JCL(*) )
     FIND( (c'EXEC' WORD 1 20)  AND (c'IKJEFT01')  AND ('REGION') )

FSU INPUT( DEV.USER*.COBOL.COPYBOOK(*) )
     FIND( (c'REDEFINES' WORD)  OR ('OCCURS' WORD)  OR ('filler.' 12 80) )

FSU INPUT( DEV.TEST.DATAX3. )
    USING COBOL DEV.TEST.COB(SDDATAX3)
     FIND( (25 (#10:#18)) OR ('Ramsay' (EMP-NAME)) )
```

Where logical AND is used, second and subsequent FIND operations are completely independent of those previous to them i.e. are not subject to the position of the data found by the previous, successful FIND *find_parms* operation. If you require selection of records based on an ordered sequence of search strings then the pattern matching and wildcard power of a unix-style "Regular Expression" is recommended.

For formatted records, a numeric search string will be treated as a signed numeric value and an arithmetic compare will occur for numeric data fields. For non-numeric fields and unformatted records, all search strings are treated as character data and a logical string compare is performed.

`CHANGE (change_parms) | ((change_parms) <op (change_parms) <op ...>)`

Specifies *change_parms* which corresponds to a search and replace string and other supported parameters to be executed by an SDE CHANGE operation.

CHANGE *change_parms* will perform character string or numeric value substitution on record data and, where OUTPUT has **not** been specified, implies Unformatted File Update or Formatted File Update.

The CHANGE operation is performed on those input records that first satisfy any supplied (WHERE and/or FIND) search criteria.

Furthermore, for **formatted** records, only records assigned a record type that matches the VIEW default record type are processed by CHANGE. The CHANGE operation may be further restricted to change only data in selected input fields as specified by the SELECT parameter. If *change_parms* includes field references and SELECT is specified, then all fields identified by *change_parms* must also be referenced by SELECT, otherwise processing stops and error ZZSD179E is returned.

For **unformatted** records, if no FIND or WHERE parameter is specified, CHANGE will be performed on all input records.

Unlike SDE file edit, *change_parms* are applied at the record level, not at the file level. Therefore, if specified, *change_parms* parameter ALL changes all occurrences of the CHNAGE search string within a record, FIRST changes the first occurrence and LAST changes the last occurrence within a record.

If multiple *change_parms* combinations are specified, each *change_parms* group must be enclosed in "( )" (parentheses) with intervening operator *op*, (either logical AND or logical OR).

- ◊ If logical operator AND is used, CHANGE will execute all the *change_parms* operations.
- ◊ If logical operator OR is used, CHANGE will execute each *change_parms* operation in turn until one successfully changes the data, at which point, no further *change_parms* operation is attempted.
- ◊ A combination of both AND and OR logical operators is not supported and will cause an error.

e.g.

```
FSU INPUT(OEM.**)
   CHANGE( (c'DB8F' c'DB9G' PREFIX ALL) AND (c'CBLI' c'CBLI160' ALL) )

FSU INPUT(DEV.USER01.JCL(*) )
     FIND(c'EXEC' WORD 1 20)
   CHANGE(c'IEWL' c'BIND' WORD FIRST 16 80)

FSU INPUT(DEV.TEST.DATAX3)
    USING(DEV.SDO(DATAX3))
   CHANGE(('Jo Smith' c'JS' (#2:#5)) OR ('J W Smith' c'JS' (#2:#5)))
```

Where logical AND is used, a change made by a *change_parms* specification may itself be changed by a subsequent *change_parms* specification within the same execution of FSU.

Also, when AND is used, second and subsequent CHANGE are not subject to the position of the data changed by any previous, successful CHANGE *change_parms* operation. If you require selection and alteration of records based on an ordered sequence of search strings then the pattern matching and wildcard power of a unix-style "Regular Expression" is recommended.

For Formatted records, a numeric search string and replace string will be treated as a signed numeric values. An arithmetic compare will occur for the search string when applied to numeric data fields and the numeric replace string converted to a field's numeric data type as appropriate. For non-numeric fields and unformatted records, all search and replace strings are treated as character data and a logical string compare is performed.

Note that, for File Update only, records are re-written using update-in-place so the record length cannot be changed. Therefore, the CHANGE operation must not alter the length of an unexpanded/unformatted record, otherwise a change error will occur. This condition will be flagged against the record in the output report.

Where the length of a search string is different to that of the replace string, then the following occurs:

- ♦ If the length of search string is greater than the length of the replace string, then words to the right of the replaced string will be shifted left.

  However, if parameter TEXT is specified and more than one blank exists before a word to the right of the replaced string, then blanks are inserted to maintain that word's position in the record.

- ♦ If the length of the search string is less than the length of replace string, then words to the right of the replaced string will be shifted right. Note, however, that CHANGE will not increase the length of formatted data beyond its defined maximum field length.

If parameter TEXT is specified, multiple, consecutive blanks are absorbed to leave at least one blank between each word. Only if no blanks are eligible to be absorbed will text to the right of the replaced string be shifted right.

`NOUPDATE`
`UPDATE`

NOUPDATE and UPDATE are applicable to Unformatted File Update and Formatted File Update only.

NOUPDATE indicates that records that would be updated by the CHANGE operation are not to be written to disk so allowing the user to first review the output report and verify that the changes are correct before re-running the FSU command with UPDATE.

UPDATE indicates that records altered by the CHANGE operation are written to disk, so replacing the previous copy of the record.

Default is NOUPDATE.

Copy/Remap Opts

Copy/Remap parameters are required for, and applicable only to, **Copy** and/or **Remap** processing.

Specification of OUTPUT defines the utility processing to be Unformatted File or Library Copy or Formatted File or Library Copy. If an Output USING structure is also specified, then processing will be Formatted File or Library Remap.

If Copy/Remap parameters are not specified, File **Search** or **Update** processing is performed.

`OUTPUT fileid | lib_dsn`

Specifies *fileid* or *lib_dsn*, an output file or library to which **all** input records will be copied.

lib_dsn is a new or existing PDS/PDSE library **without** a member name specification. If specified, *lib_dsn* indicates that **Library Copy/Remap** will occur so that input library members are copied to *lib_dsn* with their member name unchanged. Non-library member input files are ignored.

Running in batch, *fileid* may be one of the following:

- An existing physical sequential (PS) data set.
- An existing VSAM data set.
- A new or existing member of an existing PDS/PDSE library.
- A new or existing HFS/ZFS file path.

Running interactive, if *fileid* does not already exist then the user will be prompted to create it.

If a CHANGE operation is specified and activated, then record data may be changed before it is written to the output file.

The format of the output file should be compatible with input record data. e.g.

- If an output KSDS data set is specified, input records must be in key sequence, as defined by the output file, and must not contain duplicate keys. If an input record does not satisfy these conditions, it will fail to copy.
- Records will be truncated if the input record length exceeds the maximum allowed by the output file.

`USING SDO|COBOL|PL1|ADATA out_struct`

Specifies *out_struct*, the name of an SDE structure (SDO), COBOL or PL1 copybook, COBOL or PL1 ADATA file to be used to map output record data fields for use in Formatted File or Library Remap.

Formatted File Remap processing only occurs when an output file and input and output structures are supplied. Therefore, an output structure is ignored if no input structure has been specified.

During the remap process, the following will occur:

1. Input records of record type not defined in the output structure are copied without field remap.
2. Output structure record types not defined in the input structure are redundant and so are ignored.
3. Record data in input fields are copied to output fields of the same name belonging to record types of the same name.
4. The input field data will be reformatted to the data type of the output field and will be moved to the output field's position within the record map.
5. Any input fields whose field names are not part of the output record structure, will not be included in the output record.
6. Any output fields whose field names are not part of the input record structure are initialised to their default values.

See the input structure USING field for description of output USING field sub-parameters and implementation of a structure on record data.

`USING SYMNAMES ( DFSORT symbols )`

Specifies DFSORT SYMNAME symbol definitions that are to be used to format the output data records. The order in which symbol definitions are supplied dictate the order in which the fields will occur in the record type definition.

The symbol name definitions within the SYMNAMES parentheses may be supplied directly in-line and/or via input data sets/library members.

```
SYMNAMES( Card,06,04,CH  Dept,46,03,CH  Amount,49,06,PD  )
SYMNAMES( SYS1.MACLIB(EDGSMFSY)  SYS1.MACLIB(EDGSRCSY) )
SYMNAMES( CBL.DFSORT.SYM(CBLATRAC)  TCB,*,4,BI )
```

APPEND

Applicable to File copy or remap only, specifies that output records are to be appended to existing data in the output file.

If this option is not selected, then the existing records will be overwritten.

MEMBERDLM *n*

Applicable only when copying or remapping members from a PDS/PDSE library to a single output file e.g. an HFS, VSAM or sequential dataset. This option causes a delimiter record to be written before the data belonging to each member copied.

The value of *n* may be 0, 1 or 2 and indicates the format of the delimiter record to be inserted as described in the table below.

Option	Delimiter Record Format
0	No delimiter record (Default).
1	"./ ADD NAME=<member>"
2	"*>>>>>> DSN=lib.name(member) <<<<<<*"

PAD | FILL | STRIP *char*

Applicable only to copy, indicates that:

1. When copying fixed length records to variable length, contiguous trailing characters at the end of each record that match the specified character *char* are to be stripped.
   Default is not to strip trailing characters.

2. When copying variable length records to fixed length, records are to be padded with the specified character *char*.
   Default is to pad with the blank character (X'40').

The strip/pad character *char* may be specified in character, hexadecimal or binary string notation of length 1 byte. (e.g. 'A', C'a', X'40', B'11110001'.)

NEW

Applicable to Library copy or remap only, indicates that, if *lib_dsn* is as yet unallocated, then the library should be automatically allocated using the same DCB geometry and SPACE attributes as the first input library identified by the INPUT *lib_mask*.

If this option is not selected, then the user will be prompted to create the library (unless running in batch using SDEAMAIN).

REPLACE

Applicable to Library copy or remap only, indicates that, members that exist in *lib_dsn* will be replaced by input members of the same name. Note that, if more than one input library contains a member of the same name, then both will be copied but the second member copied will replace the first.

If this option is not selected, then existing members will not be replaced.

STARTREC *start_rec*
FROM

STARTREC (or FROM) specifies the record number *start_rec* of the first record to be processed in all input files identified by INPUT *fileid_mask* or *lib_mask*. All records occurring before *start_rec* are bypassed.

start_rec may be specified as an integer numeric value **123** or as a hexadecimal numeric value **X'7B'**.

If STARTREC is not specified, *start_rec* defaults to 1.

STARTKEY *start_key*

For VSAM KSDS, VRDS files or PATHs only, STARTKEY specifies a full or partial key *start_key* used to identify the first record to be processed in all input files identified by INPUT *fileid_mask* or *lib_mask*. All records occurring before *start_key* are bypassed.

start_key may be specified as a character or hex string using the standard notations (e.g. abc, 'abc', C'abc' or X'818283'). Note that upper casing of *start_key* will occur if specified as a character string without the "C" (or "c") prefix.

The record selected by *start_key* will be the first record with key field data which is greater than or equal to *start_key*.

STARTRBA *start_rba*

For VSAM ESDS files only, STARTRBA specifies a relative byte address *start_rba* used to identify the first record to be processed in all input files identified by INPUT *fileid_mask* or *lib_mask*. All records occurring before *start_rba* are bypassed.

start_rba may be specified as a decimal integer or hexadecimal value.

The record selected by *start_rba* will be the first record with a relative byte address which is greater than or equal to *start_rba*.

`FOR n_recs`
FOR specifies the maximum number of records *n_recs* to be processed from each input file identified by INPUT *fileid_mask* or *lib_mask*.

If FOR is not specified, *n_recs* is unlimited.

`FILTER filter_fileid | Filter Clause`
FILTER specifies additional record filtering criteria to be applied to input records that have been selected using the FROM, STARTKEY, STARTRBA and/or FOR parameters, otherwise it applies to all records of the input files.

FILTER parameters are specified via a filter clause which may be supplied as part of the EDIT command or referenced via *filter_fileid*, a separate sequential data set, PDS/PDSE member or HFS file. *filter_fileid* must contain the keyword FILTER followed by a valid filter clause.

Filter Clause
A filter clause must be specified in "( )" (parentheses) and may contain comment data enclosed by "/*" and "*/". If filter clause is specified via *filter_fileid*, then comment data may also occur before and after the filter clause.

The following options are supported by the filter clause.

`INCLUDE record_type`
Include only input records that are assigned the specified record type *record_type*. This parameter may be specified repeatedly to include a number of record types or to perform alternative WHERE *expr* filters for the same record type. If INCLUDE is specified, then all record types that are not referenced by an INCLUDE parameter will be excluded by default.

record_type "Record" (with field name "UnMapped") may be used to perform a filter on the unformatted record data whether or not a structure (USING *struct_name*) has been specified. In this way, a filter may test **all** records regardless of their assigned record type.

INCLUDE and EXCLUDE parameters are mutually exclusive.

`EXCLUDE record_type`
Exclude only input records that are assigned the specified record type *record_type*. This parameter may be specified repeatedly to exclude a number of record types or to perform alternative WHERE *expr* filters for the same record type. If EXCLUDE is specified, then all record types that are not referenced by an EXCLUDE parameter will be included by default.

record_type "Record" (with field name "UnMapped") may be used to perform a filter on the unformatted record data whether or not a structure (USING *struct_name*) has been specified. In this way, a filter may test **all** records regardless of their assigned record type.

INCLUDE and EXCLUDE parameters are mutually exclusive.

`WHERE expr`
WHERE applies further filter conditions to input records assigned to the record type specified by the last INCLUDE *record_type* or EXCLUDE *record_type* parameter processed.

expr is a valid SDE expression which supports function calls, *record_type* field names and references, sub-expressions, arithmetic, relational and logical operators. The result of the WHERE expression must be numeric and is treated as being Boolean in nature with a zero value indicating a "false" condition and any non-zero value indicating a "true" condition.

The WHERE expression is applied to each record assigned the record type *record_type* and, if the result is "true", the record is selected for include or exclude as indicated by the prevailing INCLUDE or EXCLUDE filter. If multiple INCLUDE/EXCLUDE *record_type* WHERE expressions exist for the same record type, then a logical OR is implied for all the expressions relating to that record type.

`STOPAFTER n_hits`
When the number of records selected by the INCLUDE or EXCLUDE filter reaches the value specified by STOPAFTER *n_hits*, then no further filter testing occurs.

If an INCLUDE filter, then all remaining untested records are excluded. If an EXCLUDE filter, then all remaining untested records are included.

`MAXINPUT n`
The maximum number of records that will be read from all input files.

Compare this with the FOR parameter which limits the number of records processed per input file or library member.

FSU will terminate normally once this threshold has been reached.

`MAXOUTPUT n`
The maximum number of records that will be written to the output file.

Compare this with the STOPAFTER option for FILTER which limits the number of output records per input file or library member.

FSU will terminate normally once this threshold has been reached.

`LIMIT n`

The maximum number of FIND hits (records hit) to report per file, or library member.

Specify a value of **zero (0)** to report all hits.

Efficiency gains may be achieved by specifying **Limit=1** where it is required only to establish whether or not each member contains a particular string, since once the first hit is reported further processing for that member will be bypassed.

Note that *LIMIT n is ignored if the FSU operation is a file-copy or update-in-place. e.g. if copying a library and making in-flight changes of one or more strings, then members will be copied complete and strings will be updated (where present) in all records, in spite of LIMIT being specified.*

`CONTEXT n`

Includes in the report the specified number of input records immediately before and after each hit in order to provide context.

`MAPDIALOG`

MAPDIALOG (MAP) causes a dialog to be displayed allowing the user to interactively "match-up" input to output record-types, and fields within those record-types.

Record-type and Field names that are identical in both the input and output structure are automatically matched. See the **EXPLICIT** option for the **MAPRECORD** parameter which may be used to prevent this action.

`MAPRECORD/MAPFIELD`

Use the MAPRECORD (MAPR) clause to match up one or more input record-types to their corresponding output record-type when reformatting record data during the copy procedure.

It's only necessary to explicitly define the match if the input record-type name is not identical to the output record-type name.

Each record-type match up clause may be immediately followed by a bracketed list of MAPFIELD clauses to match up field names within that particular record-type.

Again, it's only necessary to explicitly define the match if the input field name is not identical to the output field name.

To prevent automatic match up of identical record-type and field names, just add the **EXPLICIT** keyword immediately following **MAPRECORD**.

e.g.

```
MapRecord explicit
 (              ZARTIST              from ARTIST  /* Record-type */
   ( MapField( ZRT                  from RT                        )
     MapField( ZARTIST              from ARTIST.ARTIST             )
   )
              ZALBUM                from ALBUM   /* Record-type */
   ( MapField( ZRT                  from RT                        )
     MapField( ZALBUM               from ALBUM.ALBUM               )
   )
              ZTRACK                from TRACK   /* Record-type */
   ( MapField( ZRT                  from RT                        )
     MapField( ZRELEASE-DATE.YYYY   from RELEASE_DATE.RELEASE_YYYY )
     MapField( ZRELEASE-DATE.MM     from RELEASE_DATE.RELEASE_MM   )
     MapField( ZRELEASE-DATE.DD     from RELEASE_DATE.RELEASE_DD   )
     MapField( ZPERSISTENT-ID       from PERSISTENT_ID             )
     MapField( ZNAME                from NAME                      )
     MapField( ZFILE-SIZE           from FILE_SIZE                 )
   )
 )
```

`REPORT fileid`
`RPT`

REPORT specifies that the FSU search/update/copy/remap report is to be written to the specified sequential data set or PDS/PDSE member *fileid*.

If *fileid* exists but is uncataloged, then include the required volser as part of the fileid specification in the format *volser:data.set.name*.

The report is a structured data file designed to be browsed (and optionally printed) using a SELCOPY/i structure definition object (SDO), which is also generated by FSU. The associated SDO fileid is constructed simply by adding '.SDO' to the DSN of the sequential or partition dataset name specified by *fileid*. The DSN is therefore restricted to 40 bytes in length. e.g. If *fileid* is ZX1234.SELCOPYI.FSU.REPORT(XYZ001), the allocated SDO is ZX1234.SELCOPYI.FSU.REPORT.SDO(XYZ001).

If the report file *fileid* and/or the SDO file do not already exist, then they will automatically be allocated by FSU relying on SMS ACS to select a suitable storage group of eligable DASD volumes. File *fileid* is allocated using DCB geometry RECFM=VB, LRECL=32756, BLKSIZE=0 and a space allocation of TRACKS(150,75). SDO is allocated using DCB geometry RECFM=VB, LRECL=16380, BLKSIZE=0 and a space allocation of TRACKS(2,2).

If REPORT and option NOREPORT are both **not** specified, FSU defaults to writing the report output and its accompanying SDO to temporary, in-storage files of DSN "*user.*FSU.Dyyyyddd.Thhmmss" and "*user.*FSU.Dyyyyddd.Thhmmss.SDO" respectively.

```
NOREPORT
NORPT
```

Indicates that report generation is to be suppressed. This is most useful for copy processing without CHANGE, whereby no FSU report records are generated for input data records.

```
PRINTREPORT   BATCH | INTERACTIVE | ALL | NONE   limit
```

If REPORT is used, PRINTREPORT controls whether or not the generated FSU structured report output is to be printed. Keyword parameters determine the environment(s) in which execution of the FSU command will generate printed output.

BATCH	Batch only using program SDEAMAIN. (Default)
INTERACTIVE	Interactive only (VTAM, TSO or ISPF)
ALL	Either batch or interactive environments.
NONE	Suppress print for all environments.

The optional *limit* parameter may be specified as a numeric integer literal indicating the maximum number of pages of print output allowed. A limit value of 0 (zero) indicates that there should be no restriction placed on the number of output pages. This is the default.

If print output is required, the following PRINT utility command is executed at end-of-job.

```
PRINT FILE TRUNC LIMIT <limit>
  INDSN(      <ReportFileid>
        USING <ReportFileid>.SDO
        INITCMD ("select zFileId,zRecNo,zRecord,*  from Hit"
                 "VFMT"
                 "PUSH SAVEOPTIONS"
                 "SAVEOPTIONS OFF"
                 "RECLEN OFF"
                 "REFERENCE OFF"
                 "PREFIX OFF"
                 "POP SAVEOPTIONS"
                )
      )
```

Examples:

Use of the FSU command may result in long command streams. Therefore, it is recommended that any FSU command should be entered as text in your HOME command centre (CMX) data set.

```
<sdata fsu input(XRVHC.**.PROCLIB(*) SYS1.PROCLIB(*))   find( DSN710 )
```

Report any member records within the specified PROCLIB libraries that contain the string "DSN710".

```
<sdata fsu input(XRVHC.**.PROCLIB(*) SYS1.PROCLIB(*)) change(DSN710 DSN810 ALL)
```

For member records within the specified PROCLIB libraries, report records that contain the string "DSN710" followed by the records' appearance after replacing all occurrences of "DSN710" to "DSN810". Members records are **not** updated.

```
<sdata fsu                                                                  \
   INPUT ( SAR22.TEST.FX*.** )                                              \
   USING ( SAR22.FX100.COBOL.COPYBK.SDO )                                   \
   VIEW  ( FX_Part_02 )                                                     \
   SELECT( Part_ID, Serial_No, Batch_No, Part_Description )                 \
   WHERE ( Batch_No > 730   AND  (Fault_Type >> 'RTB'  OR  Quantity < 200) ) \
   FIND  ( c'Nut' PREFIX (Part_Description) )                               \
   CHANGE( 'screw' 'bolt' WORD (Part_Description) )                         \
   NOUPDATE
```

A Formatted File Update. Records from data sets and members of PDS/PDSE libraries whose DSNs match the specified fileid mask are filtered so that only records that are of the record type "FX_Part_02" and match the FIND and WHERE criteria are processed by the CHANGE operation. SELECT indicates a subset of fields eligible to be searched, updated, and displayed in the output report. Both the FIND and CHANGE arguments further restrict string location/update to text within the field "Part_Description" only.

FSUEND

Syntax:

```
>>-- FSUEND-------------------------------------------------------------><
```

Description:

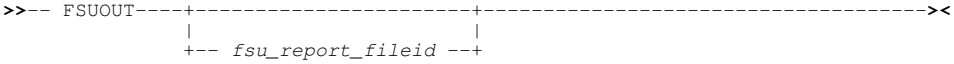
Use the FSUEND command to save and close the display of report output generated by a foreground execution of the File Search/Update/Copy/Remap utility.

If the report and its associated SDO structure file has not been saved, then FSUOUT will prompt the user to save both these files before the report is closed. The report data will be saved as a VSAM ESDS data set and the SDO as a physical sequential data set. The user will be prompted to enter allocation values but the defaults are usually acceptable.

FSUEND is assigned to <PF3> by default when an IOError has been reported or when a File Update operation has been performed. Therefore, execution of FSUEND from a command prompt is only necessary if the report generated by a File Search, Copy or Remap is to be saved.

FSUOUT

Syntax:



Description:

Use the FSUOUT command to display (browse) the saved report output from a previous execution of the File Search/Update/Copy/Remap utility.

FSUOUT executes the SDE EDIT command to display the specified FSU output report data set using a structure which has the same DSN but with additional low level qualifier ".SDO".

If no parameter is specified, the last saved FSU report output having the default report DSN format (*prefix*.FSU.Dyyyyddd.Thhmmss) is displayed.

FSUOUT may also invoked for an entry in a file list window using the "FO" prefix command.

Parameters:

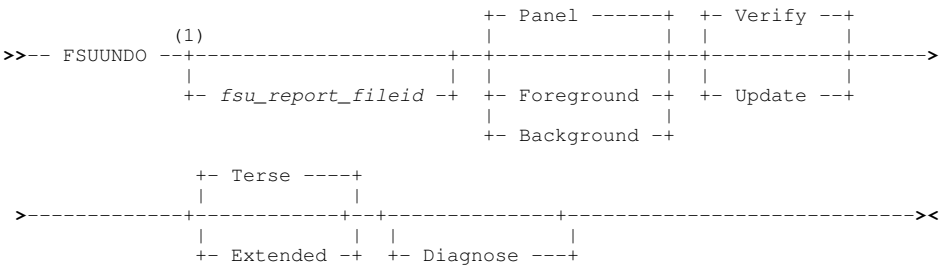
fsu_report_fileid
: The DSN of a report data set output generated by the File Search/Update/Copy/Remap utility.

Examples:

```
FSUOUT   DEV.NBJ.FSU.D2008268.T114326
```
: Display the FSU report for DSN 'DEV.NBJ.FSU.D2008268.T114326' using an existing structure of DSN 'DEV.NBJ.FSU.D2008268.T114326.SDO'.

FSUUNDO

Syntax:



Notes:

1. Parameters may be entered in any order.

Description:

Use the FSUUNDO command to execute the File Update Undo facility to restore all records in all files updated by the File Search/Update/Copy/Remap utility (FSU).

A File Search/Update/Copy/Remap report generated by the utility for an update operation, is used as input to FSUUNDO. This FSU report provides the affected fileids, the record numbers and images of the record data before and after the update was performed. Note that an attempt will be made to restore **all** records referenced as having been updated in the report.

Record data belonging to an entry flagged as "A" (After) in the "zT" field of the FSU report "Hit" records, will be replaced by data in the preceding FSU report record flagged as "B" (Before).

Before updating a record, the File Update Undo utility will first verify that the records data matches the updated record data in the report. This ensures that no restore will be performed if the record data has subsequent changes.

Unless parameter FOREGROUND or BACKGROUND is specified, FSUUNDO will open the File Search/Update/Copy/Remap panel. Other parameters specified on FSUUNDO will be reflected in the dialog fields.

Parameters:

fsu_report_fileid
: Specifies the DSN of the File Search/Update/Copy/Remap Output report to be used to identify changed data set records.

  The FSU report data set referenced may be that of **any** previous execution of the utility that involved a CHANGE operation.

  Default is the last saved FSU report, or, optionally, the FSU report displayed in the current SDE edit window view. If the current SDE edit window view contains an FSU output report (identified by its DSN format), the user will be prompted to use this report instead.

PANEL
FOREGROUND
BACKGROUND
: Specifies whether to open the FSUUNDO dialog window (PANEL), execute the FSUUNDO verify or update procedure immediately in TSO (FOREGROUND) or generate and display a JCL job deck suitable for submission to batch (BACKGROUND).

  Default is PANEL.

VERIFY
UPDATE
: Specifies whether to execute the UNDO procedure with or without performing an update of the record data.

  VERIFY provides the user with the opportunity to execute a "dry run" to examine the FSUUNDO output report for any errors before proceeding with an execution for UPDATE. It is strongly recommended that FSUUNDO is executed with VERIFY prior to performing a run for UPDATE. Use of VERIFY will be indicated at the start and end of the FSUUNDO report with the additional record beginning "** Verify Only".

  UPDATE will update records in the FSU reported data sets, so undoing the changes made by the File Update execution.

  Default is VERIFY.

TERSE
EXTENDED
: Specifies whether FSUUNDO is to output a brief (TERSE) or verbose (EXTENDED) report.

  In a terse report output, data sets or PDS(E) members that have been updated without error are represented by a single report line and data sets that have already been updated by a previous FSUUNDO run are not reported. However, more detailed report output is generated if unexpected data is found and so an error condition flagged.

  Extended report output will generate output for every successful or unsuccessful record update. See File Update Undo Output for more details.

  Default is TERSE.

DIAGNOSE
: Required only if a SELCOPY run time error occurs during execution of FSUUNDO, DIAGNOSE will remove the SELCOPY NOPRINT option and so write diagnostic report information to SYSPRINT.

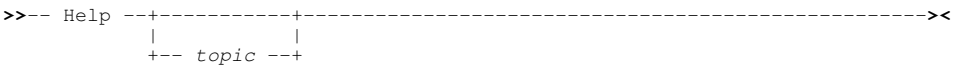
  If executing with parameter FOREGROUND, the SYSPRINT output is automatically displayed in a CBLe edit view with a DSN equal to the FSU output report DSN but with the additional low level qualifier "LIST". e.g.
  `NBJ2.DEV.FSU.D2008346.T162607.LIST`

GETXML

Executes the Data Editor primary command, GETXML to copy an XML document from a DB2 table to a data set, library member or HFS/ZFS file.

HELP

Syntax:



Description:

Use the HELP command to open the Help Window and optionally link directly to help on a specific CLI command. Where topic is not specified or not found, the relevant table of contents is displayed.

The Help window may also be opened via the Help item of the window's menu bar.

Parameters:

topic
: Display help on a specific topic.
If topic is enclosed in single or double quotes, the string is treated as the fileid of an HTML data set to be browsed. This may be the fully qualified fileid of an HTML document or the name of a PDS member that exists in the default HELP library.

If the help topic is not found, the Help Topic Index List Window is opened using the given topic as a search string.

Examples:

`HELP`
: Open the Help window contents page.

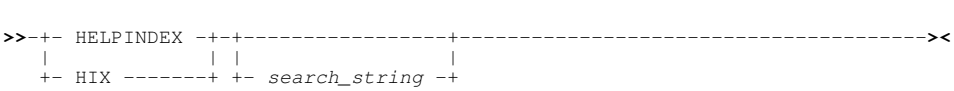
`HELP CBLe`
: Open the Help window at the CBLe command page.

`H "OEM.CBL.HTML(TEST)`
: Open a specific HTML document library member.

`H "ZZSISIZE"`
: Open the SELCOPY/i Help member name ZZSISIZE.

HELPINDEX (HIX)

Syntax:



Description:

Use the HELPINDEX command, which can be abbreviated as HIX, to open the Help Topic Index List Window and optionally filter the list with a search string. If no search string is given, the complete list of help topic titles is displayed in alphabetic order.

The Help Topic Index List window may also be opened by selecting *'Help Topic Index ...'* from the the CBLe main window menu bar.

The Help Topic Index List window is also opened if the HELP command is issued with a topic parameter which is not found. In this case the given topic is used as the search string, and all topics containing the string will be listed.

Note that the Help Topic Index List window is a standard SELCOPY/i List Window and as such supports all the features of this type of window.

Parameters:

search_string
: Search the list of help topic titles and display only those which contain the search string.

If the search string consists of more than one blank delimited sub-string, only those help topics will be displayed which have titles containing all of the sub-strings.

The search string is used to generate a filter command with a WHERE clause. For example if the command

```
hix unix command
```

is issued then the generated filter command is

```
where title<<unix & title<<command
```

HOME

Syntax:

```
>>-- HOme ------------------------------------------------------------------><
```

Description:

Edit the user's personal command centre (CMX) file. A new CBLe text edit session is opened if one is not already open.

IEBCOPYDIALOG

Syntax:

```
>>---- IEBCOPYDialog ------------------------------------------------------><
```

Description:

The IEBCOPYDIALOG command may be used to open the Execute IEBCOPY dialog window to copy members between PDS(E) libraries.

The dialog window will be opened with fields populated with parameters entered by the user during the last invocation of the window.

ISPF

Syntax:

```
>>--- ISPF --+----------------+---------------------------------------------><
             |                |
             +- ispf_command -+
```

Description:

When running in an ISPF environment, the SELCOPY/i command **ISPF** either toggles between using TSO and ISPF to manage 3270 I/O or executes an ISPF command. When used to execute an ISPF command, screen management is always handled by ISPF regardless of the current 3270 screen manager.

Note that when ISPF is the screen manager, the menu item **SwapList** is added to the CBLe main window menu bar. Selecting Swap will execute **ISPF SWAP LIST** to display ISPF's split screen menu.

It is recommended that, when running SELCOPY/i in an ISPF environment, ISPF should always be used as the 3270 screen manager to take advantage of ISPF screen split, etc. In order to do this without disrupting PFkey assignments, SELCOPY/i must run as an ISPF application with applid CBLI.

The *SELCOPY Product Suite Customisation Guide* provides instructions on customising SELCOPY/i to run as an ISPF application. When configured, there should never be any need to toggle back to TSO screen management. The supplied REXX macro, SELCOPYI, is used to run SELCOPY/i as an ISPF application with applid CBLI.

If SELCOPY/i is not defined as an ISPF application, then, when ISPF screen manager is used, SELCOPY/i function key definitions will be interpreted differently to those defined in SELCOPY/i. In this case, it is recommended that passing control to ISPF should only be carried out temporarily to perform ISPF explicit functions.

Toggling between ISPF and TSO screen management may also be achieved via the **Use TSO/ISPF** item of the System Menu.

Parameters:

ispf_command
ISPF command to be issued.

Examples:

`ISPF`
Set TSO as the screen manager if current screen management is done by ISPF **or** set ISPF as the screen manager if current screen management is done by TSO.

`ISPF SPLIT`
Set ISPF as the screen manager (if not already so) and execute ISPF SPLIT command so that the screen is split at the current cursor position.

ISPFUTIL

Syntax:

```
>>--+- ISPFUTIL -+-------------------------------------------------------><
    |            |
    +- IU -------+
```

Description:

When running SELCOPY/i in ISPF, the SELCOPY/i command **ISPFUTIL** starts the ISPF Utility Selection Panel.

The ISPF panel is started as a full screen application which returns control to SELCOPY/i only after it is closed.

The ISPF Utility Selection Panel may also be started via the **Utilities** menu of the SELCOPY/i main window menu bar.

JBOTTOM

Syntax:

```
>>--- JBottom ----------------------------------------------------------><
```

Description:

Move the focus window so that it's bottom border is justified with the bottom of the display window (i.e. the bottom border occupies the last line of the window display area). The window size is unchanged.

JLEFT

Syntax:

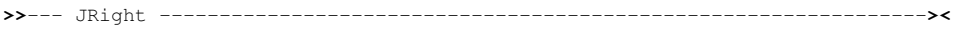
```
>>--- JLeft ------------------------------------------------------------><
```

Description:

Move the focus window so that it's left border is justified with the left hand side of the display window (i.e. the left border occupies the left column of the window display area). The window size is unchanged.

JRIGHT

Syntax:

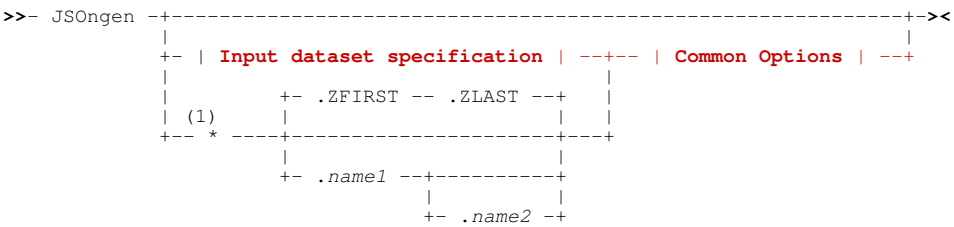


Description:

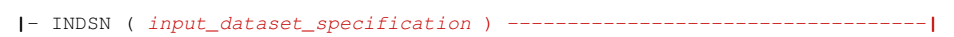
Move the focus window so that it's right border is justified with the right hand side of the display window (i.e. the right border occupies the right column of the window display area). The window size is unchanged.

JSONGEN

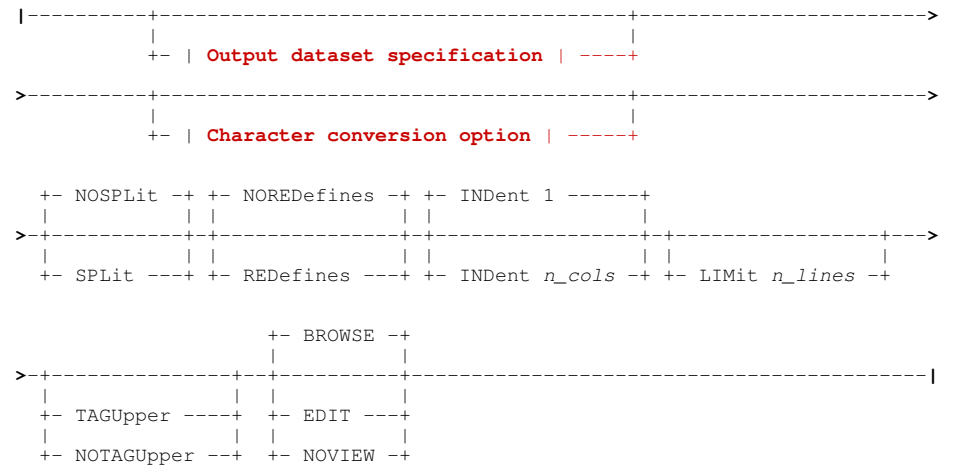
Syntax:



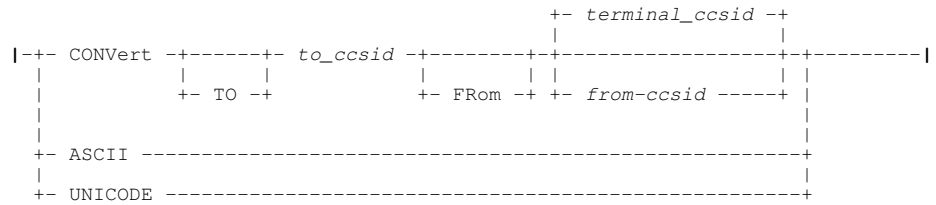
Input dataset specification:



Common Options:


|----------+----------+----------+

Character conversion option:



Notes:

1. Asterisk (*) is optional if any other parameter is specified.

Description:

JSONGEN is used to open the general JSON Generation panel or to immediately produce an exportable copy of structured data as JavaScript Object Notation (JSON) text. The structured data may belong to a specific structured data set or the current SDE view.

If asterisk (*), or any other parameter other than INDSN, is specified, then data in the current SDE edit or browse view will be used as input for JSON text generation. If no parameters are specified, the **JSON Generation** panel is opened.

JSONGEN reads an input structured data and writes an output text dataset consisting consisting of JSON {*name*:*value*} pairs. The JSON *names* correspond to the field names of the copybook/structure applied to the input dataset and the *values* to the field values expressed in character format.

The hierarchy of record types, group fields and elementary fields in the structure is represented by a corresponding nesting of JSON *name* identifiers.

The JSONGEN command may be executed in the foreground or via SDEIN input to program SDEAMAIN for batch processing.

During foreground execution a progress window is displayed showing input and output record counts, updated every second, which allows the user to interrupt processing before completion using the attention key.

Parameters:

Input dataset specification (INDSN)
Use of an JSONGEN input data set nominates a specific data set from which records are to be selected for JSON format output.

The JSONGEN input dataset may be specified explicitly in the command as the argument of the **INDSN** keyword. If the **INDSN** keyword is not present in the command then the contents of the current SDE structured dataset browse or edit view are used. If there is no current structured dataset the JSON Generation panel is opened.

INDSN (*input_dataset_specification*)
The input dataset specification is in the form of a structured edit BROWSE command (the BROWSE command verb is not required) which must be enclosed in parentheses following the INDSN keyword.

BROWSE keyword options such as **FROM**, **FOR**, **FILTER** and **VIEW** may be specified to limit the records from the input dataset which will be copied to the output dataset.

*
Required only if no other JSONGEN parameters are specified in order to immediately generate JSON output for data from the current SDE view (using defaults) as opposed to opening the general purpose JSON Generation panel or the SDE JSON Generation Panel as appropriate.

Start/End Line labels (.name1/.name2)
Applicable only to JSON output generated from data in the current SDE view, start and end line labels may be used to select a range of data records to be processed.

.name1
Corresponds to a label name *.name1* that identifies the first line in a range of SDE edit/browse lines. The preceding "." (period/dot) is mandatory.
Default is .ZFIRST.

.name2
Corresponds to a label name *.name2* that identifies the last line of a range of SDE edit/browse lines. The preceding "." (period/dot) is mandatory.
.name2 may occur on a line with a lower line sequence number than *.name1*. This is functionally equivalent to specifying *.name2* before *.name1* on the JSONGEN command.
Default is .ZLAST.

Output dataset specification (OUTDSN)
The JSONGEN output dataset may be specified explicitly in the command as the argument of the **OUTDSN** keyword. If the **OUTDSN** keyword is not present in the command then the value of the INI file variable **SDE.JSONGENOUTDSN** is

used if it exists, otherwise a default dataset name **userid.ZZS.JSONGEN** is used.

OUTDSN (*output_dataset*)
The output dataset name. Parentheses around the dataset name are accepted but not required. If this dataset exists its organisation may be sequential, a partitioned dataset member, VSAM (except LDS and KSDS) or HFS (ZFS). If it does not exist and JSONGEN is executing in batch the command terminates with an error message. If it does not exist and JSONGEN is executing interactively the user will be asked to allocate it (unless the name represents an HFS file in which case it will be implicitly defined).

This parameter may also be a DD name. If **output_dataset** consists of 8 or fewer characters and represents an allocated DD name then this allocated dataset is used for output.

MODIFY | **APPEND**
The output will be appended to the dataset if it exists (and is not partitioned). If this keyword is not specified the output will overwrite any existing dataset content.

EOL NL|CR|LF|CRLF
HFS file end of line specification. This parameter is accepted but ignored if the output dataset is not an HFS file. The values here are specified in EBCDIC, but if the output is subject to character conversion, the line end characters will also be converted. Note that ASCII does not have a NL (newline) character so if the output is being converted to a non-EBCDIC CCSID NL is changed to CR.

NL	X'15'	EBCDIC New Line. This is the default for EBCDIC output to an HFS file.
CR	X'0D'	EBCDIC Carriage Return.
LF	X'25'	EBCDIC Line Feed.
CRLF	X'0D25'	EBCDIC Carriage Return Line Feed.

Character conversion option (CONVERT/ASCII/UNICODE)

Since the purpose of JSONGEN is to produce a portable (data-interchange format) version of the data in a z/OS mainframe structured data file, and the output is character data, the coded character set identifiers (CCSIDs) of the input, output and of the JSONGEN internal constants themselves are of significance.

Even if the input and output is coded in an EBCDIC CCSID, these may differ, and both may differ from the CCSID of the JSONGEN command's internal constants. Since some of the special characters used in JSON have different code points in different EBCDIC CCSIDs (for example quotation marks) these must be dealt with consistently to produce correct JSON output.

JSONGEN uses the z/OS character conversion support supplied by IBM modules CUNLINFO (for obtaining CCSID information) and CUNLCNV (for character conversion from one CCSID to another).

The internal JSONGEN CCSID (that of the constants used to build the JSON syntax) is CCSID 285 (EBCDIC, SBCS UNITED KINGDOM).

JSONGEN assumes a default CCSID as follows:

Interactive
When executed interactively JSONGEN uses as default input CCSID that of the user's 3270 terminal.

Batch
When executed in batch JSONGEN uses as default input CCSID the value of the INI file variable **SDE.CCSID**. This variable is set automatically to the user's 3270 terminal CCSID (if not already set) during an interactive session. It can also be set using the structured data SET CCSID command.

If no explicit conversion is specified, the JSON output dataset is produced using the default CCSID and the input dataset character fields are assumed to be in the same CCSID. The internal JSONGEN constants are converted from internal CCSID 285 to the default CCSID.

CONVERT
Use this keyword to request character CCSID conversion.

to_ccsid
The CCSID of the output JSON text dataset. Internal JSONGEN character literals and input character data fields (and HFS line end characters if used) are converted to this CCSID.

from_ccsid
The input character data fields are converted from this CCSID. If this parameter is not supplied the default input CCSID is used.

ASCII
Convert the output to ASCII. This is equivalent to specifying **CONVERT TO 819**. CCSID 819 is ISO 8859-1 ASCII.

UNICODE
Convert the output to UNICODE (UTF-16). This is equivalent to specifying **CONVERT TO 1200**. CCSID 1200 is the IBM bigendian UTF-16 CCSID which is automatically transformaed to the most recent UTF-16 standard.

Ouput line splitting option (SPLIT)

For each elementary input field, JSONGEN builds one output record containing the field name and value. Depending on the nature of the input data, relatively long output records may result. If an output record is longer than the allocated

logical record length of the output dataset, this option controls how JSONGEN deals with the long output record.

NOSPLIT
Do not split the output record. Rather than truncate the output record JSONGEN terminates with an error message. This is the default.

SPLIT

Split the output record breaking it up into as many logical records as necessary. Records are split at the logical record length irrespective of the record content.

Redefined field selection option (REDEFINES)
If the structure defined for the input dataset contains redefined fields this option controls whether the field redefinitions are output.

NOREDEFINES
Do not output the field redefinitions. This is the default.

REDEFINES
Output all field redefinitions.

JSON nested indentation option (INDENT)
Nested output JSON names corresponding to the hierarchy of group and elementary data fields in the input structure are indented by a default of one space for each data item level. This option allows the specification of a different indentation value.

INDENT *n_cols*
The indentation value (default 1).

Limit number of input lines (LIMIT)
The number of input records, record segments or DB2 table rows processed may be restricted using this option.

LIMIT *n_lines*
The maximumm number of input lines. Default is all lines.

Output view option (BROWSE/EDIT/NOVIEW)
When JSONGEN is run interactively this option allows the user to request to view the output when the command completes.

BROWSE
Browse the output JSON dataset. This is the default when run interactively.

EDIT
Edit the output JSON dataset using the SELCOPYi text editor.

NOVIEW
Do not view the output JSON dataset. This option is forced when run in batch.

Uppercase tag name option (TAGUPPER/NOTAGUPPER)
This option controls whether JSON tag names generated from the structure field names are upper cased. The default action is controlled by the global SDE option, NAMECASE.

TAGUPPER
Tag names are uppercased.

NOTAGUPPER
Tag names are not uppercased. Tags match the field names with no upper case translation.

Examples:

Using the same COBOL copybook and source data set as that used in the description of XMLGEN, the following JSONGEN command may be executed to generate JSON format output:

```
  jsongen  indsn( CBL.XMLEXAMP   using cobol CBL.COB(XMLEXAMP) )
          outdsn( CBL.JSONGEN )
         noredefines    indent 3     edit
```

The JSONGEN output file is edited as a result of the EDIT keyword parameter in the command. Note that each definition starts with name, XMLEXAMPLE, which is the record type (level 1) name associated with each of the 3 input records.

```
-CBL.JSONGEN     27990 V SEQ   Size=36   Alt=0,0;0
      <---+----1----+----2----+----3----+----4----+----5----+----6----+--
00001 {
00002    "XMLEXAMPLE" :
00003    {
00004       "EMPLOYEE" :
00005       {
00006          "FIRSTNAME" : "John",
00007          "LASTTNAME" : "Doe"
00008       },
00009       "AGE" : 52,
00010       "SALARY" : 33000.00
00011    }
00012 }
00013 {
00014    "XMLEXAMPLE" :
00015    {
00016       "EMPLOYEE" :
00017       {
00018          "FIRSTNAME" : "Amy",
00019          "LASTTNAME" : "Johnston"
00020       },
00021       "AGE" : 28,
00022       "SALARY" : 41500.00
00023    }
00024 }
00025 {
00026    "XMLEXAMPLE" :
00027    {
00028       "EMPLOYEE" :
00029       {
00030          "FIRSTNAME" : "Freda",
00031          "LASTTNAME" : "Bloggs"
00032       },
00033       "AGE" : 39,
00034       "SALARY" : 27800.00
00035    }
00036 }
00037 * * * End of File * * *
```

JTOP

Syntax:

```
>>--- JTop ------------------------------------------------------------------><
```

Description:

Move the focus window so that it's top border is justified with the top of the display window (i.e. the top border occupies the first line of the window display area). The window size is unchanged.

KEYS

Syntax:

```
>>--+- KEYS -----+--+-------------------------------------------------------+--><
    |            |  |                                                       |
    +- SETFKEYS -+  |  +-- DEFAULT ------+                                   |
    |            |  |  |                |                                   |
    +- PF -------+  +--+----------------+--+--------------------------------+
    |            |     |                |  |                                |
    +- SFK ------+     +-- windowname ---+  +- pfn --+----------------------+
                       |                |           |                      |
                       +-- windowclass --+           +------------+-- cmd -+
                       |                |           |            |
                       +-- CAPTION ------+           +-- DELAY ---+
                       |                |           |            |
                       +-- BORDER -------+           +-- BEFORE --+
```

Description:

Use the KEYS command to assign a command to a (non-KeyList) function key or display the Function Keys dialog.

Parameters:

null
: Opens the Function Keys dialog. for the current window.

DEFAULT
: Selects the system default function key table.

CAPTION
: Selects the window caption function key table.

BORDER
: Selects the window border function key table.

windowname
: Selects the function key table of the window name specified.

windowclass
: Selects the function key table of the window class specified.

pfn
: The function key number 1-24.

cmd
: The command text string to be assigned. If null then the function key becomes unassigned.

DELAY
: Place the function key command on the command line when the key is pressed rather than execute it.

BEFORE
: Execute the function key command before processing any user screen inputs.

Examples:

```
KEYS  EDTWEDIT     16    'macro delblank'
```
Set PF16 to execute user edit macro DELBLANK for all windows of windowclass EDTWEDIT.

KEYS Dialog

For SELCOPYi Rel 3.20 onward, on z/OS systems SELCOPYi maintains its function key definitions in **KEYLIST**s, analogous to ISPF keylists.

Use of KeyLists may be switched on/off using the Function Keys Settings (=0.8) panel, or the KEYLIST primary command.

1. KEYS Dialog (KEYLISTs ON)
2. KEYS Dialog (KEYLISTs OFF)

KEYS Dialog (KEYLISTs ON)

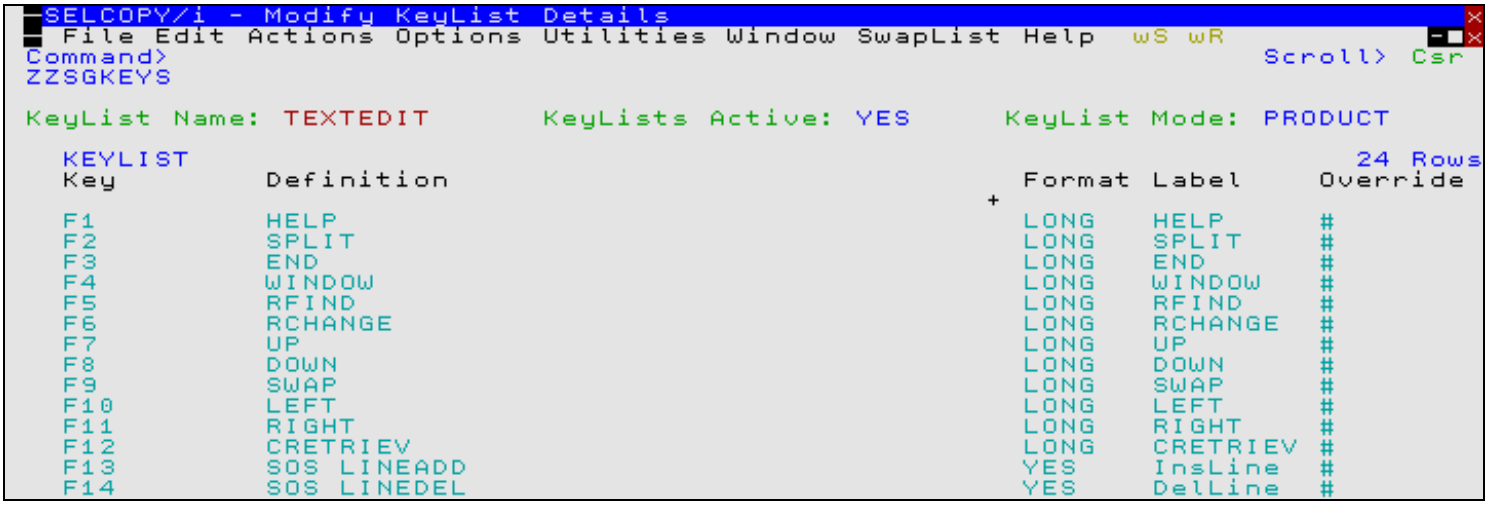


Figure 316. Keys Dialog for Text Edit - KEYLISTs On.

Column Name	Description
Key	Function Keys F1-F24
Definition	The command(s) associated with the Function Key. A number of commands may be issued if separated by the appropriate separator character.
Format	The PFSHOW format (YES|NO|LONG|SHORT) as for standard ISPF. Valid entries are YES, NO, LONG or SHORT.
Label	The text displayed by PFSHOW.
Override	The level at which the key is defined. This field will be maintained automatically and should not be updated by the user. Possible values are # (Product Default), Site, or Personal.

KEYS Dialog (KEYLISTs OFF)

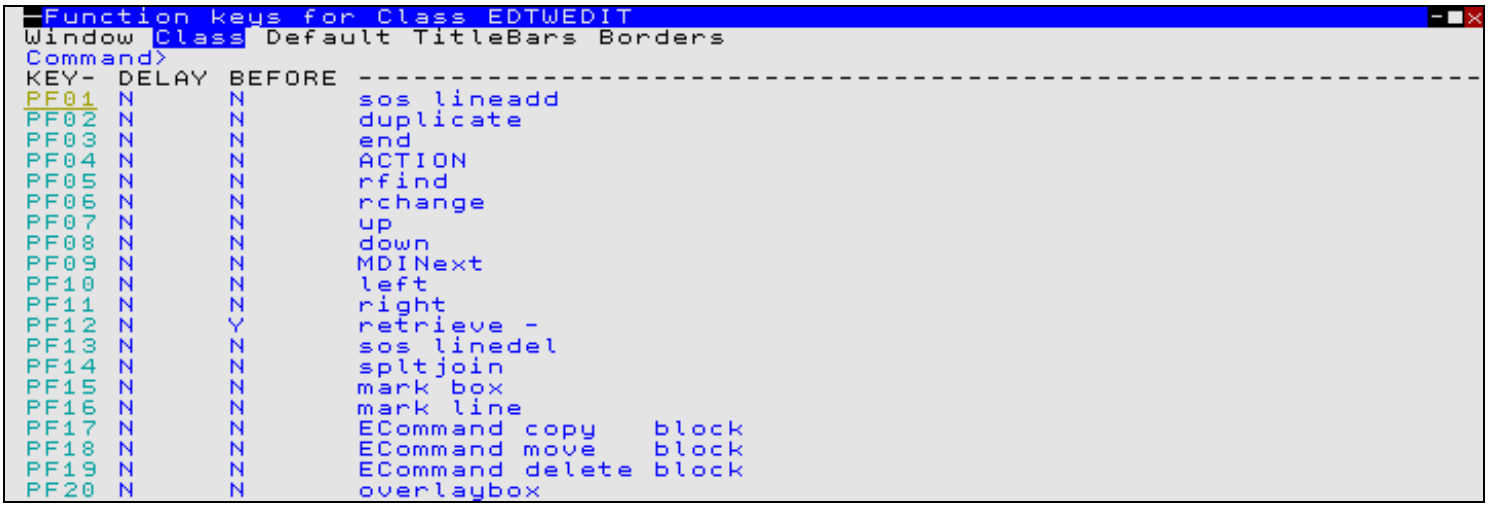
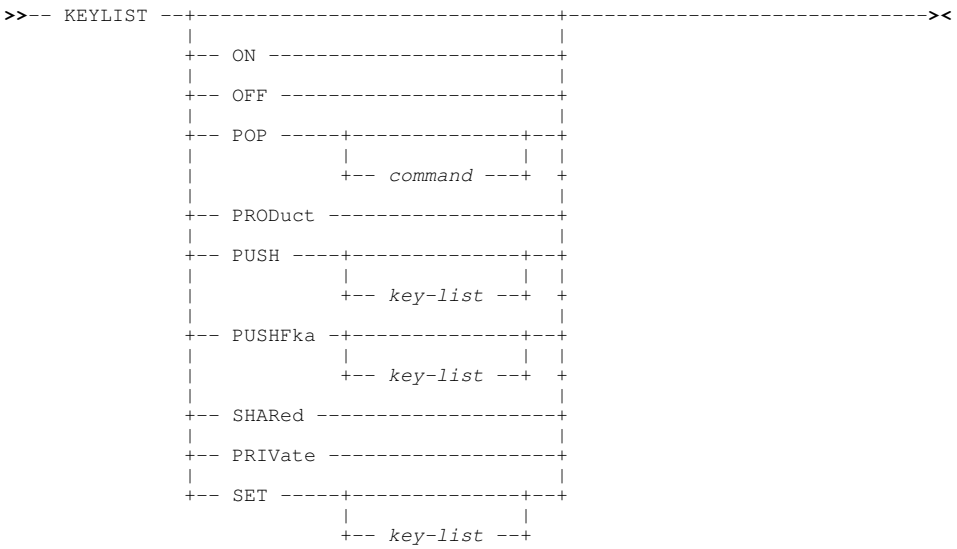


Figure 317. Keys Dialog for Text Edit - KEYLISTs Off.

Column Name	Description
KEY	PFKeys 01-24 (Non-enterable field)
DELAY	Determine whether the associated command is executed immediately when the function key is hit or merely placed on the local command line. Valid entries are Y or N.
BEFORE	Determine whether the associated command is executed before or after any other CBL3270 screen input. (e.g. a command line command or prefix area command.) Valid entries are Y or N.
untitled	The command(s) associated with the PFKey. A number of commands may be issued if separated by the appropriate separator character.

KEYLIST

Syntax:



Description:

Use the KEYLIST command with no parameters to display a selection-list of all loaded Function Key list objects. From the list any KeyList may be selected, either for view/edit or for use by the window from which the KEYLIST command was entered.

KEYLIST PUSH, PUSHFKA and POP

Primarily introduced for use in keyboard macros supported by 3270 emulator software, KEYLIST PUSH and POP may be assigned to PFKeys to support simple switching between keylist definitions.

Using PUSH and POP techniques, a 3270 emulator macro may comprise a number of PFKey keystrokes each performing separate primary commands. Note that, if primary commands were to be coded directly in a keyboard macro, the macro would have to ensure that the cursor was first positioned at a command prompt. This cannot be performed reliably and so assigning the required commands to PFKeys, which do not depend on being executed at a command prompt, simplifies the macros and provides more reliable execution.

For example, suppose a keyboard macro is to be assigned to Ctl-S in order to save the data set in the focus Text Edit or Data Edit view. Since primary command SAVE is not by default assigned to one of the standard F1-F24 PFKeys, we must first open the Function Keys Settings panel (=0.8) to edit our personal keylist settings. By adding entries for list names DATAEDIT, TEXTEDIT, @WINBORD and @WINTITL as follows, the default action on the F24 key is updated to save the current keylist value and then set the current keylist to be @@ALTXX (without changing the function key display area).

```
TEXTEDIT PF24   Long    @@ALTXX    KEYLIST PUSHF @@ALTXX
DATAEDIT PF24   Long    @@ALTXX    KEYLIST PUSHF @@ALTXX
@WINBORD PF24   Long    @@ALTXX    KEYLIST PUSHF @@ALTXX
@WINTITL PF24   Long    @@ALTXX    KEYLIST PUSHF @@ALTXX
```

In the same personal keylist settings, we define a new, alternate keylist @@ALTXX with entries that execute a command and then reset the keylist to the saved version. e.g.

```
@@ALTXX  PF19   Long    SAVE       KEYLIST POP   SAVE
```

Having saved these changes, we can write a 3270 emulator macro that simply executes F24 followed by F19, and then assign it to Ctl-S. Use of Ctl-S will then have the desired affect of saving edited changes to the dataset in the focus view regardless of the cursor position within the focus window display area.

Alternate keylists, for use with 3270 keyboard macros, are supplied as standard in the SELCOPYi product keylist table. Please contact the CBL support desk <support@cbl.com> for assistance if you wish to make use of these keylists in conjuction with 3270 keyboard macros. A zipped archive of keyboard macros for IBM's Personal Communications and Tom Brennan's Vista3270 emulator software is also available from the CBL web site (www.cbl.com).

Parameters:

ON
: Sets the use of KEYLISTs on.

`OFF`
: Sets the use of KEYLISTs off, in which case SELCOPYi uses Function Key settings at 5 levels (i.e. Window, Class, Default, TitleBars and Borders). The function actioned by any key is then determined by these levels as discussed in the topic Function Keys.

`POP [ command ]`
: Used in conjunction with KEYLIST PUSH or KEYLIST PUSHFKA, POP will restore the current keylist back to the keylist setting saved by PUSH.

  Optional parameter *command*, representing any supported primary command, may be specified so that the reset of the keylist occurs only after *command* has been executed.

`PRODUCT`
: Definitions will be loaded from the product supplied KeyList table only. Alteration of key definitions using the KEYS dialog will not be permitted in PRODUCT mode.

`PUSH [ key_list ]`
: Performs the same action as parameter SET except that the current keylist name is also saved. A subsequent **KEYLIST POP** may be executed to reset the keylist back to this saved setting.

`PUSHFKA [ key_list ]`
: Performs the same action as PUSH but will not refresh the function key display area at the bottom of the screen. See PFSHOW for details on function key display.

`SHARED`
: Definitions will initially be loaded from the product supplied KeyList table, then site-wide overrides will be applied by loading an installation defined table. Alteration of key definitions using the KEYS dialog will not be permitted SHARED mode.

`PRIVATE`
: Alteration of key definitions using the KEYS dialog will be permitted. Definitions will be loaded as for SHARED with additional personal overrides loaded from (and saved to) the user's own KeyList table.

`SET [ key_list ]`
: Sets the KeyList for the current window. If a keylist name is not supplied then a temporary keylist named **@TMPnnnn** (where nnnn is the next sequence number) will be created, initially modelled on the current keylist. Any temporary keylist will be destroyed when a different keylist name is set for the owning window, or that window is closed.

  e.g.

```
KEYLIST SET TEXTEDIT
```

KEYLIST Dialog

For SELCOPYi Rel 3.20 onward, on z/OS systems SELCOPYi maintains its function key definitions in **KEYLIST**s, analogous to ISPF keylists.

Use of KeyLists may be switched on/off using the Function Keys Settings (=0.8) panel, or the KEYLIST primary command.

The KEYLIST dialog allows the user to select from a list of all currently loaded Keylists, either for view/edit (default) or for use by the invoking window.

Column Name	Description
KeyList	Name of a loaded KeyList

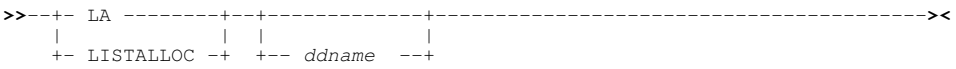
Prefix Line Commands

The following prefix commands may be entered in the prefix area against any entry in the generated list.

Command	Description
<Dflt>	Prefix Line command E.
E	Open the KEYS dialog to view/edit the key definitions for this KeyList.
U	Use this keylist. Closes the KEYLIST dialog and sets the keylist for the window that invoked it.

LA

Syntax:


>>--+- LA --------+--+-------------+---><
| | | |

Parameters:

entry

Specifies the fileid mask used to select cataloged entries, which is placed in the Entry field of the Associations List window.

The fileid mask represents a DSN mask that supports the following wild cards:

- `*` A single asterisk represents a DSN qualifier, or zero or more characters within a DSN qualifier.
- `**` A double asterisk represents zero or more qualifiers within a DSN. Double asterisk must be preceded or followed by either a "." (dot/period) or a blank. It cannot precede or follow an alphanumeric character.
- `%` A single percent sign represents exactly one character, other than "." (dot/period), within a DSN qualifier. Up to 8 percent signs can be specified in each qualifier.

If the last character of the fileid mask is "." (dot/period), then this marks the end of the low level DSN qualifier within the fileid mask. The trailing "." is stripped and no wildcard string is appended to the fileid mask. e.g.

```
DEV*.                 becomes:   DEV*
DEV.OEM.TRSPAN*.      becomes:   DEV.OEM.TRSPAN*
DEV.*.*SAMP%%.        becomes:   DEV.*.*SAMP%%
```

If the last character of the fileid mask is **not** "." (dot/period), then a default trailing wild card string is automatically appended to the fileid mask as follows:

1. If the fileid mask is a single qualifier or the last character of the fileid mask is "*" (asterisk), then a wildcard string of ".**" is appended. e.g.

   ```
   DEV                   becomes:   DEV.**
   DEV*                  becomes:   DEV*.**
   DEV.OEM.TRSPAN*       becomes:   DEV.OEM.TRSPAN*.**
   DEV.*.*SPA*           becomes:   DEV.*.*SPA*.**
   ```

2. Otherwise a wildcard string of "*.**" is appended. e.g.

   ```
   DEV.OEM.TRSPAN        becomes:   DEV.OEM.TRSPAN*.**
   DEV.*.*SPA%           becomes:   DEV.*.*SPA%*.**
   SYS1.*.Z19            becomes:   SYS1.*.Z19*.**
   ```

Note that a warning message is displayed if the high level qualifier of the fileid mask is "*" (asterisk) or "**" (double asterisk). A fileid mask of this type would result in all catalogs being searched which would take some time to execute and would use a large amount of system resources.

catalog

Specifies the catalog in which to search for the requested entry.

This is a catalog DSN. Specifying a catalog DSN is unneccessary if an alias exists for the fileid mask high level qualifier (HLQ) in the master catalog. In this case, the appropriate catalog DSN will automatically be inserted in this field. If the HLQ contains a wild card, then all matching aliases are interrogated, the required catalogs are searched and the last catalog searched placed in the Catalog> field.

An "*" (asterisk) may be specified to imply the default catalog name. This need only be specified if the *types* parameter is to be used.

Default is the master catalog.

The *catalog* string is placed in the Catalog field of the Catalog List window.

types

Specifies the catalog entry types required. Default is all types. One or more of the following types may be specified with no intervening blanks:

A	non-VSAM (or VSAM SAM) data set.
B	MVS - Generation data group.
C	Cluster.
G	Alternate Index.
H	MVS - Generation data set.
R	VSAM PATH.
X	Alias.
U	User catalog connector entry.
L	MVS - Tape volume catalog library entry.
W	MVS - Tape volume catalog volume entry.

Default is to select entries of all types.

The *types* string is placed in the Types field of the Catalog List window.

Examples:

```
las   CBL.%%C
```
List associations for cataloged entries matching the fileid mask "CBL.%%C*.**".

```
las   CBL.**   *    A
```
List associations for non-VSAM cataloged entries matching the fileid mask "CBL.**". (This will display defined ALIAS names for non-VSAM data sets.)

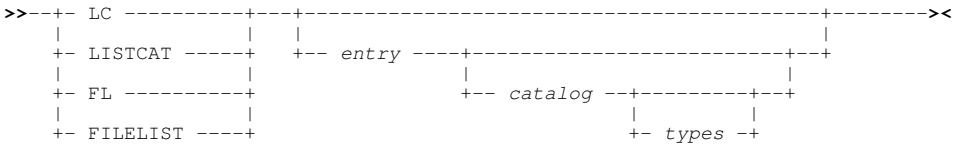
```
las   NBJ      *    H
```
List associations for GDG data sets matching the fileid mask "NBJ*.**". (This will display the GDG Base name for each selected GDG data set.)

```
las   NBJ.**.X232.   *    R
```
List associations for VSAM PATH entries matching the fileid mask "NBJ.**.X232".

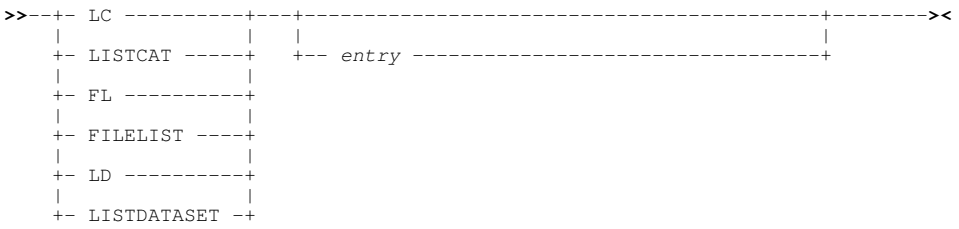
LC

Syntax:

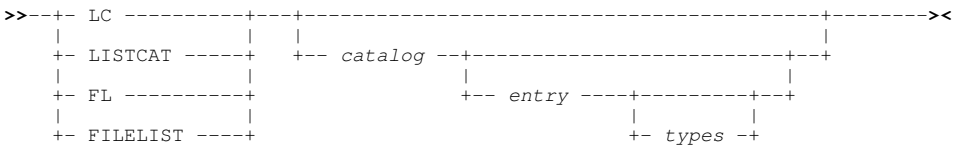
Open an MVS Cataloged Entries List Window:



Open a CMS File List Window:



Open a VSE Catalog List Window:



Description:

For both MVS and VSE, the LC (List Catalog entries) command is used to open a Catalog List window and optionally list basic information about entries in the catalog. This is a less detailed list than that generated by the LD command.

For CMS, the LC command opens the File List window in place of the Catalog List window and displays information about matching files entries that reside on the accessed mini-disk(s).

For MVS only, if *entry* is not a valid MVS data set name mask, then the LC command opens the List HFS Path window to display information about matching entries in the HFS or ZFS file systems.

For VSE only, the LC command is supported only if the CBL software product CBLVCAT is installed and active. The LC command uses CBLVCAT to read the specified VSAM catalog records and obtain information about the cataloged files.

The Catalog List window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Parameters:

entry

Specifies the fileid mask which is placed in the Entry field of the MVS/VSE Catalog List window or the File field of the CMS File List window.

◊ For **MVS** systems, the fileid mask represents a DSN mask that supports the following wild cards:

`*` A single asterisk represents a DSN qualifier, or zero or more characters within a DSN qualifier.

`**` A double asterisk represents zero or more qualifiers within a DSN. Double asterisk must be preceded or followed by either a "." (dot/period) or a blank. It cannot precede or follow an alphanumeric character.

`%` A single percent sign represents exactly one character, other than "." (dot/period), within a DSN qualifier. Up to 8 percent signs can be specified in each qualifier.

If the last character of the fileid mask is "." (dot/period), then this marks the end of the low level DSN qualifier within the fileid mask. The trailing "." is stripped and no wildcard string is appended to the fileid mask. e.g.

```
DEV*.                 becomes:   DEV*
DEV.OEM.TRSPAN*.      becomes:   DEV.OEM.TRSPAN*
DEV.*.*SAMP%%.        becomes:   DEV.*.*SAMP%%
```

If the last character of the fileid mask is **not** "." (dot/period), then a default trailing wild card string is automatically appended to the fileid mask as follows:

1. If the fileid mask is a single qualifier or the last character of the fileid mask is "*" (asterisk), then a wildcard string of ".**" is appended. e.g.

```
DEV                   becomes:   DEV.**
DEV*                  becomes:   DEV*.**
DEV.OEM.TRSPAN*       becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA*           becomes:   DEV.*.*SPA*.**
```

2. Otherwise a wildcard string of "*.**" is appended. e.g.

```
DEV.OEM.TRSPAN        becomes:   DEV.OEM.TRSPAN*.**
DEV.*.*SPA%           becomes:   DEV.*.*SPA%*.**
SYS1.*.Z19            becomes:   SYS1.*.Z19*.**
```

Note that a warning message is displayed if the high level qualifier of the fileid mask is "*" (asterisk) or "**" (double asterisk). A fileid mask of this type would result in all catalogs being searched which would take some time to execute and would use a large amount of system resources.

◊ For **VSE** systems, the fileid mask is a valid CBLVCAT LISTCAT KEY parameter string. i.e. entries with file name **beginning** with the specified string or, if prefixed by "/" (slash), entries with file name **containing** the specified string. (See the CBLVCAT User Manual.)

If no fileid mask is specified, all entries will be selected.
Note that wild cards are not supported within the VSE fileid mask, however, "*" (asterisk) is tolerated if placed at the end of the fileid mask.

An "*" (asterisk) may also be specified in place of the fileid mask to imply that all entries are to be selected. This need only be specified if the *types* parameter is to be used.

◊ For **CMS** systems, the fileid mask may consist of up to 3 qualifiers representing a filename filetype filemode combination where qualifiers are separated by one or more blanks or a "." (dot/period).

A single "*" (asterisk) wild card may be used to represent an entire qualifier or zero or more characters at a particular position within the qualifier. Wild card "*" may be specified more that once, anywhere within a qualifier.

Default CMS filemode qualifier is "A", default CMS filetype qualifier is "*".

catalog

Specifies the catalog in which to search for the requested entry.

For **MVS** systems, this is a catalog DSN. Specifying a catalog DSN is unneccessary if an alias exists for the fileid mask high level qualifier (HLQ) in the master catalog. In this case, the appropriate catalog DSN will automatically be inserted in this field. If the HLQ contains a wild card, then all matching aliases are interrogated, the required catalogs are searched and the last catalog searched placed in the Catalog> field.

An "*" (asterisk) may be specified to imply the default catalog name. This need only be specified if the *types* parameter is to be used.

For **VSE** systems, this is a disk label assigned to the VSAM catalog for which entries are to be listed.

Default for both MVS and VSE is the master catalog.

The *catalog* string is placed in the Catalog field of the Catalog List window.

types
Specifies the catalog entry types required. Default is all types. One or more of the following types may be specified with no intervening blanks:

A	non-VSAM (or VSAM SAM) data set.
B	MVS - Generation data group.
C	Cluster.
G	Alternate Index.
H	MVS - Generation data set.
R	VSAM PATH.
X	Alias.
U	User catalog connector entry.
L	MVS - Tape volume catalog library entry.
W	MVS - Tape volume catalog volume entry.

Default is to display entries of all types.

The *types* string is placed in the Types field of the Catalog List window.

Examples:

```
lc   CBL.%%C
```
List MVS cataloged entries matching the fileid mask "CBL.%%C*.**".
```
lc   CBL.SYS*.**   *    A
```
List MVS non-VSAM cataloged entries matching the fileid mask "CBL.SYS*.**".

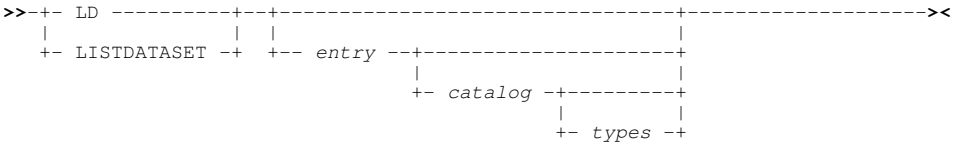
```
lc   IJSYSCT  *    U
```
List user catalogs in the VSE VSAM master catalog.

```
lc   VSESPUC  /CICS
```
List all files containing the string "CICS" in the VSE VSAM catalog "VSESP.USER.CATALOG".

```
fl   CBL*.EXEC
```
List CMS files that match the fileid mask "CBL* EXEC A".

LD

Syntax:



Description:

For CMS only, the LD and LC commands are synonyms for the FL command and so opens the File List window. Refer to documentation for LC (FL) for syntax and parameter specification.

The LD command is not supported on **VSE** systems.

For MVS, the LD (List Dataset details) command is used to open a Dataset List window and optionally list entries in the catalog together with the details of their geometry obtained either from the catalog or the VTOC. This is a more detailed list than that generated by the LC command.

Where *entry* is identified as being an HFS path (i.e. begins with "." or contains "/"), the HFS Path list window is opened instead.

The Dataset List window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Parameters:

entry
Specifies the fileid mask used to identify required catalog entries.

The fileid mask represents a DSN mask that supports the following wild cards:

- `*` A single asterisk indicates that either a qualifier or one or more characters within a qualifier can occupy that position. An asterisk can precede or follow a set of characters.
- `**` A double asterisk indicates that zero or more qualifiers can occupy that position. A double asterisk cannot precede or follow any characters; it must be preceded or followed by either a dot or a blank.
- `%` A single percent sign indicates that exactly one character can occupy that position. (Up to 8 percent signs can be specified in each qualifier.)

Unless the **last** character of the fileid mask is a wild card "*" (asterisk) or a "." (dot/period), then a default trailing wild card string is appended to the fileid mask as follows:

1. If the fileid mask is a single qualifier or the last qualifier is length 8, a wildcard string of ".**" is appended. e.g.

```
DEV                  becomes:   DEV.**
DEV.OEM.TRSPAN00     becomes:   DEV.OEM.TRSPAN00.**
DEV.*.TRSPAN00       becomes:   DEV.*.TRSPAN00.**
```

2. A wildcard string of "*.**" is appended. e.g.

```
DEV.OEM.CBL202       becomes:   DEV.OEM.CBL202*.**
SYS1.ZOS             becomes:   SYS1.ZOS*.**
```

The ammended *entry* string is placed in the Entry field of the Dataset List window.

catalog

Specifies the catalog in which to search for the requested entry.

This is a catalog DSN. Specifying a catalog DSN is unneccessary if an alias exists for the fileid mask high level qualifier (HLQ) in the master catalog. In this case, the appropriate catalog DSN will automatically be inserted in this field. If the HLQ contains a wild card, then all matching aliases are interrogated, and the required catalogs are searched.

An "*" (asterisk) may be specified to imply the default catalog name. This need only be specified if the *types* parameter is to be used. Default is the master catalog.

The *catalog* DSN searched is placed in the Catalog field of the Dataset List window.

types

Specify the catalog entry types required. Default is all types. One or more of the following types may be specified with no intervening blanks:

A	non-VSAM (or VSAM SAM) data set.
B	MVS - Generation data group.
C	Cluster.
G	Alternate Index.
H	MVS - Generation data set.
R	VSAM PATH.
X	Alias.
U	User catalog connector entry.
L	MVS - Tape volume catalog library entry.
W	MVS - Tape volume catalog volume entry.

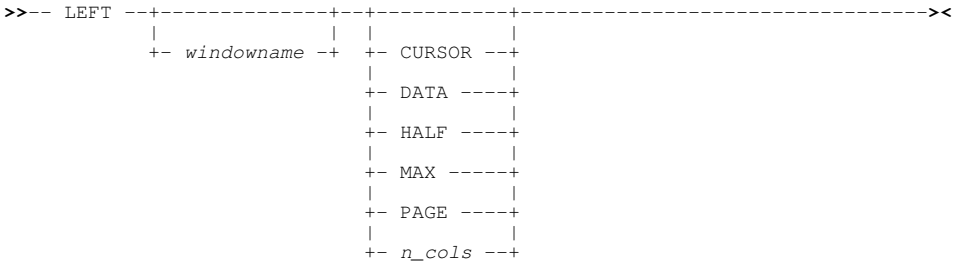
The *types* parameter string is placed in the Catalog field of the Dataset List window.

Examples:

```
LD CBL.%%C
LD CBL.SYS*.**   USERCAT.CBLCAT  A
```

LEFT

Syntax:



Description:

Scroll the view of the data within the specified window left towards the first column of the displayable data.

The extent by which data is scrolled is determined by the CURSOR, DATA, HALF, PAGE, MAX or *n_cols* parameter which may be specified using any one of three methods determined in the following order of precedence:

1. The scrolling command verb, LEFT, and one of these scrolling parameters is explicitly specified on the command line.
2. The scrolling parameter is specified on the command line and a PFKey assigned to LEFT is actioned.
   Note that the contents of a command line are appended to the command stream assigned to a PFKey when that PFKey is actioned.
3. No scrolling parameter is specified, so the current value of the "Scroll>" field is used.
4. No scrolling parameter is specified and no "Scroll>" field is present, so a defualt of one column is used.

List windows may contain fields that have **KEY** attribute **YES** defined in the Field Descriptor Block. Fields with this attribute are always in view and may not be scrolled right or left. If the cursor is positioned in a column belonging to this type of field, then, for LEFT CURSOR and RIGHT CURSOR, the cursor is considered to be outside the display area. All columns of data that do not belong to a **KEY** field are scrollable using LEFT and RIGHT.

By default this command is assigned to function key **PF10**.

Parameters:

windowname
: The window name of the window in which the display is to be scrolled. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

CURSOR
: The scrollable column on which the cursor is positioned becomes the last scrollable column of the display.
If the cursor is positioned outside the display area, in a KEY field or on the last scrollable column within the display area, then LEFT PAGE is executed instead.

DATA
: Scroll left so that the first scrollable column in the current display area becomes the last scrollable column of the display.

HALF
: Scroll a number of columns so that the column situated half way along the width of the current display of scrollable columns, becomes the last scrollable column of the display.

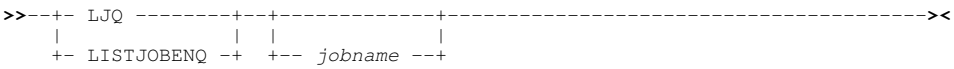
MAX
: Scroll left to display the first scrollable column of data.

PAGE
: Scroll left so that the scrollable column of data to the left of the first scrollable column in the current display, becomes the last scrollable column of the display.

n_cols
: Scroll left a specified number of floating columns.
The scrollable column of data that is *n_cols* to the left of the first scrollable column becomes the new first scrollable column of the display.

LJQ

Syntax:



Description:

Use the LJQ (List MVS Job Enqueues) command to open a Job Enqueue List window containing outstanding MVS enqueues held by a given job.

The Job Enqueue List window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Note: Not implemented for CMS or VSE.

Parameters:

jobname
The name of the job for which the ENQueues are to be listed.

This parameter is placed in the JobName field of the Job Enqueue List window.

See Also:

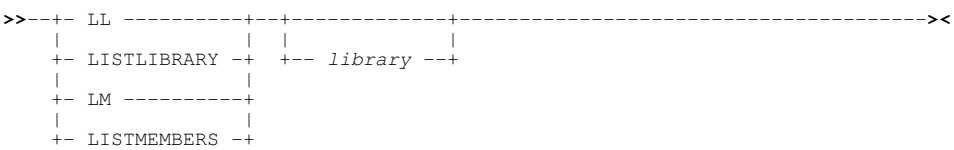
LQ Command

Examples:

```
LJQ  NBJTSO
```
List Enqueues for job NBJTSO.

LL

Syntax:



Description:

Use the LL (List Library) command to open a Library List window and optionally list the members of an z/OS PDS/PDSE or z/VSE LIBR library.

Alternatively, if a member generation mask is specified, LL will open a Library Member Generations list window. Listing member generations is only valid for PDSE version 2 libraries that have been allocated a MAXGENS value.

Parameters:

library
The name of the library for which the contents are to be listed and is placed in the Library> field of the Library List window or Library Member Generations window.

◊ For **z**/**OS**, *library* is a PDS (or PDSE) dataset name which may be followed by one or more member name masks with or without member generation masks.

If specified, the member name masks must immediately follow the library DSN and be enclosed in "( )" (parentheses). If no member name masks are specified, a default of mask of "(*)" is used indicating that all member entries are to be selected and, since no generation mask is implied, a library member list window will be opened.

```
LL  NBJ.INSTALL.JCLLIB(ZZSI*)                Members.
LL  NBJ.COBOL.CPYBOOK(%%CVX*)
```

```
LL  DEV.OEM.CBL202.CBLI.HELP.HTML(S*AN%  WIN*, *R)

LL  NBJ.GENS05.ASM.SOURCE(APEZJUMP.<=-1)     Member Generations.
LL  NBJ.GENS05.ASM.SOURCE(APE*.=-1)
LL  NBJ.GENS05.ASM.SOURCE(APE*.<-2   CNV*.<-2   EDT*.<-2)
```

◊ For **z/VSE** the library parameter can be:

1. A library name. In this case the statistics for the library are listed. e.g.

   ```
   LL  CBLLIB
   ```

2. A library name and sublibrary name. In this case the sublibrary name may be a mask containing "*" (asterisk) wild cards as supported by VSE Librarian. The statistics for all sublibraries which fit the sublibrary name mask are listed. e.g.

   ```
   LL  CBLLIB.TEST*
   ```

3. A library name, sublibrary name and member name and type. In this case the member name and type may be a mask containing "*" (asterisk) wild cards. The statistics for all members which fit the mask are listed. e.g.

   ```
   LL  CBLLIB.TEST01.*.Z
   ```

Examples:

`LL  CBL.JCL`
List all members of the CBL.JCL PDS.

`LL  CBL.PDSE2G10.JCL(U01* U05%%%A1 U05%%%A5)`
Open the Library List window to display only prime member generations of the CBL.PDSE2G10.JCL PDSE version 2 library (MAXGENS 10) that match the specified member masks.

`LL  CBL.PDSE2G10.JCL(U01*.<-5  U05%%%A1.<-5  U05%%%A5.<-5)`
Open the Library Member Generations List window to display only those member generations whose member name matches one of the member masks and whose relative generation number is less than -5.

`LL  PRD2.CBL.*.*`
List all members of the PRD2.CBL LIBR sublibrary.

`LL  PRD2.*`
List all LIBR sublibraries of PRD2.

LLS

Syntax:

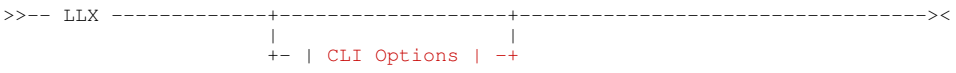
```
>>---- LLS ------------------------------------------------------------------><
```

Description:

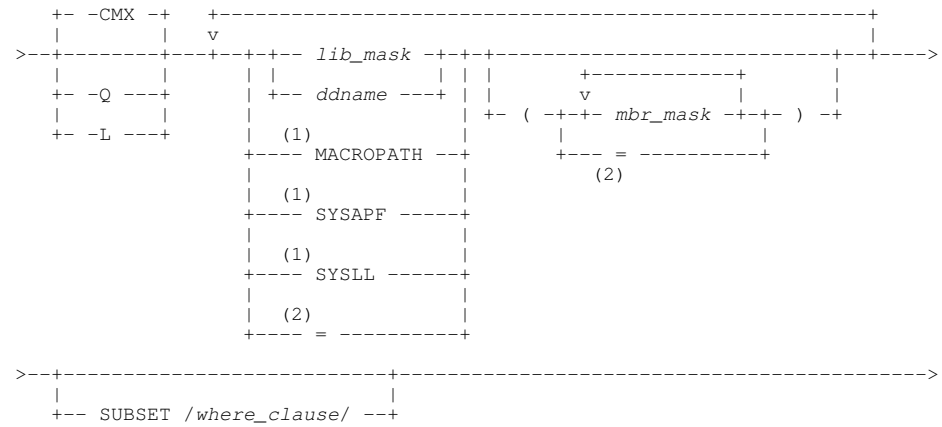
For SELCOPYi on z/OS only, LLS will open the List Loaded Structures panel to view and manage SDE structures that are currently loaded in storage.

LLX

Syntax:



CLI Options:



Notes:

1. Parameters MACROPATH, SYSAPF and/or SYSLL should be specified once only. If repeated the search will be repeated for all libraries in that concatenation.
2. "=" may be used if it is not the first library specification (*lib_mask*, *ddname*, etc.) or *mbr_mask* group referenced by the LLX command.

Description:

The LLX command invokes the Search for Library Members utility to locate members in PDS/PDSE libraries whose member names match the specified member mask(s), or, if executed with no parameters, opens the Search for Library Members Panel.

Output may be to a temporary CMX command file, which gets automatically displayed in a SELCOPY/i CBLe text edit window view, or multiple List Library Members windows, one for each library searched.

Parameters:

-CMX

Specifies that output format is an unsaved temporary file of DSN '%user%.LLX.Dyyyyddd.Thhmmss.TXT', with all information gathered in command file (CMX) format suitable for ACTION point-and-shoot <F16> edit of selected members.

This output format is default.

-Q

The -Q (Quiet) option implies -CMX, but suppresses pop-up windows that prompt the user for a decision to continue the search when either the number of members found in a library is greater than 999, or no members have been found in the last 10 libraries.

-L

The -L (List) option specifies that output format is multiple List Library Members windows. This option executes more quickly than -CMX (or -Q) but can result in a large number of open SELCOPY/i list window views.

lib_mask

Specifies a library DSN mask which identifies one or more PDS/PDSE libraries in which to search for the specified *mbr_mask* member mask(s).

lib_mask may optionally contain the following wild card characters:

*	A single asterisk represents an entire DSN qualifier or zero or more characters within a DSN qualifier mask.
**	A double asterisk represents zero or more qualifiers A double asterisk must be preceded by or follow a dot (period) and may not be used within a single DSN qualifier mask.

`%` A single percent sign represents exactly one character within a DSN qualifier. (Up to 8 percent signs can be specified in each qualifier.)

If no *lib_mask* is specified, then at least one of *ddname*, MACROPATH, SYSAPF or SYSLL must be specified.

ddname
Specifies an existing MVS DDname which has been allocated to one or more PDS/PDSE library data set names. The library or library concatenation allocated to *ddname* will be searched for the specified *mbr_mask* member mask(s).

If no *ddname* is specified, then at least one of *lib_mask*, MACROPATH, SYSAPF or SYSLL must be specified.

mbr_mask
Specifies a member name mask which identifies one or more member names to be found. Multiple member masks may be entered on each specified *lib_mask*, *ddname* MACROPATH, SYSAPF and/or SYSLL parameter, constituting a member mask group.

mbr_mask may optionally contain the following wild card characters:

`*` A single asterisk represents an entire member name or zero or more characters within a member name mask.

`%` A single percent sign represents exactly one character within a member name mask. Up to 8 percent signs can be specified in each member name mask.

If specified, the member name mask must immediately follow the PDS/PDSE *lib_mask* or *ddname* and be enclosed in "( )" (parentheses). Multiple member name masks, all specified within the single set of parentheses, must be separated by one or more blanks and/or "," (commas).

The special character "=" (equals) may be used in place of a *mbr_mask* group in second and subsequent *mbr_mask* group specifications. This equates to be the *mbr_mask* group entered on the last *lib_mask*, *ddname*, MACROPATH, SYSAPF or SYSLL parameter.

Default is to locate **all** members in libraries identified by *lib_mask*, *ddname*, MACROPATH, SYSAPF or SYSLL.

MACROPATH
Indicates that all libraries in the user's current CBLe text editor macro path will be searched for the specified *mbr_mask* member mask(s).

This list of libraries may be displayed using the CBLe text edit command, QUERY MACROPATH.)

SYSAPF
Indicates that all APF authorised load libraries will be searched for the specified *mbr_mask* member mask(s).

This list of libraries may be displayed using the APF List Window (command SYSAPF.)

SYSLL
Indicates that all load libraries in the active Link List concatenation will be searched for the specified *mbr_mask* member mask(s).

This list of libraries may be displayed using the Link List Window (command SYSLL.)

=
The special character "=" (equals) may represent a library specification (*lib_mask*, *ddname*, MACROPATH, SYSAPF or SYSLL) or a complete *mbr_mask* group.

"=" may be used in this capacity only if it is not the first library specification or *mbr_mask* group entered in the LLX command.

If used as a library specification, it is substituted with the previous library specification entered in the LLX command. If used as a *mbr_mask* group, it is substituted with the *mbr_mask* group belonging to the previous library specification entered in the LLX command.

SUBSET */where_clause/*
The SUBSET parameter specifies a list window WHERE Clause (*where_clause*) used to apply additional search criteria on matching library member names.

The *where_clause* supports filter criteria only on field names returned by List Library Members windows for MVS load libraries and non-load libraries. See *"List Library Members"* for details of these field names, descriptions and their data types.

Examples:

```
LLX   SYSEXEC(CBLII)  SYSPROC(=)  JGE.EXEC(=)
```
Search all libraries in the SYSEXEC and SYSPROC concatenations, as well as library 'JGE.EXEC', for member name "CBLII".

`LLX -Q  CBL.JCL(SEL*)  LAC.JCL(*)`
Search library 'CBL.JCL' for all members beginning "SEL", and library 'LAC.JCL' for all members. The user will not be prompted to continue the search as a result of encountering > 999 member name matches in either library.

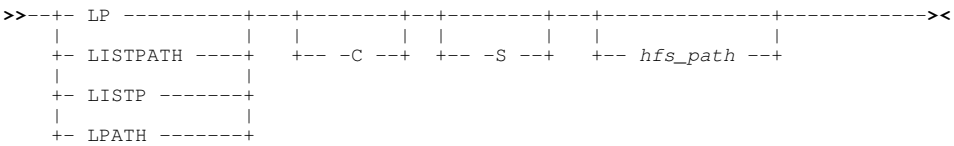
`LLX  CBL.**.JCL  SUBSET /WHERE LASTMOD => '2011/06/01'/`
Search all libraries with DSN matching the library DSN mask 'CBL.**.JCL' reporting all members that have been altered on or after 2011/06/01.

`LLX  STEPLIB(CBL*)  SUBSET /WHERE AC=1/`
Search all load libraries in the library concatenation allocated to DDname STEPLIB for module names beginning "CBL" that have been Link Edited with AC(1).

`LLX -L  NBJ.**.JCL`
Open a List Library Members window, one each for every library matching the DSN mask 'NBJ.**.JCL', which lists all members in that library.

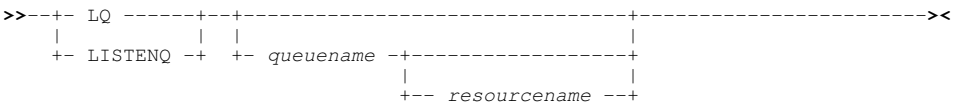
LP

Syntax:


>>--+- LP ----------+---+--------+--+--------+---+--------------+-----------><
| | | |

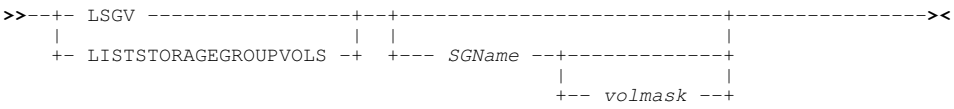
LQ

Syntax:


>>--+- LQ ------+--+--------------------------------+------------------------><
| | | |

LSGV

Syntax:



Description:

Use the LSGV command to open an SMS Storage Group Volumes list window and optionally specify storage group name and volume id mask. The list displays volumes belonging to an SMS pool storage group.

The SMS Storage Group Volumes List window may also be opened via the 'StorGrp Vols' item of the List Menu (=3.14) or on selecting 'SMS StorGrp Vols' from the Utilities/List menu in the CBLe main window menu bar.

Parameters:

SGName
The 1-8 character SMS Storage Group name. This parameter is placed in the **SMS Storage Group>** field of the SMS Storage Group Volumes List window.

volmask
Specifies a volume id mask. The mask supports the following wild cards:

*	A single asterisk represents zero or more characters.
%	A single percent sign represents a single character.

By default, a volume id mask that is less than 6 characters in length and does not contain an * (asterisk) wild card will be treated as having an implied trailing * wild card.

This parameter is placed in the **SMS Volume serial pattern>** field of the SMS Storage Group Volumes List window.

See Also:

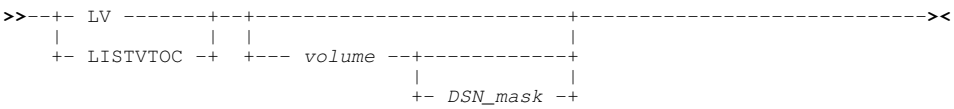
LSG Command

Examples:

```
LSGV   CBLDB2
```
List all volumes belonging to the SMS pool storage group "CBLDB2".

LV

Syntax:



Description:

Use the LV (List VTOC Files) command to open a VTOC File List window and optionally list, by data set name, entries contained in a DASD volume's Volume Table of Contents (VTOC).

The VTOC File List window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Note: Not implemented for CMS.

Parameters:

volume
The 1-6 character volume id containing the required VTOC.

This parameter is placed in the Volume field of the VTOC File List window.

DSN mask

Note: The DSN mask parameter is not yet implemented for VSE.

Select only catalog entries that match the specified dataset name mask. The dataset name mask supports the following wild cards:

- `*` A single asterisk indicates that either a qualifier or one or more characters within a qualifier can occupy that position. An asterisk can precede or follow a set of characters.
- `**` A double asterisk indicates that zero or more qualifiers can occupy that position. A double asterisk cannot precede or follow any characters; it must be preceded or followed by either a dot or a blank.
- `%` single percent sign indicates that exactly one character can occupy that position. (Up to 8 percent signs can be specified in each qualifier.)

A mask that does not contain an * (asterisk) wild card will be appended with *.** to list those data sets whose names begin with the mask string.

This parameter is placed in the DSN mask field of the VTOC File List window.

See Also:

LX Command
List VTOC entries by Extent.

Examples:

```
LV   CBLM02  CBL.SELC%%%.*
LV   CBLM02  SYS%.AX*.A*B*.**
```

LVOL

Syntax:

```
>>--+- LVOL ----+--+-------------+-------------------------------------------><
    |           |  |             |
    +- LISTVOL -+  +--- volume --+
```

Description:

Use the LVOL (List Volumes) command to open a DASD Volumes List window and optionally display the attributes of selected DASD volumes defined to your system.

The DASD Volumes window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Parameters:

volume

Select only defined DASD volumes that match the specified volume id mask. The volume mask supports the following wild cards:

- `*` An asterisk indicates that one or more characters within the volume id can occupy that position. An asterisk can precede or follow a set of characters.
- `%` A single percent sign indicates that exactly one character can occupy that position. (Up to 6 percent signs can be specified.)

A volume field that does not contain an * (asterisk) wild card will be appended with "*" to list all DASD volumes whose volume ids begin with the volume string.

This parameter is placed in the Volume field of the DASD Volumes window.

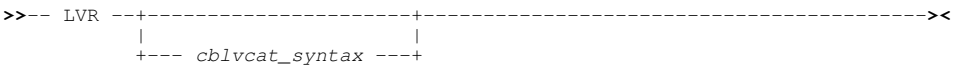
Examples:

`LVOL *`
List all volumes.

`LVOL SYS%%A`
List all volumes with 6 character volume name beginning with the characters 'SYS' and ending with 'A'.

LVR

Syntax:



Description:

LVR opens the CBLVCAT Raw window and optionally executes CBLVCAT control statements.
The CBLVCAT Raw window may also be opened via the "Raw" menu item of the Execute CBLVCAT window.

Parameters:

cblvcat_syntax
Valid CBLVCAT syntax to be executed when the CBLVCAT Raw window is opened.
This parameter is placed in the "VCAT command line>" field of the CBLVCAT Raw window.

See Also:

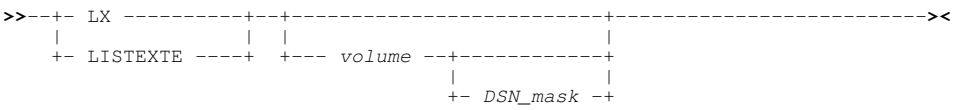
VCAT Command

Examples:

```
LVR   listvcat key=nbj type=c
LVR   listvtoc vol=cblmct
```

LX

Syntax:



Description:

Use the LX (List VTOC Extents) command to open a VTOC Extent List window and optionally list, by physical extent, the entries contained in a DASD volume's Volume Table of Contents (VTOC).

The VTOC Extent List contains an entry for each extent on the volume, including free extents and volume control areas such as the VTOC and the label area.

The VTOC Extent List window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Note: Not implemented for CMS or VSE.

Parameters:

volume
The 1-6 character volume id containing the required VTOC.

This parameter is placed in the Volume field of the VTOC Extent List window.

DSN mask
Note: The DSN mask parameter is not yet implemented for VSE.

Select only catalog entries that match the specified dataset name mask. The dataset name mask supports the following wild cards:

- `*` A single asterisk indicates that either a qualifier or one or more characters within a qualifier can occupy that position. An asterisk can precede or follow a set of characters.
- `**` A double asterisk indicates that zero or more qualifiers can occupy that position. A double asterisk cannot precede or follow any characters; it must be preceded or followed by either a dot or a blank.

`%` single percent sign indicates that exactly one character can occupy that position. (Up to 8 percent signs can be specified in each qualifier.)

A mask that does not contain an * (asterisk) wild card will be appended with *.** to list those data sets whose names begin with the mask string.

This parameter is placed in the DSN mask field of the VTOC Extent List window.

See Also:

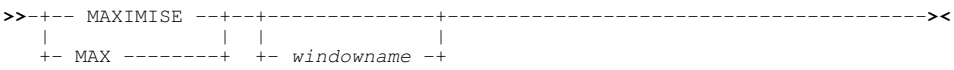
LV Command
List VTOC entries by Data Set Name.

Examples:

```
LX   CBLM01
```
List extents on volume id CBLM01.

MAXIMISE

Syntax:


>>-+-- MAXIMISE --+--+--------------+---><
| | | |

MDIPREV

Syntax:

```
>>--+- MDIPREV -----------------+-------------------------------------------><
    |                           |
    +- Window - (minus-sign) --+
```

Description:

For use in MDI applications only (e.g. CBLe and SELCOPY Debug), this command sets the focus window to be the previous MDI child window in the ring of MDI child windows.

The MDI application's ring of child windows is maintained in creation sequence and wraps round from the first created to the last created.

MERGE

Syntax:

```
>>- MERGE --+-----------------------+-------------------------------------><
            |                       |
            +-- | Merge Options | --+


Merge Options:

 |--+- Foreground -+-- KEYLEN keylen -- KEYPOS keypos --------------------->
    |              |
    +- Batch ------+

                                        +----------------+
                                        v                |
 >--- OUTFILE out_fileid ------ INFILES -+-- in_fileid --+----------------|
```

Description:

MERGE is used to generate SELCOPY batch control statements to merge multiple datasets that have been sorted by a specified key position and length, into a single output file.

Records from each of the specified input datasets are merged so that they are in ascending order of key value when written to the output dataset. If the key length (KEYLEN) and position (KEYPOS) values are set to 0 (zero), then the input datasets are simply concatenated and written to the output data set in the order they are specified.

If specified with no parameters, MERGE will open the Merge Datasets utility panel.

Parameters:

`FOREGROUND | BATCH`
: Selects whether the utility is to be run immediately in the foreground or will generate a batch job suitable for submission to batch.

  If BATCH is selected, the job statement and SYSOUT class defined by the JCL Information for generated Batch Jobs setting panel will be used to build the JCL statements. The generated job is then displayed in a text edit view.

`KEYLEN` *`keylen`*
: Specifies the length of the key field within the input data records.

`KEYPOS` *`keypos`*
: Specifies the position of the key field within the input data records. Note that this value is **not** an offset, as used for IDCAMS KSDS definition.

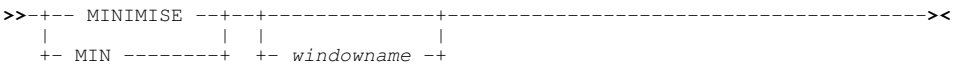
`OUTFILE` *`out_fileid`*
: The dataset name of the PDS/PDSE library member or a GDG generation, physical sequential or VSAM dataset to which all input records will be written.

`INFILES` *`in_fileid`*
: Specifies up to 10 input datasets. These may be any mixture of PDS/PDSE library members, GDG generation datasets, physical sequential datasets and VSAM datasets.

  Note that input datasets are presumed to have been sorted in ascending order key sequence, where the key field is the same as that defined by KEYPOS and KEYLEN.

MINIMISE

Syntax:



Description:

This command minimises the specified window.

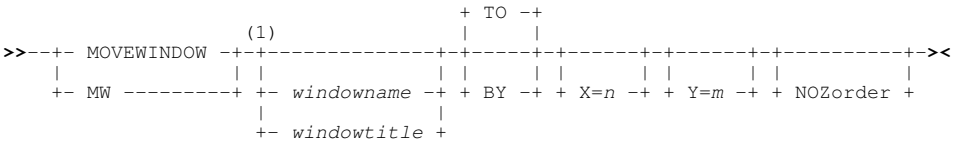
This command is equivalent to selecting the Minimise Button of the window to be minimised.

Parameters:

windowname
: The window name of the window to minimise. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

MOVEWINDOW

Syntax:



Notes:

1. MOVEWINDOW with no parameters displays this help.

Description:

Use MOVEWINDOW to move the specified window to an absolute X,Y coordinate or to an X,Y coordinate relative to its current position. Coordinates 0,0 reference the top left corner of the desktop.

The current position of a window is defined to be the X,Y coordinate of the top left corner of the window.

Note that it is not possible to move a window completely outside the SELCOPY/i main window. At least one column of the title bar must be viewable within the desktop. Therefore, limits governed by the currnet window position and the 3270 session screen size are imposed on the X,Y values supplied on the MOVEWINDOW command. Specifying X,Y values that fall outside these limits result in the limit value being used.

A window may also be moved by dragging then dropping the window using the titlebar. Similarly, the default function key table for the TitleBar level contains MOVEWINDOW commands allocated to PFkeys F7, F8, F10 and F11 to position the window up and down 1 line and left and right 1 column respectively.

Parameters:

windowname
: The window name of the window to be moved. *windowname* is not case sensitive. If a matching window name is not found, the search for a matching window is repeated for *windowtitle*.

If neither *windowname* nor *windowtitle* is supplied, the focus window is assumed.

windowtitle
: A character string which is a substring of the window title of the window to be moved. *windowtitle* is not case sensitive and may occur at any offset within the window title.

If *windowtitle* contains blanks then it must be enclosed in single- or double-quotes.

If *windowtitle* matches the window name of an already open window, then it is treated as a being *windowname* and that window is selected.

If *windowtitle* is found within the window title of more than one window, then the window selected is the one in which *windowtitle* occurs at the lowest offset within the title string. If *windowtitle* occurs at the same lowest offset in more than

one window title, then the window that occurs next in the window chain is selected.

windowtitle may **not** match any of the parameter keywords supported by MOVEWINDOW.

`TO`
Indicate that an absolute X,Y position follows.

`BY`
Indicate that a relative X,Y position follows.

`X=`*n*
Define the horizontal (column) coordinate.

If absolute, *n* must be a positive integer. The window will be moved horizontally so that the top left corner of the window is in column *n*.

If relative, *n* is an integer that may be prefixed by "+" (plus) or "-" (minus) to indicate a positive or negative horizontal displacement. The window will be moved horizontally *n* columns to the right (positive) or left (negative) from its current position.

There is no default value for *n*. If omitted, the horizontal coordinate is unchanged.

`Y=`*m*
Define the vertical (row) coordinate.

If absolute, *m* must be a positive integer. The window will be moved vertically so that the top left corner of the window is in row *m*.

If relative, *m* is an integer that may be prefixed by "+" (plus) or "-" (minus) to indicate a positive or negative vertical displacement. The window will be moved vertically *m* rows downwards (positive) or upwards (negative) from its current position.

There is no default value for *m*. If omitted, the vertical coordinate is unchanged.

`NOZORDER`
Indicates that the window focus is to remain unchanged following the MOVEWINDOW operation.

Examples:

```
MOVEWINDOW  TO X=32
```
Window is moved to column 32, row 1 (Y=1 is default).

```
MW EDTWEDIT2  TO X=2 Y=5
```
Window with window name EDTWEDIT2 (a text edit view) is moved to column 2, row 5.

```
MW CBL.JCL(SQ11740)  TO X=9 Y=5
```
Window with window title CBL.JCL(SQ11740) is moved to column 9, row 5.

```
MW TO X=32 Y=80
```
Window is moved to column 32, row 80. However, if row 80 is outside the 3270 display, then the window is moved so that the title bar is displayed in the last visible row.

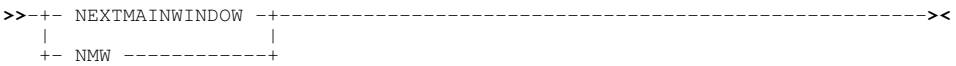
```
MW BY X=2 Y=-5
```
Window is moved 2 columns to the right and 5 columns upwards.

See Also:

SIZEWINDOW

NEXTMAINWINDOW

Syntax:



Description:

This command sets the focus window to the next main window i.e. one that is an immediate child of the desktop window. e.g. instances of CBLe, SELCOPY Debug, CBLVCAT Interactive or any list windows/dialogs created directly from the desktop menu bar or command line.

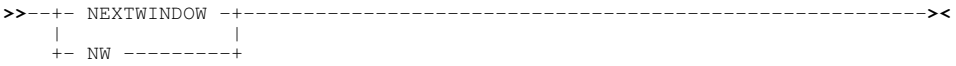
The ring is maintained in creation sequence and wraps round from the first created to the last created.

See Also:

PREVMAINWINDOW NEXTWINDOW PREVWINDOW MDINEXT MDIPREV

NEXTWINDOW

Syntax:



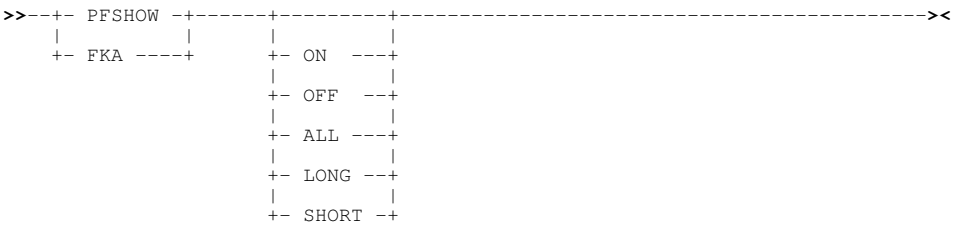
Description:

This command sets the focus window to the next window in the ring of all windows.

The ring is maintained in creation sequence and wraps round from the last created to the first created.

PFSHOW (FKA)

Syntax:



Description:

Use PFSHOW to control the function key display area at the bottom of the screen. Just like the standard ISPF version of this command, PFSHOW (or FKA) without parameters will toggle between each display mode.

The parameters ON, OFF, LONG and SHORT behave in the same way as the standard ISPF version.

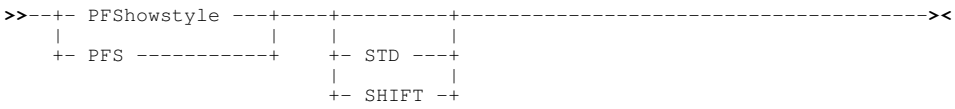
The **ALL** option is provided in addition to those supported by standard ISPF, and displays keys that are unset as well as those that are.

The benefit of this being that on a screen that is 160 wide, function keys 1-12 are always displayed on the first line, with the corresponding "Shift" keys (F13-F24) directly underneath.

See also PFSHOWSTYLE which controls the way function keys 13-24 are displayed.

PFSHOWSTYLE (PFS)

Syntax:



Description:

Use **PFSHOWSTYLE** to control the display of Function Keys 13-24, which are normally assigned to your PC keyboard as **Shift**-F1 to **Shift**-F12. PFS without parameters toggles between the two modes, SHIFT and STD.

Parameters:

STD
As per standard ISPF, Function Keys 13-24 are displayed as **F**13 to **F**24.

SHIFT
Within SELCOPY/i only, lower-case "**s**" is used to denote the **shift** key:

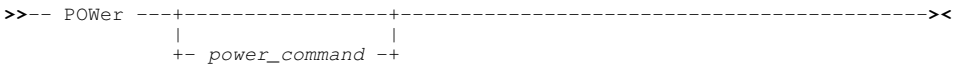
◊ Function Key 13 is displayed as **s**1
◊ Function Key 14 is displayed as **s**2
◊ Function Key 15 is displayed as **s**3
◊ Function Key 16 is displayed as **s**4
◊ Function Key 17 is displayed as **s**5
◊ Function Key 18 is displayed as **s**6
◊ Function Key 19 is displayed as **s**7
◊ Function Key 20 is displayed as **s**8
◊ Function Key 21 is displayed as **s**9
◊ Function Key 22 is displayed as **s**10
◊ Function Key 23 is displayed as **s**11
◊ Function Key 24 is displayed as **s**12

For example (with PFSHOW ALL in effect) ...

```
F1=HELP      F2=SPLIT     F3=END       F4=WINDOW    F5=RFIND     F6=RCHANGE
F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=CRETRIEV
s1=InsLine   s2=DelLine   s3=DupLine   s4=ACTION    s5=MrkBox    s6=MrkLine
s7=SPLTJOIN  s8=BoxFuncs  s9=SwapList s10=UNDO     s11=REDO     s12=ResetBox
```

POWER

Syntax:



Description:

For VSE only, use the POWER command to open a POWER Command Output window and optionally execute a VSE POWER command.

The Power Command Output window may also be opened via the File menu of the SELCOPY/i main window menu bar.

If SELCOPY/i INI variables System.VSESMLogon=No (i.e. no Security Manager is active) and System.TrustedUser=No, then POWER commands are restricted to PDISPLAY operations only.

Parameters:

power_command
Any supported VSE POWER command.

This parameter is placed in the POWER Command field of the POWER Command Output window.

Note that some POWER commands are not supported for cross partition usage (e.g. PDISPLAY STATUS)

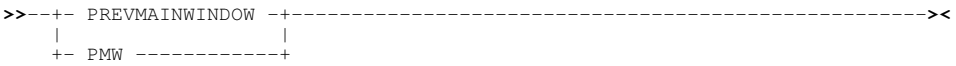
Examples:

```
POWER D LST
```
Display the POWER list queue.

```
POW  PRELEASE RDR,CBLTEST
```
Release entry CBLTEST from the POWER reader queue.

PREVMAINWINDOW

Syntax:



Description:

This command sets the focus window to the previous main window i.e. one that is an immediate child of the desktop window, e.g. instances of CBLe, SELCOPY Debug, CBLVCAT Interactive or any list windows/dialogs created directly from the desktop menu bar or command line.

The ring is maintained in creation sequence and wraps round from the first created to the last created.

Because MDI applications such as the CBLe editor and SELCOPY Debug have many child windows of their own (navigable with WINDOW [+]/- commands), this command is necessary to switch directly between SELCOPY/i application windows.

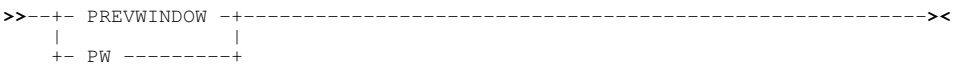
By default, F4 is set to **WINDOW**.

See Also:

NEXTMAINWINDOW NEXTWINDOW PREVWINDOW MDIPREV MDIPREV

PREVWINDOW

Syntax:



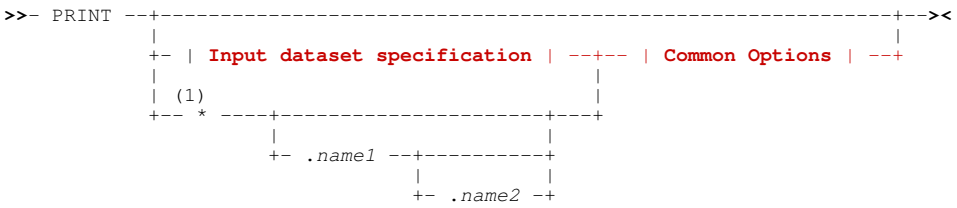
Description:

This command sets the focus window to the previous window in the ring of all windows.

The ring is maintained in creation sequence and wraps round from the first created to the last created.

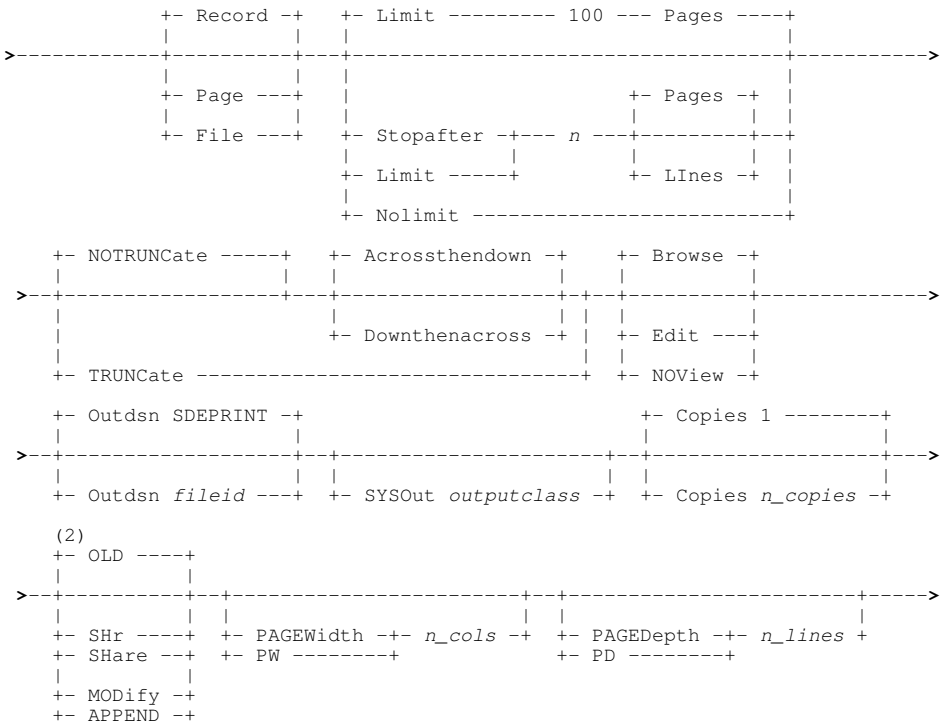
PRINT

Syntax:



Input dataset specification:

Common Options:



Notes:

1. Asterisk (*) is optional if any other parameter is specified.
2. Default disposition is SHR if OUTDSN is a PDS/PDSE library member.

Description:

PRINT is used to open the general Print File panel or to immediately print the data belonging to a specific data set or the current SDE view.

If asterisk (*), or any other parameter other than INDSN, is specified, then data in the current SDE edit or browse view will be printed. If no SDE view exists, the **Print File** panel is opened.

PRINT may be used to print a single page of data in the current SDE view (PAGE), all data within the current line in single-record view (RECORD), or all data within all displayed lines starting at the current line (FILE).

PRINT may be executed interactively or in batch as SDEIN input to the SELCOPY/i batch program (SDEAMAIN). In both environments, the INDSN *input_dataset_specification* may include the browse parameter, INITCMD, to execute data formatting

commands before the data is printed.

Whether or not output is to a SYSOUT data set, the dimensions of the printed SDE view are not determined by the actual SDE view display, but correspond to the page width and page depth values.

The print output page depth is defined as the value set by the PAGEDEPTH parameter or else the value set by the PAGEDEPTH option (default 60 lines). The page depth value includes the 5 Print header lines so that the number of lines of data printed will be 5 less than the PAGEDEPTH value.

The print output page width is defined as the value set by the PAGEWIDTH parameter. If PAGEWIDTH is not specified then, for output to SYSOUT data sets and HFS files only, the output page width is defined by the value set by the PAGEWIDTH option (default 133 columns). Otherwise, the output page width is equal to the file's maximum record length value.
Note that, for print of records in a single record view, page width is limited to a maximum of 255 columns.

Batch Execution:

Virtual SDE views may be opened in batch to edit or browse a data set or DB2 table, optionally using a structure to format the data. A sequence of standard SDE primary commands may then follow the EDIT or BROWSE command to format the data display and/or navigate, filter and/or alter data in the view before printing it.

Note that SDE options set during an interactive SDE edit/browse session are saved in the user's INI file when SELCOPY/i is closed (see SAVEOPTIONS). As part of SDEAMAIN (SELCOPY/i batch program) execution, the user INI file allocated to DDname ZZSUSERI is used to initialise the SDE environment. If ZZSUSERI is not allocated, the user INI file belonging to the RACF userid associated with the job execution is used. Therefore, SDE options that dictate the appearance of virtual SDE edit views are subject to any changes made by a user during an interactive SELCOPY/i SDE session.

To overcome this. it is recommended that a DDname ZZSUSERI be allocated to DUMMY and that all options governing display of the data are set explicitly within the SDEIN control statements, prior to executing PRINT.

Print Records in Single Record View:

A single record view print of only the current record is achieved using PRINT RECORD. Multiple records may be printed in single record view using PRINT FILE.

For PRINT FILE output of data in single record view, the display is scrolled down one page at a time (DOWN PAGE) to print all displayed fields. The display is scrolled right to display the first fields of the next visible record, table row or record segment and the process repeated until the print limit or End-of-Data is reached. Page numbers restart at page 1 for each new record, segment or table row printed in single record view.

Record data printed in single record view is restricted to a maximum page width of 255.

Print All Records in Multi Record View:

Print of formatted records displayed in a multi record view is supported whether visible records are assigned the same or different record-types.

For PRINT FILE output of data in multi record view, the display is scrolled down one page at a time (DOWN PAGE) to print all displayed records, table rows or record segments until the print limit or End-of-Data is reached. Furthermore, if NOTRUNCATE is active and the length of printed data lines exceeds the page width value, the print of data lines in the SDE view is also scrolled across to the right one page at a time (RIGHT ALL PAGE) until all data in the longest line has been printed.

The order in which these pages are printed (i.e print all pages scrolling down first or across first) is determined by specification of parameter ACROSSTHENDOWN or DOWNTHENACROSS. Note that DOWNTHENACROSS is supported only if all visible records are assigned the same record-type.

Because this type of print exhibits a two dimensional quality, page numbers assigned to multi record view pages printed using PRINT FILE are of the format *x.y*. *x*=1,2,3,... and represents the page number when scrolled down, *y*=1,2,3,... and represents the page number when scrolled across.

Default Output:

By default, PRINT output is written to DDname SDEPRINT.

When executed interactively, if SDEPRINT has not already been allocated and parameter SYSOUT is not specified, PRINT will allocate SDEPRINT to a DSN specified by the SDE Print File in the Batch Job Settings (=0.6) panel. If the default SDE Print File is not set, the default is DSN %user%.SDEPRINT.

In batch, SDEPRINT is output for all SDE commands (including other PRINT commands) that are run as part of the same SDEAMAIN execution. OUTDSN *fileid* may be specified to print the output to a specified DDname, sequential or VSAM DSN, PDS/PDSE library DSN and member name or HFS fileid.

Parameters:

Input dataset specification (INDSN)
Use of a PRINT input data set nominates a specific data set from which records are to be printed.

The PRINT input dataset may be specified explicitly in the command as the argument of the **INDSN** keyword. If the **INDSN** keyword is not present in the command then the contents of the current SDE structured dataset browse or edit view are used. If there is no current structured dataset the Print File panel is opened.

INDSN (*input_dataset_specification*)
The input dataset specification is in the form of a structured edit BROWSE command (the BROWSE command verb is not required) which must be enclosed in parentheses following the INDSN keyword.

BROWSE keyword options such as **FROM**, **FOR**, **FILTER** and **VIEW** may be specified to limit the records from the input dataset which will be copied to the output dataset.

*
Required only if no other PRINT parameters are specified in order to immediately print data from the current SDE view (using defaults) as opposed to opening the general Print File panel or the SDE PRINT File Panel as appropriate.

.name1
Applicable only to print of the current SDE view, *.name1* corresponds to a label name identifying the first record of a range of data records to be selected for print. The preceding "." (period/dot) is mandatory.
Default is .ZFIRST.

.name2
Applicable only to print of the current SDE view, *.name2* corresponds to a label name identifying the last record of a range of data records to be selected for print. The preceding "." (period/dot) is mandatory.
.name2 may occur on a line with a lower line sequence number than *.name1*. This is functionally equivalent to specifying *.name2* before *.name1* on the PRINT command.
Default is .ZLAST.

PAGE
Print a single page containing only the currently displayed data in the SDE view.

FILE
Print the currently displayed data in the SDE view and all data displayed in all lines that follow until the print limit or End-of-Data is reached.

RECORD
Print the current line only in single-record view.
RECORD is default.

{LIMIT *n* PAGES | LINES} | NOLIMIT
LIMIT (synonym STOPAFT) limits the amount of output printed whereas NOLIMIT (or LIMIT 0) imposes no limit on the printed output, the print operation ending when End-of-Data is reached.

The amount of print output may be limited by a number *n* of pages or lines represented by *n* PAGES or *n* LINES. Note that a line limit includes blank lines that are printed following data displayed in the virtual SDE view.
Default is LIMIT 100 PAGES.

TRUNCATE | NOTRUNCATE
Applicable to FILE print of data in a multi-record view only.

TRUNCATE indicates that record data that spans across more than one page is truncated so that only the first page of all records are printed scrolling downwards.

NOTRUNCATE indicates that record data that spans across more than one page is printed using one of the ACROSSTHENDOWN or DOWNTHENACROSS methods. (See below)

Default is NOTRUNCATE.

ACROSSTHENDOWN | DOWNTHENACROSS
Applicable to FILE print of data in a multi-record view only when NOTRUNCATE is selected, these parameters define the order in which pages are printed.

ACROSSTHENDOWN specifies that all pages containing data that span across more than one page are to be printed before scrolling down to print the lines of data displayed below. i.e. Page numbers are in the order 1.1, 1.2, 1.3,..., 2.1, 2.2, 2.3,... , 3.1, 3.2, 3.3,...

DOWNTHENACROSS is valid only if all visible records in the SDE view are of the same mapped record-type. It specifies that, by scrolling downwards, the leftmost display of all data lines are to be printed first before returning to the first view of the printed data, scrolling across (right) one page and once again printing all views scrolling downwards. This is repeated until all data in the longest line has been printed. i.e. Page numbers are in the order 1.1, 2.1, 3.1,..., 1.2, 2.2, 3.2,..., 1.3, 2.3, 3.3,...

BROWSE | EDIT | NOVIEW
Supported for interactive execution of PRINT to a non-SYSOUT data set only, these parameters determine whether or not (NOVIEW) the output print file is displayed in a window view following PRINT processing, and if so whether the file is opened for BROWSE or EDIT.

Default is BROWSE.

`OUTDSN fileid`
OUTDSN identifies the location of the printed output.

fileid may be a DDname, a sequential or VSAM data set name, PDS/PDSE library DSN and member name or an HFS fileid.

By default, *fileid* is DDname SDEPRINT which is the standard SDEAMAIN output destination which must already be allocated. See Default Output above for further details.

`SYSOUT outputclass`
Applicable only if OUTDSN is a DDname which is not already allocated. In this case, the specified DDname will be dynamically allocated to a system output (SYSOUT) data set.

If PRINT is run interactively without parameter OUTDSN, output is to DDname SDEPRINT. If SDEPRINT is not already allocated and SYSOUT *outputclass* is specified, SDEPRINT is allocated to a system file instead of to DSN specified by the Print DSN value in the Batch Job Settings (=0.6) panel.

outputclass is a single alpha-numeric character or "*" (asterisk) representing the data set output class used on allocation of the specified SYSOUT DDname.

`COPIES n_copies`
Applicable only if OUTDSN is **not** SDEPRINT but is a DDname which is not already allocated. *n_copies* is an integer value between 1 and 255 specifying number of printed copies to be defined on allocation of the specified SYSOUT DDname.
Default is COPIES 1.

`OLD | SHR | SHARE | MODIFY | APPEND`
Applicable only if OUTDSN specifies the DSN of an existing sequential, VSAM or PDS/PDSE library data set. One of these parameters may be specified as the disposition used on allocation of the data set.

OLD	Exclusive, unshared ENQ for overwrite of any existing file data.
SHR | **SHARE**	Shared ENQ for overwrite of any existing file data.
MODIFY | **APPEND**	Exclusive, unshared ENQ for appending output to any existing file data.

Default for sequential and VSAM data sets is OLD.
Default for PDS/PDSE library data sets is SHR.

`PAGEWIDTH | PW n_cols`
Set the print output page width to *n_cols* columns.
Default for SYSOUT data sets and HFS files is the value set by option PAGEWIDTH, otherwise the default is the maximum record length of the output file.

`PAGEDEPTH | PD n_lines`
Set the print output page depth to *n_lines* lines.
Default is the value set by option PAGEDEPTH.

See Also:

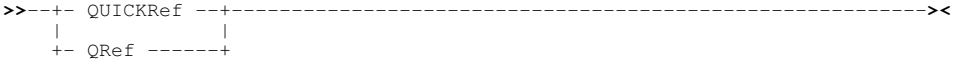
PAGEDEPTH PAGEWIDTH LEFT RIGHT

PUTXML

Executes the Data Editor primary command, PUTXML to copy an XML document from a data set, library member or HFS/ZFS file to an XML column entry of a DB2 table.

QUICKREF

Syntax:



Description:

Display the SELCOPY/i Quick Reference help.

QUIT

Syntax:

```
>>-- QUIT ------------------------------------------------------------------><
```

Description:

Use QUIT to exit and close the current SELCOPYi window.

If the current window is the SELCOPYi main window, then a pop-up window prompts the user to confirm whether or not to quit the SELCOPYi session.

RECOVER

Syntax:

```
>>- RECOVER ---- libname(mbrname.generation) --------------------------------><
```

Description:

Recover a previous generation of a PDSE version 2 member so that it becomes the prime member copy.

For z/OS PDSE version 2 libraries allocated a MAXGENS value, multiple versions of a member's data may exist in the library. By default, every time a member is saved, a new generation of the member data is written to DASD leaving the previous generation's data intact. At any time a previous generation of the member may be recovered using RECOVER command.

The prime member generation before the recovery takes place becomes the next previous generation (i.e. relative generation -1) after RECOVER has executed.

Parameters:

libname
: The complete dataset name of a PDSE version 2 library containing the member generation.

mbrname
: The name of the member for which a previous generation is to be recovered.

generation
: The absolute or relative value of the generation to be recovered.

Examples:

`RECOVER CBL.SELCOPYI.INIT.JCLLIB(INSPRE01.-2)`
: Recover relative generation -2 of JCL member INSPRE01.

RENAME

Syntax:

Rename an MVS data set, HFS file or PDS(E) member, or a CMS file on an accessed minidisk:

```
>>-- REName ------------------------ fileid1 --------- fileid2 --------------><
```

Rename a VSE sequential or VSAM file:

```
>>-- REName ----+- volid ---+-- : -- fileid1 --------- fileid2 --------------><
                |           |
                +- catdsn --+
```

Description:

Rename a single sequential DASD file, HFS file, PDS(E) member or VSAM file.

To succeed, the user must have sufficient read/write authority for the file and no exclusive ENQ or LOCK should already exist for the file.

In an MVS environment, when renaming a PDS member, parameters should be specified in one of the following formats:

1. fileid1 is the data set and member name of the member to be renamed and fileid2 is the new member name only.
2. fileid1 is the **quoted** data set and member name for the member to be renamed and fileid2 is the **quoted** data set and new member name.

This second method also applies to MVS sequential and VSAM data sets whereupon RENAME executes the following IDCAMS command:

```
   ALTER  fileid1  NEWNAME(fileid2)
```

Therefore, types of file that may be renamed and the supported format of fileid1/2 is governed by the IDCAMS ALTER command. (See "DFSMS Access Method Services for Catalogs".)

For HFS files, specification of a leading "." (dot/period) or "/" (slash) in the HFS path name is mandatory in order to distinguish it from an MVS data set name. Both fileid1 and fileid2 may be specified as an absolute or relative HFS path and may reference a file name, directory name, hard link or symbolic link.

For VSE, sequential files may only be renamed if the CBL software product **CBLVCAT** is licensed. SELCOPY/i uses CBLVCAT's MOD operation to perform the rename.

Parameters:

volid
: For VSE sequential disk files, this is the volume serial number of the DASD volume on which the sequential file resides.

catdsn
: For VSE VSAM files, this is the full fileid of the VSAM catalog to which the VSAM managed file belongs.

fileid1
: The current fileid in full of the file to be renamed. For HFS, this may be an absolute or relative path name.

fileid2
: The new fileid to be assigned to the file. For HFS, this may be an absolute or relative path name.

Examples:

`rename  cbl.ssc.ctl(sstest)    sstest01`
: Rename an MVS PDS(E) member.

`rename "cbl.jcl(cblins01)"    "cbl.jcl(install)"`
: Rename an MVS PDS(E) member.

`ren     cbl.cbli.test.file     nbj.test.data`
: Rename an MVS sequentail or VSAM data set.

`rename   SYSWK1:CBL.SELCOPY.NAM    CBL.SELCOPY.NAM.NEWNAME`
: Rename a VSE sequential disk file. (CBLVCAT must be licensed.)

`rename   VSESP.USER.CATALOG:CBL.TEST.KSDS   CBL.TEST.KSDS.NEWNAME`
: Rename a VSE VSAM managed data set.

`rename   ./nbj.tmp.gz    /scr/install.x1832.gzip`
: Rename an HFS file and move it to a new directory.

REPORT

Examples:

The following examples are as they might appear in a plain text file (e.g. the user's HOME command centre file) suitable for execution using the ACTION key.

```
<REPORT RUN RPTDEF( USER123.SELCOPYI.RPT(T030SUM) )              \
          SMF-INPUT-BEG        \
             USER123.SMF030       \
             TYPES(30-5)          \
             DATELO( 2018/09/15 13:00) \
             DATEHI( 2018/09/20 ) \
          SMF-INPUT-END
```

Using the report definition saved in dataset "**USER123.SELCOPYI.RPT(T030SUM)**", produce a report from all **SMF Record-Type 30 SubType 5** records contained in dataset **USER123.SMF030** provided they fall within the DATELO/DATEHI timestamp range.

Sample Report Definition:

The **TO30SUM** member may contain the following report definition statements:

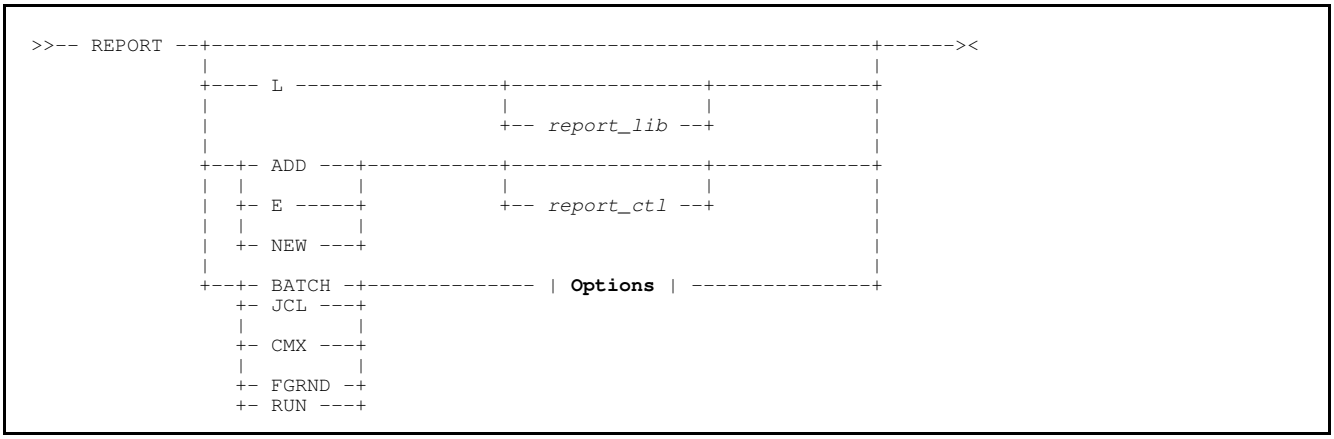
```
TITLE:
 Job/DD EXCPs Report (from SMF Type 30 Subtype 5)

COLUMNS:
 SMF030_Identification.zJOBNAME            'Job Name'
 SMF030_Identification.zSIT                'Job Start'
 SMF030_Common_Address_Space_Work.ZTME     'Job End'
 SMF030_EXCP.zDDN                          'DDName'
 SMF030_EXCP.zBLK                          'EXCP Blks'

REPEAT:
 SMF030_EXCP
```

Execution of the REPORT command would create a report output that looks something like the following:

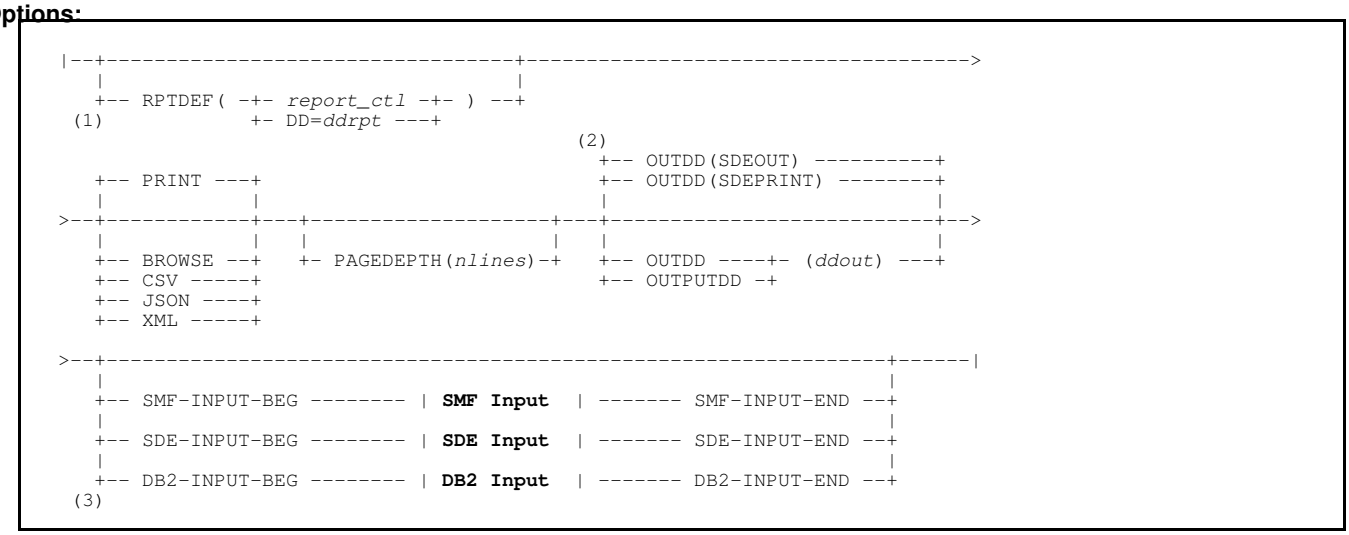
Sample Report Output:

```
12018/09/14 12:04   Job/DD EXCPs Report (from SMF Type 30 Subtype 5)  PAGE    1

 Job Name Job Start              Job End                DDName   EXCP Blks
 -------- ---------------------- ---------------------- -------- -----------
 SMFCLEAR 2018/09/04 01:09:18.03 2018/09/04 01:09:29.73 INDD1          14400
                                                        DUMPOUT           786
                                                        SYSPRINT            0
                                                        SYSIN               2
 SMFCLEAR 2018/09/04 02:55:15.33 2018/09/04 02:55:26.97 INDD1          14400
                                                        DUMPOUT           783
                                                        SYSPRINT            0
                                                        SYSIN               2
 SMFCLEAR 2018/09/04 04:41:32.55 2018/09/04 04:41:45.63 INDD1          14400
                                                        DUMPOUT           786
                                                        SYSPRINT            0
                                                        SYSIN               2
 SMFCLEAR 2018/09/04 06:27:59.30 2018/09/04 06:28:09.90 INDD1          14400
                                                        DUMPOUT           788
                                                        SYSPRINT            0
                                                        SYSIN               2
--------------------------- 6178 line(s) not displayed --------------------------
12018/09/14 12:04   Job/DD EXCPs Report (from SMF Type 30 Subtype 5)  PAGE  113

 Job Name Job Start              Job End                DDName   EXCP Blks
 -------- ---------------------- ---------------------- -------- -----------
 SMFCLEAR 2018/09/07 09:54:14.26 2018/09/07 09:54:25.94 INDD1          14400
                                                        DUMPOUT           796
                                                        SYSPRINT            0
                                                        SYSIN               2
 SMFCLEAR 2018/09/07 11:31:38.69 2018/09/07 11:31:51.49 INDD1          14400
                                                        DUMPOUT           792
                                                        SYSPRINT            0
                                                        SYSIN               2
 SMFCLEAR 2018/09/07 13:08:53.59 2018/09/07 13:09:05.39 INDD1          14400
                                                        DUMPOUT           786
                                                        SYSPRINT            0
                                                        SYSIN               2


                                                                  ===========
 == Grand Totals ( 5846 Items)                                      1139389
                                                                  ===========
```

Syntax:



Options:



(1) RPTDEF is mandatory when running in BATCH.

(2) OUTDD(SDEOUT) if SDEOUT allocated. Otherwise, OUTDD(SDEPRINT) if running in batch, or no OUTDD operand if running in foreground.

(3) SDE, DB2 or SMF input is mandatory unless running in the foreground and RUN is specified to generate a report on formatted data in the current Data Editor view.

Description:

REPORT provides a command-line interface (CLI) to the Report Utility panel.

A full description of the REPORT utility including working samples, panels and report definition control statement syntax may be found in the **SELCOPYi REPORT Utility** manual online at:

www.cbl.com/pdf/SELCOPYi_3.50_Report_Utility.pdf

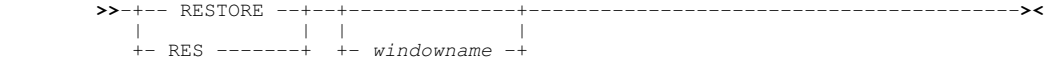
The report produced will typically consist mainly of data extracted from a list of data fields from a single Record-Type, but you may select fields from more than one record-type and even generate your own values based on meaningful calculations. The report may reference fields from both Primary (Base) and Secondary segments.

A user definable heading will be printed at the top of each page, followed by user definable column headings for each selected field.

Grand totals will automatically be printed for any selected field containing integer data, and sub-totals will also be printed if a sort/control-break has been requested.

RESTORE

Syntax:



Description:

This command restores the specified window from a maximised or minimised state back to its original size and position.

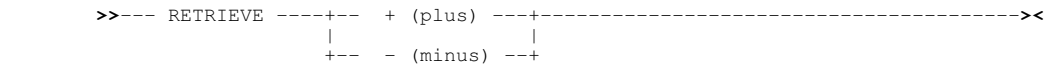
This command is equivalent to selecting the Restore Button of the window to be minimised.

Parameters:

windowname
: The window name of the window to maximise. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

RETRIEVE

Syntax:



Description:

For each window, SELCOPYi stores a history of the executed commands. RETRIEVE may be used to recall commands from the ring of executed commands, placing them at the command prompt of the focus window.

Parameters:

+
: Recall commands scrolling forwards through the ring.

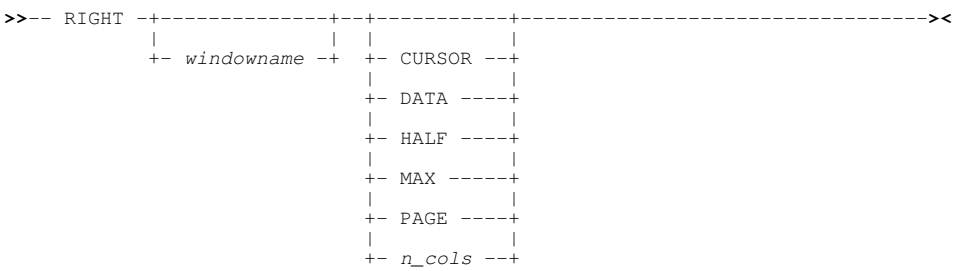
-
: Recall commands scrolling backwards through the ring.

See Also:

CRETRIEV

RIGHT

Syntax:



Description:

Scroll the view of the data within the specified window right towards the last column of the displayable data.

The extent by which data is scrolled is determined by the CURSOR, DATA, HALF, PAGE, MAX or *n_cols* parameter which may be specified using any one of three methods determined in the following order of precedence:

1. The scrolling command verb, RIGHT, and one of these scrolling parameters is explicitly specified on the command line.
2. The scrolling parameter is specified on the command line and a PFKey assigned to RIGHT is actioned.
   Note that the contents of a command line are appended to the command stream assigned to a PFKey when that PFKey is actioned.
3. No scrolling parameter is specified, so the current value of the "Scroll>" field is used.

4. No scrolling parameter is specified and no "Scroll>" field is present, so a defualt of one column is used.

List windows may contain fields that have **KEY** attribute **YES** defined in the Field Descriptor Block. Fields with this attribute are always in view and may not be scrolled right or left. If the cursor is positioned in a column belonging to this type of field, then, for LEFT CURSOR and RIGHT CURSOR, the cursor is considered to be outside the display area. All columns of data that do not belong to a **KEY** field are scrollable using LEFT and RIGHT.

By default this command is assigned to function key **PF11**.

Parameters:

windowname
The window name of the window in which the display is to be scrolled. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

CURSOR
The scrollable column on which the cursor is positioned becomes the first scrollable column of the display.
If the cursor is positioned outside the display area, in a KEY field or on the first scrollable column within the display area, then RIGHT PAGE is executed instead.

DATA
Scroll right so that the last scrollable column in the current display area becomes the first scrollable column of the display.

HALF
Scroll a number of columns so that the column situated half way along the width of the current display of scrollable columns, becomes the first scrollable column of the display.

MAX
Scroll right to display the last scrollable column of data.
Where the display area is able to contain all columns of data, the first scrollable column becomes the first scrollable column of the display. Otherwise, the last scrollable column of data becomes the last column of the scrolled display.

PAGE
Scroll right so that the column of data to the right of the last scrollable column in the current display, becomes the first scrollable column of the display.

n_cols
Scroll right a specified number of columns.
The column of data that is *n_cols* to the right of the first scrollable column becomes the new first scrollable column of the display.

SDATA

Syntax:

```
>>-- SData -- sde_command ------------------------------------------------------><
```

Description:

Direct a command to the SELCOPYi Structured Data Environment (SDE).

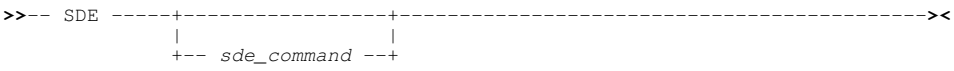
The SDATA command allows SDE commands to be issued from any SELCOPYi window.
If the CBLe text editor main window has been stopped, SDATA will start the CBLe main window and open an edit view for the user's HOME CMX file before executing the SDATA command.
Also see the SDATA CBLe CLI command.

Parameters:

sde_command
Any SDE command.

SDE

Syntax:



Description:

Performs the same action as SDATA except that, if no arguments are specified the Structured Edit dialog window is opened.

Parameters:

sde_command
Any SDE command.
Default is EDITDIALOG.

SDSF

Syntax:

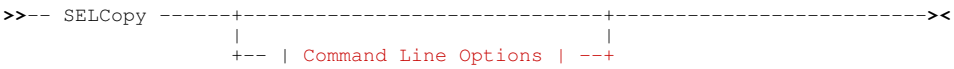
Description:

When running SELCOPY/i in TSO (with or without ISPF), the SELCOPY/i command **SDSF** starts the System Display and Search Facility.

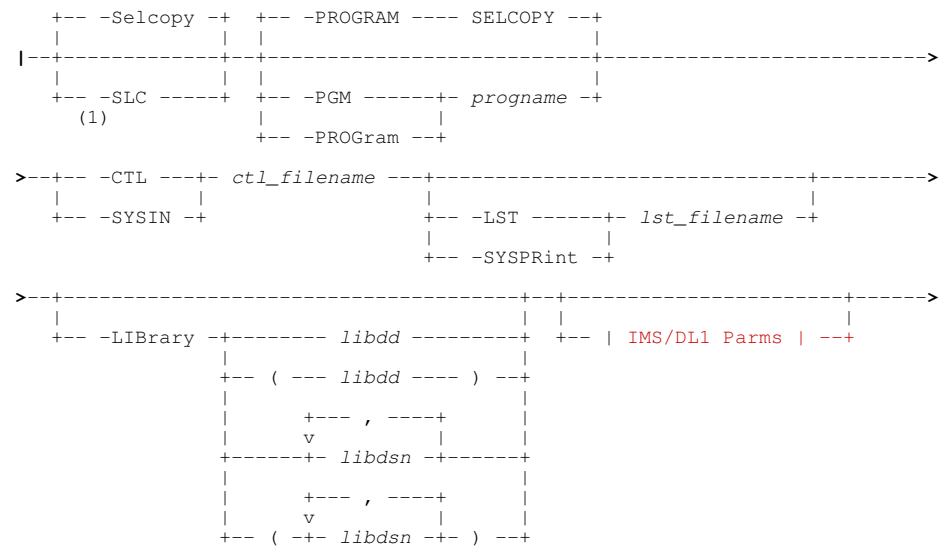
SDSF is started as a full screen TSO application which returns control to SELCOPYi only after it is closed. SDSF may also be started via the **Utilities** menu of the SELCOPYi main window menu bar.

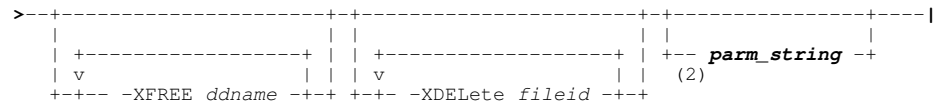
SELCOPY

Syntax:

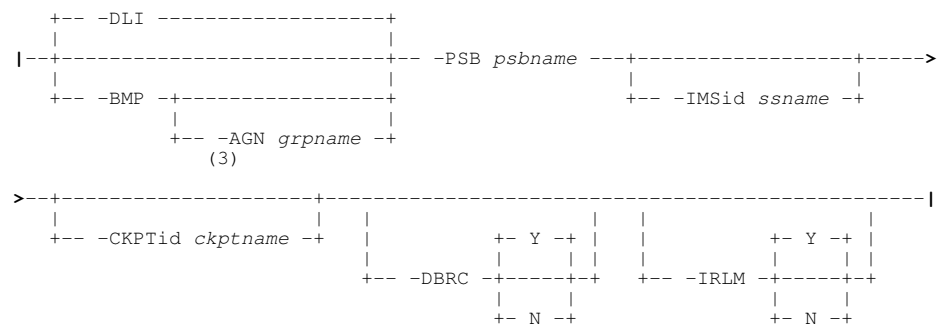


Command Line Options:





IMS/DL1 Parms:



Notes:

1. Parameter -SLC is default if -PROGRAM SLC is specified.
2. A SELCOPY/SLC paramater string, if present, must be specified after all keyword parameters.
3. Parameter -AGN is effective only for IMS 9.1 or below where AGN security is supported, otherwise the parameter is ignored.

Description:

Use the SELCOPY command to start the SELCOPY Debug & Development utility.

The utility supports debugging of SELCOPY source control statements using either the SELCOPY Assembler (SELCOPY batch) or the SELCOPY C++ (SLC batch) language interpreter.

If specified with no parameters, the SELCOPY/debug Menu is displayed prompting for either a SELCOPY SYSIN input file or JCL batch job input. Note that JCL input will be pre-processed by SELCOPY Debug to split the job into its constituent SELCOPY and non-SELCOPY job steps. Each job step may then be executed in isolation with the option of executing SELCOPY job steps using the debugger. If parameters are specified, SELCOPY debug starts immediately using the specified SYSIN file *ctl_filename* which contains the SELCOPY control statements.

SELCOPY Debug opens a text edit view for the *ctl_filename* SYSIN/SYSIPT source file input. If *ctl_filename* is supplied as an explicit fileid (DSN) which is not locked (with an exclusive ENQ) by another process, and the user has sufficient authority, the file is edited read-write. In this case, updates can be made to the SYSIN source and execution repeated without leaving the SELCOPY Debug application.

Where *ctl_filename* refers to a previously allocated filename (DD/FILEDEF/DLBL), which may be a concatenation of datasets, then it is opened read-only by the SELCOPY Debug SYSIN/SYSIPT text edit view. A token of 8 or fewer characters with no embedded dots is automatically treated an allocated DDname/file name.

Read-only access to SELCOPY control statements held in CA-LIBRARIAN members is made available on z/OS systems via SELCOPYi's ALLOC, which supports the SUBSYS(LAM) parameter.

See SELCOPY Debug & Development for full information on the features of the SELCOPY Debug application, including point-and-shoot options provided through the F16 key (SdbPopup).

Parameters:

`-SLC | -SELCOPY`
: Parameters -SLC and -SELCOPY specify the SELCOPY language interpreter to be used.

  Parameter -SLC identifies the program source as being for the SELCOPY C++ version and implies a default program name (-PGM) of SLC. Parameter -SELCOPY identifies the program source as being for the SELCOPY Assembler (BAL) version and implies a default program name of SELCOPY.

  Default is -SELCOPY unless -PROGRAM SLC is specified, in which case the default is -SLC.

`-PROGRAM` *`progname`*
`-PGM`
: Specifies *progname*, the name of the program load MODULE or PHASE to be used to process the control statements. This program name is used in place of the default program name implied by the selected SELCOPY language interpreter.

  Default program name is SELCOPY unless -SLC is specified, in which case the default is SLC.

-CTL *ctl_filename*
-SYSIN

Specifies *ctl_filename*, the full DSN (fileid) or allocated DDname/FILEDEF/DLBL identifying the file (z/OS data set, HFS file path or PDS/PDSE library member, CMS or VSE sequential file) containing the source control statements.

-LST *lst_filename*
-SYSPRINT

Currently performs no action. The -LST *lst_filename* parameter is reserved to identify an output file for records written to the SYSPRINT debug window view.

-LIBRARY *libdd | libdsn...*

For z/OS systems only, -LIBRARY identifies one or more load libraries to be included before the current environment's search library chain. This is equivalent to supplying a JCL STEPLIB statement in a batch job and so may be used to control which SELCOPY module is executed and also any routines executed via the SELCOPY CALL statement.

The parameter aguments may be enclosed in parentheses and are as follows:

- ◊ A single DDname *libdd* which has been allocated to one or more load libraries.
- ◊ One or more load library DSNs *libdsn* separated by ',' (comma), ';' (semi-colon) or blank characters. Note that if blank separators are used, the list of DSNs must be enclosed in '( )' (parentheses).

IMS/DL1 Parms

The SELCOPY Assembler program supports data management calls to IMS/DL1 data bases to perform segment I/O operations for offline (DLI) processing or via an IMS online batch message processing (BMP) region. Currently, the SLC program does not support calls to IMS/DL1.

If the SELCOPY control statement input includes operations that perform IMS/DL1 database I/O, then the SELCOPY program must be started as a subtask of the IMS/DL1 region controller (DFSRRC00). To achieve this using SELCOPY Debug, the -PSB parameter must be specified.

IMS/DL1 parameters specified on the SELCOPY primary command are passed as parameters to the program DFSRRC00.

-DLI
-BMP

Identifies the region as DLI (a DLI offline batch region) or BMP (an online batch region).
Default is -DLI.

-AGN *grpname*

Applicable only to online batch regions (-BMP) and for versions of IMS 9.1 and earlier where Security Maintenance utility and AGN security is supported. Parameter -AGN *grpname* identifies the application group name definition required to be able to successfully run the SELCOPY program against IMS online databases.

-PSB *psbname*

The Program Specification Block name to be used to process IMS/DL1 DB segments.

-IMSID *ssn*

Specifies a 4-character IMS region identifier that will override the identifier specified during system definition of the running IMS system. The IMS identifier is used in IMS messages that are written to the system log.

-CKPTID *ckptid*

Specifies an 8-character IMS symbolic checkpoint id which identifies the checkpoint at which the program will be re-started. The SELCOPY program must perform an extended restart (XRST) DL1 call immediately upon startup in order to recover the work area buffer.

-DBRC [Y | N]

Indicates whether or not IMS Database Recovery Control is to be used in the DLI batch region. (Note -DBRC is ignored for BMP regions.)

If -DBRC is specified with "Y" (or without "Y" or "N"), then DBRC will be used. If -DBRC is specified with "N", then DBRC will not be used.

If -DBRC is not specified, the local system default will be used. i.e. DBRC is used if DBRC=YES or FORCE is configured in the DFSIDEF0 module.

-IRLM [Y | N]

Indicates whether or not IMS Resource Lock Manager (with default name IRLM) is to be used in the DLI batch region. (Note -IRLM is ignored for BMP regions.)

If -IRLM is specified with "Y" (or without "Y" or "N"), then IRLM will be used. If -IRLM is specified with "N", then IRLM will not be used.

If -IRLM is not specified, the local system default will be used.

-XFREE *ddname*

For z/OS systems only, -XFREE specifies an allocated DDname which is to be freed when the SELCOPY Debug session ends.

-XDELETE *fileid*

For z/OS systems only, -XDELETE specifies the fileid (DSN) of a sequential or VSAM data set, library member or HFS file which is to be erased when the SELCOPY Debug session ends.

-XDELETE *fileid*

For z/OS systems only, -XDELETE specifies the fileid (DSN) of a sequential or VSAM data set, library member or HFS file which is to be erased when the SELCOPY Debug session ends.

parm_string
Specifies a parameter string, *parm_string*, which is to be passed to the SELCOPY or SLC program. The *parm_string* must follow all other parameter keywords and arguments on the SELCOPY primary command.

The parameter string arguments may be referenced in SELCOPY control statements via POS PARM. Additionally, for SLC program only, parameter string arguments may also be referenced via POS ARG and %*n* substitution variables.

A *parm_string* is ignored if IMS/DL1 parameters are specified.

Examples:

```
SELC     -CTL CBL.SSC.CTL(DIRD01)
```
Start the SELCOPY Debug application and use the SELCOPY progam to interpret and execute control statements loaded from the library member CBL.SSC.CTL(DIRD01).

```
SELC -SLC  -PGM SLC#330  -CTL CTLIN  -XFREE CTLIN  'John' 55
```
Start the SELCOPY Debug application and use program name SLC#330 to execute control statements interpreted using the SLC language interpreter and loaded from the file (data set, HFS file path or library member) allocated to CTLIN.

Parameter string "'John' 55" will be passed to the program and CTLIN will be freed when the Debug session ends.

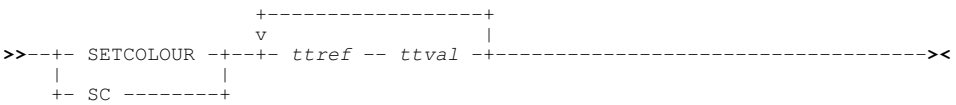
```
SELC  -CTL SYS3.CBL.SELCOPY.CTL001  -LIB (DEV.CBL.LOAD DEV.TEST.RTN001.LOAD)
```
Start the SELCOPY Debug application and use the SELCOPY progam to interpret and execute control statements loaded from file (DSN) SYS3.CBL.SELCOPY.CTL001.

Load library DEV.CBL.LOAD then DEV.TEST.RTN001.LOAD will be searched before libraries in the search chain when locating the SELCOPY program and any programs/load modules called in the control statements using the SELCOPY CALL operation.

```
<ALLOC  DD(INCTL)  DSN('SYS3.NBJ.EQU001'  'SYS3.TEST.SELCOPY.CTL(SQ10249)')  SHR
<SELCOPY  -CTL INCTL
```
In this example, the Text Editor primary command, ALLOCATE command is first used to allocate a concatenation of two DSNs to DDname, INCTL. These commands may be entered as text in a file (e.g. the user's home CMX file), edited or viewed using the Text Editor and executed using the ACTION facility (F16).

SETCOLOUR

Syntax:



Description:

Use SETCOLOUR to remap the appearence of a colour and its associated highlighting.

SELCOPY/i maintains a translate table that defines how a colour/highlight style combination is to be displayed. Each colour/highlight style combination may be remapped so that it is displayed as a different colour/highlight style.

A colour/highlight style combination is represented as a pair of descriptor characters, the first of which is the initial of the colour, and the second the initial of a highlight style. All valid combinations are as follow:

		2nd Char			
		B	**D**	**R**	**U**
1st Char	**B**	Blue Blink	Blue Default	Blue REVVideo	Blue Underline
	G	Green Blink	Green Default	Green REVVideo	Green Underline
	P	Pink Blink	Pink Default	Pink REVVideo	Pink Underline
	R	Red Blink	Red Default	Red REVVideo	Red Underline
	T	Turquoise Blink	Turquoise Default	Turquoise REVVideo	Turquoise Underline
	W	White Blink	White Default	White REVVideo	White Underline
	Y	Yellow Blink	Yellow Default	Yellow REVVideo	Yellow Underline

These descriptor pairs reference cells in the translate table as well as describing the values within a cell. The default translate table is such that each cell referenced by a colour/highlight combination contains that colour/highlight combination. i.e.

		2nd Char			
		B	**D**	**R**	**U**
1st Char	**B**	BB	BD	BR	BU
	G	GB	GD	GR	GU
	P	PB	PD	PR	PU
	R	RB	RD	RR	RU
	T	TB	TD	TR	TU
	W	WB	WD	WR	WU
	Y	YB	YD	YR	YU

A number of **ttref** and **ttval** pairs may be specified, each defining a single update to the translate table.

Parameters:

ttref
A colour/highlight style pair referencing a cell in the translate table.

ttval
A colour/highlight style pair to be inserted into the translate table cell referenced by *ttref*. Where SELCOPY/i uses the *ttref* colour/highlight style combination, the *ttval* colour/highlight style is displayed instead.

Examples:

```
SETCOLOUR   BD BR
```
Blue Default will be displayed as Blue REVVideo.

```
SC    RR YB    PD TU
```
Red REVVideo will be displayed as Yellow Blink and Pink Default will be displayed as Turquoise Underline.

SETFOCUS

Syntax:

```
>>--+- SETFOCUS -+---+--------------+-----------------------------------------><
    |              |   |              |
    +- SF -------+   +- windowname -+
```

Description:

Use the SETFOCUS command to change the focus window.

Parameters:

windowname
The name of the window to receive the focus. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

SHOWPOPUPMENU

Syntax:

```
>>--+- SHOWPOPUPMENU -+---+-------------+-------------------------------------><
    |                  |   |             |
    +- SPM ------------+   +-- winname --+
```

Description:

The SHOWPOPUPMENU command displays the options popup menu for the current storage display window.

Storage display windows include SELCOPY Debug Work Area and POS windows, CBLe Hex display windows and the CBLNAME window.

By default, the SHOWPOPUPMENU command is assigned to PF5 in storage display windows. The options popup menu may also be opened via the system menu button of the storage display window.

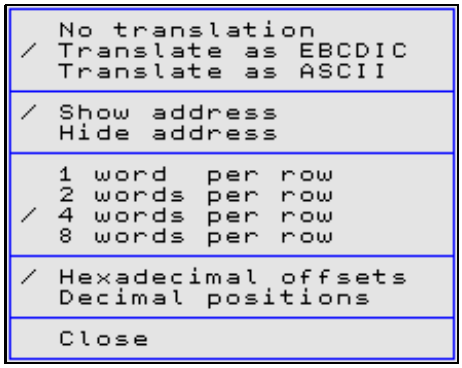


Figure 318. Storage Window Popup.

The mark "/" against items in the menu identifies the current status of the storage display window.

`No translation`
`Translate as EBCDIC`
`Translate as ASCII`
: Defines the interpretation of the hexadecimal data in the character field. (i.e. ASCII or EBCDIC.) If No translation is selected, then the character field is suppressed.

`Show address`
`Hide address`
: Defines whether the field containing the address in storage of each row of data is displayed or suppressed.

`1/2/4/8 words per row`
: Defines the number of words (length 4 bytes) are displayed in each row of data.

`Hexadecimal offsets`
`Decimal Positions`
: Defines whether the numeric field, displaying the displacement of each row of data relative to the first byte of data in the storage window, is presented as a hexadecimal offset or as a decimal position.
(e.g. row displayed as hexadecimal offset X'0000f0' is equivalent to decimal position 241.)

`Close`
: Closes the storage display window.

Parameters:

`winname`
: The name of the storage window for which the popup menu will apply. If not supplied then the name of the window in which the command is issued (via a command line or a function key) is assumed.

SHOWWATTR

Syntax:

```
>>--+- SHOWWATTR -+-------------------------------------------------------><
    |               |
    +- SWA --------+
```

Description:

Use the SHOWWATTR command to open the **Window Attributes** window to display the attributes of all open windows.

The Window Attributes window is essentially a List window and has the same charactersitics as List windows. For example select, sort and filter to display new views of the data are supported.

From a REXX edit macro you may use the following command to determine the focus window name.

```
'list SWA //  subset /select NAME where FOCUS="Y"/  stem MYSWA columns'
```

This will set the REXX variable "MYSWA.1.NAME" to the window name of the focus window, meaning that commands may be directed to that window using the WINDOWCOMMAND (WINCMD) command. e.g.

```
'wincmd' MYSWA.1.NAME 'where LRECL=80 and RECFM="F"'
```

For an example of its usage please see the distributed REXX macro "LISTSELD" (type "EM LISTSELD" to edit the macro) which provides a simplistic "dialog" interface for selecting and ordering the columns visible in such windows as Dataset, Library and VTOC lists.

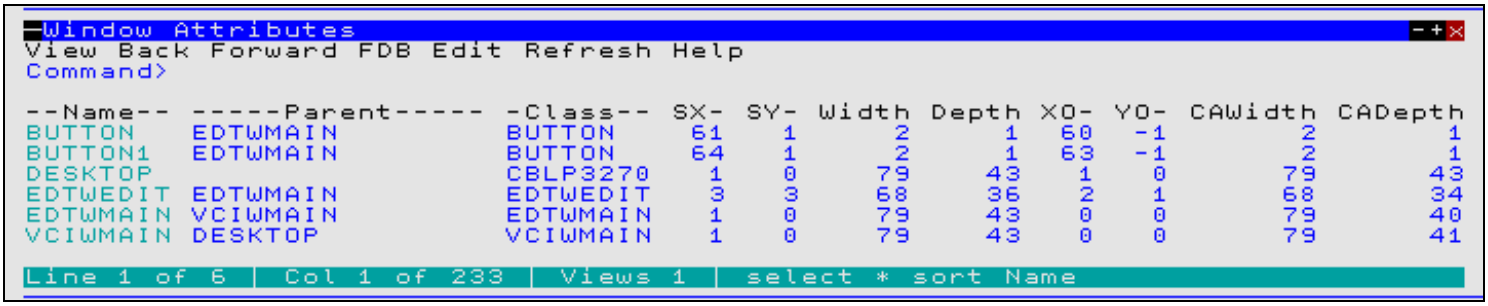


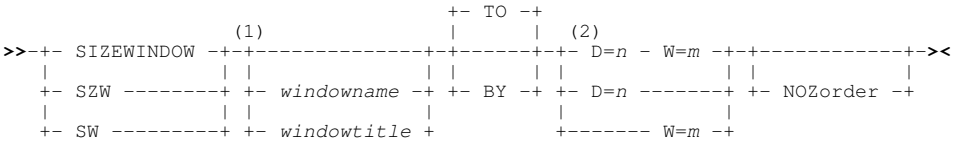
Figure 319. Window Attributes Window.

Columns Displayed:

Name	Type	Description
Name	Char	Window name
Parent	Char	Window parent name
Class	Char	Window class
SX	Int	Screen x coordinate
SY	Int	Screen y coordinate
Width	Int	Window width
Depth	Int	Window depth
XO	Int	X offset within parent
YO	Int	Y offset within parent
CAWidth	Int	Client area width
CADepth	Int	Client area depth
CAXO	Int	Client area x offset
CAYO	Int	Client area y offset
CursorX	Int	Cursor x coordinate
CursorY	Int	Cursor y coordinate
Title	Char	Window title
KeyList	Char	Keylist name
Focus	Char	Focus window

SIZEWINDOW

Syntax:



Notes:

1. SIZEWINDOW with no parameters displays this help.
2. D= and W= parameters may be entered in any order.

Description:

Use SIZEWINDOW to resize the specified window to an absolute depth and/or width or to a depth and/or width relative to the window's current size.

The window depth and window width are the number of rows and columns respectively that are displayed in the window's displayable area.

Note that the window's displayable area includes items such as the title bar and command line, but do not include the window's horizontal and vertical borders which occupy an additional 2 rows and 4 columns respectively. e.g. To determine the number of rows/columns of text that would be displayed on resizing a CBLe edit view, deduct 2 from the depth value (title bar, command line) and, if SCALE is set on, deduct an extra 1 for the scale line. If PREFIX is set on, deduct 6 columns from the width value for the prefix area.

The current position of the window is unchanged following a SIZEWINDOW command.

A window may also be resized by dragging the border of the window.

Parameters:

windowname
: The window name of the window to be resized. *windowname* is not case sensitive. If a matching window name is not found, the search for a matching window is repeated for *windowtitle*.

  If neither *windowname* nor *windowtitle* is supplied, the focus window is assumed.

windowtitle
: A blank delimited character string which is a substring of the window title of the window to be resized. *windowtitle* is not case sensitive and may occur at any offset within the window title.

  If *windowtitle* contains blanks then it must be enclosed in single- or double-quotes.

  If *windowtitle* matches the window name of an already open window, then it is treated as a being *windowname* and that window is selected.

  If *windowtitle* is found within the window title of more than one window, then the window selected is the one in which *windowtitle* occurs at the lowest offset within the title string. If *windowtitle* occurs at the same lowest offset in more than one window title, then the window that occurs next in the window chain is selected.

  windowtitle may **not** match any of the parameter keywords supported by SIZEWINDOW.

TO
: Indicate that an **absolute** depth and/or width follows. (Default)

BY
: Indicate that a **relative** depth and/or width follows.

D=*n*
: Define the window depth change.

  If absolute, *n* must be a positive integer. The number of rows displayed is equal to n and the bottom border of the window is raised or lowered accordingly.

  If relative, *n* is an integer that may be prefixed by "+" (plus) or "-" (minus) to indicate a positive or negative adjustment to the number of rows displayed. The number of rows displayed is increased or decreased by *n* rows and the bottom border of the window is raised or lowered accordingly.

  There is no default value for *n*. If omitted, the depth is unchanged.

W=*m*
: Define the window width change.

If absolute, *m* must be a positive integer. The number of columns displayed is equal to m and the right border of the window is adjusted accordingly.

If relative, *m* is an integer that may be prefixed by "+" (plus) or "-" (minus) to indicate a positive or negative adjustment to the number of columns displayed. The number of columns displayed is increased or decreased by *m* columns and the right border of the window is adjusted accordingly.

There is no default value for *m*. If omitted, the width is unchanged.

`NOZORDER`
Indicates that the window focus is to remain unchanged following the SIZEWINDOW operation.

Examples:

`SIZEWINDOW TO D=21`
Focus window is resized to display 21 rows. Number of columns is unaltered.

`SZW EDTWEDIT2 TO D=20 W=80`
Window with window name EDTWEDIT2 (a text edit view) is resized to be 20 rows by 80 columns.

`SZW CBL.JCL(SQ11740) TO D=+9 W=+5`
Window with window title CBL.JCL(SQ11740) is resized to be 9 rows deeper and 5 columns wider than its current size.

`SZW W=33 D=14`
Focus window is resized to display 14 rows and 33 columns.

`SZW BY D=8 W=-4`
Focus window is resized to display an additional 8 rows and 4 columns less.

See Also:

MOVEWINDOW and Text Editor primary command SET WINSIZE.

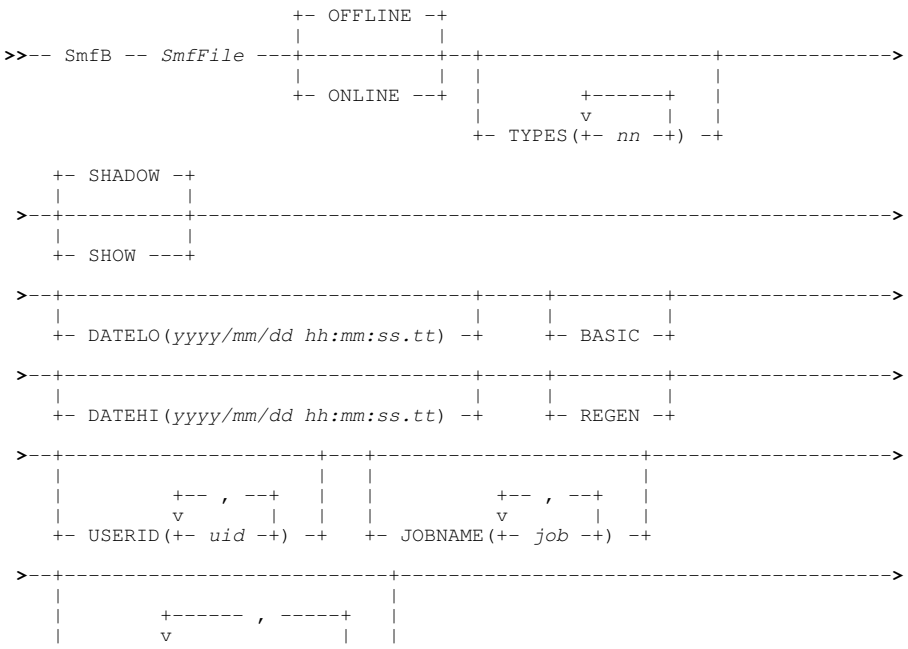
SMFB

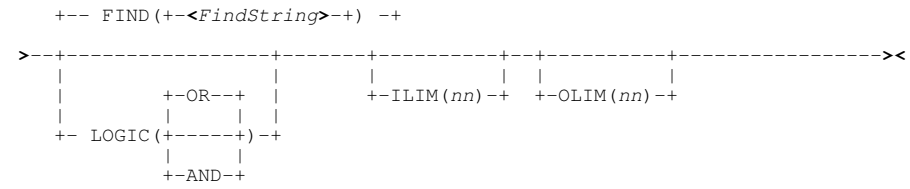
Examples:

`SmfB USER123.SMF.DATA Types( 14 15 30-5 ) ILim(10000)`
Selecting from the first 10000 records, display SMF types 14, 15 along with type 30 records that are subtype 5.

`SmfB USER123.SMF.DATA DateLo( 2018/09/15 13:00) DateHi( 2018/09/20 ) basic`
Display SMF records of any type using basic common header layout "SMFnnn" for all records, provided they fall withing the DATELO/DATEHI timestamp range.

Syntax:





Description:

SMFB provides a command-line interface (CLI) to the SMF Browse Utility.

Record Filtering

SMF records may be filtered so that only records that satisfy specified selection criteria are loaded and retained for subsequent browse.

A combination of record filtering parameters may be specified, each relating to one of the following record filtering classes:

1. Input record limit (ILIM)
2. Output record limit (OLIM)
3. Earliest timestamp (DATELO)
4. Latest timestamp (DATEHI)
5. Content match (TYPE, FIND, USERID and JOBNAME)

In order to be selected, a record must return a true condition for **all** the record filtering classes for which parameters have been specified. If no record filtering parameters are specified, then all records are selected for display.

The "Content match" class encompasses filter criteria specified for any combination of parameters **TYPE**, **FIND**, **USERID** and **JOBNAME**, each parameter itself returning either a true or false condition. The overall condition returned by the "Content match" class is based on the logical relationship between each of these parameters. The value specified by the **LOGIC** parameter determines this relationship as follows:

- For **LOGIC(OR)**, a true condition will be returned for the "Content match" class if at least **one** of the specified content match parameters returns a true condition. This is the default if LOGIC is not specified.
- For **LOGIC(AND)**, a true condition will be returned for the "Content match" class if **all** of the specified content match parameters return a true condition.

Each of the "Content match" parameters (TYPE, FIND, USERID and JOBNAME) support specification of multiple, **alternate** values. If any of these alternate values match the data in the record, then a true condition will be returned for that parameter.

e.g. A "Content match" class specification of **TYPES(14,15) JOBNAME(SMF*,DCH*) LOGIC(AND)** will return a true condition if the SMF record is identified as being either type 14 or 15 **AND** it contains a value beginning "SMF" or "DCH" in the zJOBNAME field.

Storage Management

If none of the record filtering parameters TYPES, DATELO, DATEHI, USERID, JOBNAME or FIND are specified, then only the first group of 100 records are loaded. Thereafter, additional groups of 100 records are loaded (and unloaded) as required (e.g. when scrolling vertically or on execution of a search or filter command). Beyond a few hundred records, SELCOPYi will attempt to minimise the amount of storage used by unloading as many groups of 100 records as possible.

If any of these record filtering parameters are specified, then the same technique applies except that the record group size starts at 1 and increments by 1 up to a maximum of 50 records. This is to minimize the amount of storage and I/O required in order to display each screen of selected records.

Parameters:

SmfFile
The name of a single existing, sequential or VSAM data set, GDG file generation, HFS file or PDS/PDSE library member containing SMF records that is to be browsed.

TYPES(*rr1,rr2 rr3-ss1 rr4#ss2 rr5:rr6*)
Specifies the SMF record types/subtypes to be selected.

A list of one or more record type (and optional subtype) numbers must be provided in brackets immediately following (no blank) the TYPES keyword, each separated by either blank or comma.

To request subtype *ss* of record type *rr* you may specify either *rr-ss* or *rr#ss*. e.g. 42-3 will specifically select SMF type 42 records of subtype 3.

Where a subtype is not included on a type specification, a colon ":" may be inserted between 2 record type values to identify a range of record types. e.g. 60:69 may be used to identify all SMF record types between 60 and 69 inclusively.

DATELO(*yyyy/mm/dd hh:mm:ss.tt*)
Specifies the minimum timestamp for record selection specified in **yyyy/mm/dd hh:mm:ss.t** format, which may be specified in as much detail as necessary. e.g. **2018/09/12** will be treated as **2018/09/12 00:00:00.0**

DATEHI(*yyyy/mm/dd hh:mm:ss.tt*)
Specifies the maximum timestamp for record selection in **yyyy/mm/dd hh:mm:ss.t** format, which may be specified in as much detail as necessary. e.g. **2018/09** will be treated as **2018/09/99 99:99:99.9**

ILIM(*nn*)
Specifies the maximum number of records that may be read from the SMF Dataset for potential selection.

See panel field Output Limit for further discussion.

OLIM(*nn*)
Specifies the maximum number of records that may be selected for browse.

See panel field Output Limit for further discussion.

FIND(*FindString1, FindString2, ...*)
FIND is one of the "Content match" class parameters and specifies one or more alternate, comma separated strings to be located at any position within an SMF record.

If a record contains any of the search strings then a true condition will be returned for the **FIND** parameter. e.g.

```
FIND( SYS1.MACLIB, SYS1.MIGLIB, SYS1.MODGEN, SYS1.MSGEN)
```

Find strings may be specified as any of the following:

- An **unquoted** string.

  Alpha character matching is case-insensitive unless wildcard characters are used. The string must not match a FIND command keyword and must not contain comma or blank characters. e.g. **ABC**.

- A **quoted** string (using either apostrophes or quotation marks).

  Alpha character matching is case-insensitive unless wildcard characters are used. e.g. **'A,B C'** and **"a,b C"** are equivalent.

- A **character literal** string specified as a quoted string prefixed with "C".

  Alpha character matching is case-sensitive and may include wildcard characters. e.g. **C'A,B C'** and **C'a,b C'**. are **not** equivalent.

- A **hexadecimal** string specified as a quoted string of hex digits prefixed with "X". e.g. **X'81C2C340'**.

- A **picture** string specified as a quoted string prefixed with "P".

  Special characters represent generic groups of characters as described below. Any character in a picture string that is not one of these special characters is untranslated. e.g. **P'A##-BC'**.

String	Description
P'='	Any character.
P'¬'	Any non-blank character.
P'.'	Any non-displayable character.
P'#'	Any numeric character, 0-9.
P'-'	Any non-numeric character.
P'@'	Any uppercase or lowercase alpha character.
P'<'	Any lowercase alpha character.
P'>'	Any uppercase alpha character.
P'$'	Any non-alphanumeric special character.

- A **regular expression** string specified as a quoted string prefixed with "R".

  Regular expressions enable powerful string pattern matching at the cost of rather complex syntax and potentially extended command processing time.

  For syntax and usage see Regular Expressions in the Text Editor documentation. e.g. **R'A:d+x'** would match the upper case character "A" followed by 1 or more numeric digits followed by character "x".

- An **unquoted**, **quoted** or **character literal** string containing one or more **wildcard** characters.

  Single-character wildcard '**%**' (percent) represents exactly one occurrence of any character. Multiple-character wildcard '*****' (asterisk) represents zero or more occurrences of any character.

  Beware of the following when using wildcard characters in a find string:

  1. Alpha character matching becomes case-sensitive even in **unquoted** and **quoted** strings.

2. For find strings containing a multiple-character wildcard, once a match has been found on the characters in the find string that precede it, the characters in the find string that follow it may be matched at any subsequent location within the SMF record. e.g. For find string 'ABC*DEF', 'ABC' may be matched in the first three characters of the record and 'DEF' in the last three characters of the record.

USERID(*uid1, uid2* etc)

USERID is one of the "Content match" class parameters and specifies one or more alternate, comma separated user id values to be matched in SMF record types known to contain a User Id field **(zUserId)** at a fixed location within the record data. This fixed position may be different for each of the SMF record types.

If a record contains any of the user id values then a true condition will be returned for the **USERID** parameter. If no match is found or the SMF record is not of a type known to contain a User Id field at a fixed location, then a false condition will be returned.

A user id value may be specified as an **unquoted**, **quoted** or **character literal** string and may contain one or more **wildcard** characters as described for the FIND parameter. e.g. A user id search string '*1' will match a user id value of any length (maximum 8) that ends with 1. A user id search string 'ABC%DEF' will match a user id beginning with 'ABC' followed by any other single character followed by 'DEF'.

Unlike a find string, a user id search value does not imply a trailing "*" (asterisk) wildcard and must start at the fixed position within the SMF record at which the zUserId field is located, for length 8 characters. If no wildcard characters are specified, a User id search is padded with blanks or truncated to 8 characters. If no wildcards are specified and the search sting is an **unquoted** or **quoted** string, then the alpha characters will be upper cased. e.g. A user id search string 'abc' will match user id 'ABC' but not 'ABC1', 'ABCXXX' or 'XABC'. Search string '%abc' would match 'XABC' only and '*abc*' would match all of these user ids.

The following SMF Record-types are known to contain a User Id field.

004	014	018	026	035	042	062	065	068	110
005	015	020	030	036	060	063	066	069	118
006	017	025	034	040	061	064	067	080	119
010									

JOBNAME(*job1, job2* etc)

JOBNAME is one of the "Content match" class parameters and specifies one or more alternate, comma separated job name values to be matched in SMF record types known to contain a Job Name field **(zJobName)** at a fixed location within the record data. This fixed position may be different for each of the SMF record types.

If a record contains any of the job name values then a true condition will be returned for the **JobName** parameter. If no match is found or the SMF record is not of a type known to contain a Job Name field at a fixed location, then a false condition will be returned.

A job name value may be specified as an **unquoted**, **quoted** or **character literal** string and may contain one or more **wildcard** characters as described for the FIND parameter.

Differences between the **JOBNAME** and **FIND** parameter specifications are as described for the USERID parameter.

The following SMF Record-types are known to contain a Job Name field.

004	010	017	025	034	040	061	064	067	080
005	014	018	026	035	042	062	065	068	110
006	015	020	030	036	060	063	066	069	118

LOGIC(AND) / LOGIC(OR)

The **LOGIC** parameter defines the logical operator to be applied between conditions returned by the 4 "Content match" class parameters (TYPES, FIND, USERID and JOBNAME).

OR indicates that the "Content match" class will return a true condition when a true condition is returned from **ANY** of the "Content match" class fields.

AND indicates that the "Content match" class will only return a true condition if true conditions are returned from **ALL** "Content match" class fields for which values have been specified.

For both **OR** and **AND** logical operations, a true condition will be returned for the "Content match" class if no selection condition values are specified in any of the "Content match" class fields.

ONLINE

ONLINE indicates that the SMF dataset is in the format as written directly by SMF. These are typically **SYS1.xxxx.MAN1/2/3/etc** datasets (SELCOPYi does not support browse of SMF records directly from the System Logger).

Browse of online datasets should be undertaken with caution as SELCOPYi will keep a SHR enqueue on the file for the duration of the browse session, which may interfere with SMF's archiving process.

ONLINE datasets include a **4-byte record descriptor word (RDW) prefix** at the start of each record, so record-type field mapping must be offset by this amount.

OFFLINE

OFFLINE indicates that the SMF dataset is the format as written by the SMF archiving tool (IFASMFDP) which does not include a 4-byte (RDW) record prefix. This is the default.

SHADOW

Applicable only when BASIC is not specified and affects only SMF records that are mapped using Base/Secondary segments.

SHADOW indicates that secondary segments should initially display as shadow lines. This is the default.

Note that while secondary segments are shadowed (thereby occupying much less screen space) the detail for any single shadowed segment may be displayed in a separate window by placing the cursor on the shadow line and pressing the **"ZoomW"** key **(Shift-F5).**

The shadow lines themselves may be suppressed using the SHADOW (SHAD) set option and/or the HIDE primary command.

Regardless of your initial setting for this option, the VBASE primary command may be used to switch back and forth throughout your browse session.

SHOW

Applicable only when BASIC is not specified and affects only SMF records that are mapped using Base/Secondary segments.

SHOW indicates that secondary segments should initially display in full detail

Regardless of your initial setting for this option, the VBASE primary command may be used to switch back and forth throughout your browse session.

BASIC

All records will be mapped using the basic layout **SMFnnn**.

Note that **"nnn"** here is a literal and does not represent the variable record-type number.

The basic layout includes the common header fields only (e.g. SMF record/sub-type, timestamp), followed by the field **"Rest"** which displays the tail end of the record as uninterpreted character data.

Use HEX ON to display hexadecimal representations.

If BASIC is not specified then SELCOPYi will use comprehensive mapping for supported record types.

Each supported SMF record/sub-type will be assigned a separate layout **SMFnnn_xxx_yyyy_zzz**, where **"nnn"** represents the variable record-type number, and **"xxx_yyyy_zzz"** is the short description
e.g. **SMF014_INPUT_or_RDBACK_Dataset**

SMF records that comprise potentially repeating groups will be mapped over several logical Primary/Secondary Segments.

Any selected records whose SMF record type is currently unsupported will be mapped using the "Basic Layout" record type **SMFnnn.**

REGEN

Specify REGEN to reload full layout definitions, and should only be used when you have updated your SMF definitions library since there is a significant performance implication to generating the layout Structure Definition Object (SDO) from the text library members.

Another implication of recreating the layout SDO is that any user updates will be lost. These may include

- ◊ any "permanent" alterations to the field display order/selection and/or column-widths made via the SDE SELECT Columns panel
- ◊ any row colouring options specified via the RCOLOUR command
- ◊ any column colouring options specified via the CCOLOUR command

See Also:

SMF Browse Utility panel
SMFEXTRC primary command.
REPORT primary command.

SMFEXTRC

Examples:

The following examples are as they might appear in a plain text file (e.g. the user's HOME command centre file) suitable for execution using the ACTION key.

```
<SmfExtrc  CBL.SMF.GDG    USER123.SMF030 Types(30-1 30-5)
```

Extract SMF type 30 subtypes 1 and 5 records from all generations of **"CBL.SMF.GDG"**, placing them in output dataset **"USER123.SMF030"**.

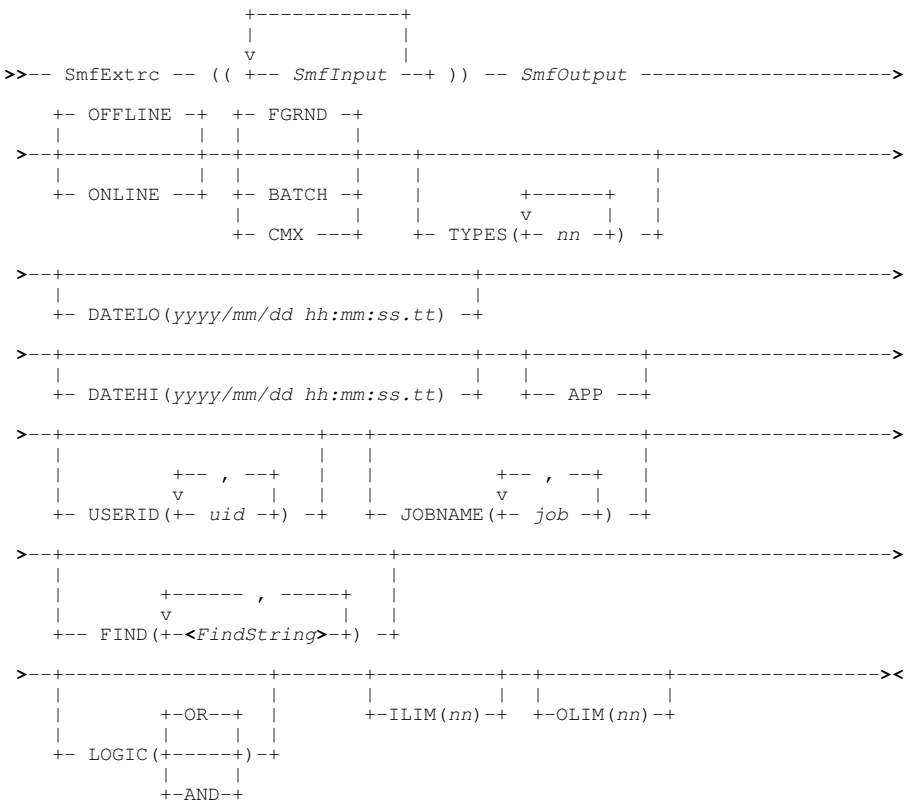
```
<SmfExtrc (( CBL.SMF.GDG.G1225V00 CBL.SMF.GDG.G1226V00 )) USER123.SMFEXT     \
            DateLo( 2018/09/15 13:00) DateHi( 2018/09/20 )
```

Extract SMF records of any type from **"CBL.SMF.GDG.G1225V00"** and **"CBL.SMF.GDG.G1226V00"** provided they fall withing the DATELO/DATEHI timestamp range, placing them in output dataset **"USER123.SMFEXT"**.

```
<SmfExtrc                      \
  ((                           \
    CBL.SMF.GDG(-3)            \
    CBL.SMF.GDG(-2)            \
    CBL.SMF.GDG(-1)            \
    CBL.SMF.GDG(0)             \
  ))      \
    USER123.SMFJOB.EXTRACT     \
    JobName('SEL4 ', 'SEL9 ' ) \
    OLim( 10000 )
```

Selecting from the -3 up to the 0 generation of **"CBL.SMF.GDG"**, extract records of the various record types that reference either job name **"SEL4"** or **"SEL9"** only, placing them in output dataset **"USER123.SMFJOB.EXTRACT"**. Limit the output to a maximum of 10,000 records.

Syntax:



Description:

SMFEXTRC provides a command-line interface (CLI) to the SMF Extract Utility.

Record Filtering

SMF records may be filtered so that only records that satisfy specified selection criteria are selected for extraction.

A combination of record filtering parameters may be specified, each relating to one of the following record filtering classes:

1. Input record limit (ILIM)
2. Output record limit (OLIM)
3. Earliest timestamp (DATELO)
4. Latest timestamp (DATEHI)
5. Content match (TYPE, FIND, USERID and JOBNAME)

In order to be selected, a record must return a true condition for **all** the record filtering classes for which parameters have been specified. If no record filtering parameters are specified, then all records are selected for display.

The "Content match" class encompasses filter criteria specified for any combination of parameters **TYPE**, **FIND**, **USERID** and **JOBNAME**, each parameter itself returning either a true or false condition. The overall condition returned by the "Content match" class is based on the logical relationship between each of these parameters. The value specified by the **LOGIC** parameter determines this relationship as follows:

- For **LOGIC(OR)**, a true condition will be returned for the "Content match" class if at least **one** of the specified content match parameters returns a true condition. This is the default if LOGIC is not specified.
- For **LOGIC(AND)**, a true condition will be returned for the "Content match" class if **all** of the specified content match parameters return a true condition.

Each of the "Content match" parameters (TYPE, FIND, USERID and JOBNAME) support specification of multiple, **alternate** values. If any of these alternate values match the data in the record, then a true condition will be returned for that parameter.

e.g. A "Content match" class specification of **TYPES(14,15) JOBNAME(SMF*,DCH*) LOGIC(AND)** will return a true condition if the SMF record is identified as being either type 14 or 15 **AND** it contains a value beginning "SMF" or "DCH" in the zJOBNAME field.

Parameters:

SmfInput
: The name of an existing, sequential or VSAM data set, GDG base, GDG file generation, HFS file or PDS/PDSE library member from which SMF records are to be selected for extraction.

  If a GDG base name is provided then all generations (oldest to newest) will be processed as input.

  If more than one input dataset is required a blank separated list maybe supplied between a leading **"((** (double open-parentheses) and a trailing **"))"** (double close-parentheses).

SmfOutput
: The name of a sequential or VSAM data set, GDG file generation, HFS file or PDS/PDSE library member to which selected SMF records are to be written.

TYPES(*rr1,rr2 rr3-ss1 rr4#ss2 rr5:rr6*)
: Specifies the SMF record types/subtypes to be selected.

  A list of one or more record type (and optional subtype) numbers must be provided in brackets immediately following (no blank) the TYPES keyword, each separated by either blank or comma.

  To request subtype *ss* of record type *rr* you may specify either *rr-ss* or *rr#ss*. e.g. 42-3 will specifically select SMF type 42 records of subtype 3.

  Where a subtype is not included on a type specification, a colon ":" may be inserted between 2 record type values to identify a range of record types. e.g. 60:69 may be used to identify all SMF record types between 60 and 69 inclusively.

DATELO(*yyyy/mm/dd hh:mm:ss.tt*)
: Specifies the minimum timestamp for record selection specified in **yyyy/mm/dd hh:mm:ss.t** format, which may be specified in as much detail as necessary. e.g. **2018/09/12** will be treated as **2018/09/12 00:00:00.0**

DATEHI(*yyyy/mm/dd hh:mm:ss.tt*)
: Specifies the maximum timestamp for record selection in **yyyy/mm/dd hh:mm:ss.t** format, which may be specified in as much detail as necessary. e.g. **2018/09** will be treated as **2018/09/99 99:99:99.9**

ILIM(*nn*)
: Specifies the maximum number of records that may be read from each SMF input dataset for potential selection.

  See panel field Output Limit for further discussion.

OLIM(*nn*)
: Specifies the maximum number of records that may be selected for output to the SMF output dataset

  See panel field Output Limit for further discussion.

FIND(*FindString1, FindString2, ...*)
: **FIND** is one of the "Content match" class parameters and specifies one or more alternate, comma separated strings to be located at any position within an SMF record.

  If a record contains any of the search strings then a true condition will be returned for the **FIND** parameter. e.g.

```
FIND( SYS1.MACLIB, SYS1.MIGLIB, SYS1.MODGEN, SYS1.MSGEN)
```

Find strings may be specified as any of the following:

- An **unquoted** string.

  Alpha character matching is case-insensitive unless wildcard characters are used. The string must not match a FIND command keyword and must not contain comma or blank characters. e.g. **ABC**.

- A **quoted** string (using either apostrophes or quotation marks).

  Alpha character matching is case-insensitive unless wildcard characters are used. e.g. **'A,B C'** and **"a,b C"** are equivalent.

- A **character literal** string specified as a quoted string prefixed with "C".

  Alpha character matching is case-sensitive and may include wildcard characters. e.g. **C'A,B C'** and **C'a,b C'**. are **not** equivalent.

- A **hexadecimal** string specified as a quoted string of hex digits prefixed with "X". e.g. **X'81C2C340'**.

- A **picture** string specified as a quoted string prefixed with "P".

  Special characters represent generic groups of characters as described below. Any character in a picture string that is not one of these special characters is untranslated. e.g. **P'A##-BC'**.

String	Description
P'='	Any character.
P'¬'	Any non-blank character.
P','	Any non-displayable character.
P'#'	Any numeric character, 0-9.
P'-'	Any non-numeric character.
P'@'	Any uppercase or lowercase alpha character.
P'<'	Any lowercase alpha character.
P'>'	Any uppercase alpha character.
P'$'	Any non-alphanumeric special character.

- A **regular expression** string specified as a quoted string prefixed with "R".

  Regular expressions enable powerful string pattern matching at the cost of rather complex syntax and potentially extended command processing time.

  For syntax and usage see Regular Expressions in the Text Editor documentation. e.g. **R'A:d+x'** would match the upper case character "A" followed by 1 or more numeric digits followed by character "x".

- An **unquoted**, **quoted** or **character literal** string containing one or more **wildcard** characters.

  Single-character wildcard '**%**' (percent) represents exactly one occurrence of any character. Multiple-character wildcard '*' (asterisk) represents zero or more occurrences of any character.

  Beware of the following when using wildcard characters in a find string:

  1. Alpha character matching becomes case-sensitive even in **unquoted** and **quoted** strings.
  2. For find strings containing a multiple-character wildcard, once a match has been found on the characters in the find string that precede it, the characters in the find string that follow it may be matched at any subsequent location within the SMF record. e.g. For find string 'ABC*DEF', 'ABC' may be matched in the first three characters of the record and 'DEF' in the last three characters of the record.

USERID(***uid1, uid2*** **etc)**

USERID is one of the "Content match" class parameters and specifies one or more alternate, comma separated user id values to be matched in SMF record types known to contain a User Id field **(zUserId)** at a fixed location within the record data. This fixed position may be different for each of the SMF record types.

If a record contains any of the user id values then a true condition will be returned for the **USERID** parameter. If no match is found or the SMF record is not of a type known to contain a User Id field at a fixed location, then a false condition will be returned.

A user id value may be specified as an **unquoted**, **quoted** or **character literal** string and may contain one or more **wildcard** characters as described for the FIND parameter. e.g. A user id search string '*1' will match a user id value of any length (maximum 8) that ends with 1. A user id search string 'ABC%DEF' will match a user id beginning with 'ABC' followed by any other single character followed by 'DEF'.

Unlike a find string, a user id search value does not imply a trailing "*" (asterisk) wildcard and must start at the fixed position within the SMF record at which the zUserId field is located, for length 8 characters. If no wildcard characters are specified, a User id search is padded with blanks or truncated to 8 characters. If no wildcards are specified and the search sting is an **unquoted** or **quoted** string, then the alpha characters will be upper cased. e.g. A user id search string 'abc' will match user id 'ABC' but not 'ABC1', 'ABCXXX' or 'XABC'. Search string '%abc' would match 'XABC' only and '*abc*' would match all of these user ids.

The following SMF Record-types are known to contain a User Id field.

004	014	018	026	035	042	062	065	068	110
005	015	020	030	036	060	063	066	069	118
006	017	025	034	040	061	064	067	080	119
010									

JOBNAME(*job1, job2* etc)

JOBNAME is one of the "Content match" class parameters and specifies one or more alternate, comma separated job name values to be matched in SMF record types known to contain a Job Name field **(zJobName)** at a fixed location within the record data. This fixed position may be different for each of the SMF record types.

If a record contains any of the job name values then a true condition will be returned for the **JobName** parameter. If no match is found or the SMF record is not of a type known to contain a Job Name field at a fixed location, then a false condition will be returned.

A job name value may be specified as an **unquoted**, **quoted** or **character literal** string and may contain one or more **wildcard** characters as described for the FIND parameter.

Differences between the **JOBNAME** and **FIND** parameter specifications are as described for the USERID parameter.

The following SMF Record-types are known to contain a Job Name field.

004	010	017	025	034	040	061	064	067	080
005	014	018	026	035	042	062	065	068	110
006	015	020	030	036	060	063	066	069	118

LOGIC(AND) / LOGIC(OR)

The **LOGIC** parameter defines the logical operator to be applied between conditions returned by the 4 "Content match" class parameters (TYPES, FIND, USERID and JOBNAME).

OR indicates that the "Content match" class will return a true condition when a true condition is returned from **ANY** of the "Content match" class fields.

AND indicates that the "Content match" class will only return a true condition if true conditions are returned from **ALL** "Content match" class fields for which values have been specified.

For both **OR** and **AND** logical operations, a true condition will be returned for the "Content match" class if no selection condition values are specified in any of the "Content match" class fields.

ONLINE

Indicates that the input SMF dataset(s) are in the format as written directly by SMF. These are typically **SYS1.xxxx.MAN1/2/3/etc** datasets (SELCOPYi does not support browse of SMF records directly from the System Logger).

ONLINE datasets include a **4-byte record descriptor word (RDW) prefix** at the start of each record, so record-type field mapping must be offset by this amount.

The output dataset will be created in the same format as the input.

OFFLINE

Indicates that the input SMF dataset(s) are the format as written by the SMF archiving tool (IFASMFDP) which does not include a 4-byte (RDW) record prefix. This is the default.

The output dataset will be created in the same format as the input.

APP

Indicates that any selected input records should be appended to the output dataset.

FGRND

Indicates that immediate foreground execution is required. This is the default.

FGRND

Indicates that JCL should be produced for submission to batch.

CMX

Indicates that command line interface should be produced. The SMFEXTRC primary command is displayed in a Text-Edit window in a format suitable for execution using the ACTION key (Shift-F4) ready to be copied into your HOME file (=4).

See Also:

SMF Extract Utility panel
SMFB primary command.
REPORT primary command.

SMFRPT

SMFRPT command has been deprecated. Use **REPORT** instead.

Examples:

The following examples are as they might appear in a plain text file (e.g. the user's HOME command centre file) suitable for execution using the ACTION key.

```
<SMFRPT RUN T030SUM               \
          SMFRPT-INPUT-BEG        \
             USER123.SMF030       \
             TYPES(30-5)          \
             DATELO( 2018/09/15 13:00) \
             DATEHI( 2018/09/20 ) \
          SMFRPT-INPUT-END
```

Using the report definition saved in dataset "*userid* **.SELCOPYI.SMF.RPT(T030SUM)**", produce a report from all **SMF Record-Type 30 SubType 5** records contained in dataset **USER123.SMF030** provided they fall within the DATELO/DATEHI timestamp range.

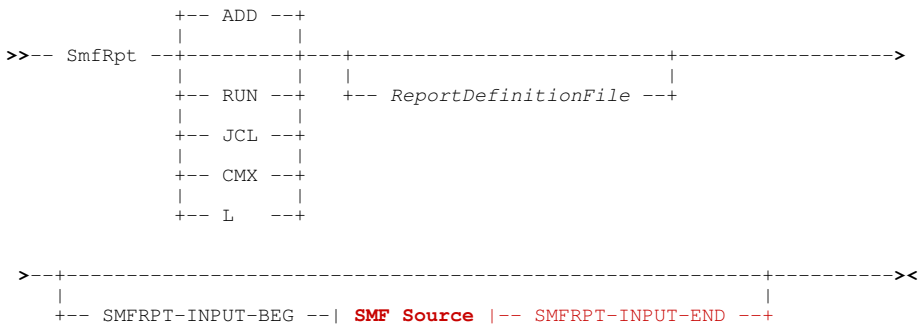
The **TO30SUM** member may contain the following report definition statements:

```
TITLE:
 Job/DD EXCPs Report (from SMF Type 30 Subtype 5)

COLUMNS:
 SMF030_Identification.zJOBNAME            'Job Name'
 SMF030_Identification.zSIT                'Job Start'
 SMF030_Common_Address_Space_Work.ZTME     'Job End'
 SMF030_EXCP.zDDN                          'DDName'
 SMF030_EXCP.zBLK                          'EXCP Blks'

REPEAT:
 SMF030_EXCP
```

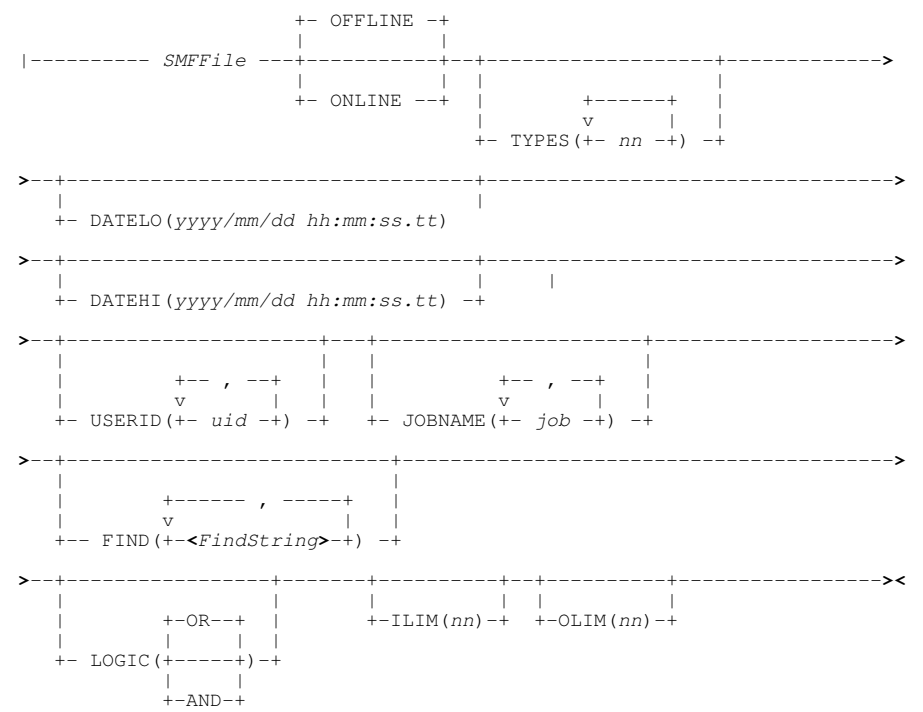
Executing of the SMFRPT command would create a report output that looks something like the following:

```
12018/09/14 12:04   Job/DD EXCPs Report (from SMF Type 30 Subtype 5)  PAGE    1

 Job Name Job Start               Job End                 DDName   EXCP Blks
 -------- ----------------------- ----------------------- -------- -----------
 SMFCLEAR 2018/09/04 01:09:18.03 2018/09/04 01:09:29.73 INDD1          14400
                                                        DUMPOUT           786
                                                        SYSPRINT            0
                                                        SYSIN               2
 SMFCLEAR 2018/09/04 02:55:15.33 2018/09/04 02:55:26.97 INDD1          14400
                                                        DUMPOUT           783
                                                        SYSPRINT            0
                                                        SYSIN               2
 SMFCLEAR 2018/09/04 04:41:32.55 2018/09/04 04:41:45.63 INDD1          14400
                                                        DUMPOUT           786
                                                        SYSPRINT            0
                                                        SYSIN               2
 SMFCLEAR 2018/09/04 06:27:59.30 2018/09/04 06:28:09.90 INDD1          14400
                                                        DUMPOUT           788
                                                        SYSPRINT            0
                                                        SYSIN               2
--------------------------- 6178 line(s) not displayed --------------------------
12018/09/14 12:04   Job/DD EXCPs Report (from SMF Type 30 Subtype 5)  PAGE  113

 Job Name Job Start               Job End                 DDName   EXCP Blks
 -------- ----------------------- ----------------------- -------- -----------
 SMFCLEAR 2018/09/07 09:54:14.26 2018/09/07 09:54:25.94 INDD1          14400
                                                        DUMPOUT           796
                                                        SYSPRINT            0
                                                        SYSIN               2
 SMFCLEAR 2018/09/07 11:31:38.69 2018/09/07 11:31:51.49 INDD1          14400
                                                        DUMPOUT           792
                                                        SYSPRINT            0
                                                        SYSIN               2
 SMFCLEAR 2018/09/07 13:08:53.59 2018/09/07 13:09:05.39 INDD1          14400
                                                        DUMPOUT           786
                                                        SYSPRINT            0
                                                        SYSIN               2


                                                                  ===========
                                                                      1139389
                                                                  ===========
```

Syntax:



SMF Source:



Description:

SMFRPT provides a command-line interface (CLI) to the SMF Report Utility.

The layout of your desired report should be specified using control statements saved in the Report Definition dataset.

The report produced will consist mainly of data extracted from a list of data fields from a single SMF Record-Type. The report can reference fields from both Primary (Base) and Secondary segments.

A user definable heading will be printed at the top of each page, followed by user definable columns heading for each selected field.

Grand totals will automatically be printed for any selected field containing integer data, and sub-totals will also be printed if a sort/control-break has been requested.

See SMF Report Definition Control Statements for full details of supported report definition control statements.

The utility supports sourcing SMF data from either the **"SYS1.xxxx.MANn"** dataset format (VSAM ESDS with 4-byte "RDW" prefix) that is written to directly by SMF, or the dataset format written by the **SMF UNLOAD utility**(normally **RECFM=VBS** with no 4-byte "RDW" prefix).

SMF Features Further Information provides information about SELCOPYi's handling of SMF data.

Record Filtering

SMF records may be filtered so that only records that satisfy specified selection criteria are selected for reporting.

A combination of record filtering parameters may be specified, each relating to one of the following record filtering classes:

1. Input record limit (ILIM)

2. Output record limit (OLIM)
3. Earliest timestamp (DATELO)
4. Latest timestamp (DATEHI)
5. Content match (TYPE, FIND, USERID and JOBNAME)

In order to be selected, a record must return a true condition for **all** the record filtering classes for which parameters have been specified. If no record filtering parameters are specified, then all records are selected for display.

The "Content match" class encompasses filter criteria specified for any combination of parameters **TYPE**, **FIND**, **USERID** and **JOBNAME**, each parameter itself returning either a true or false condition. The overall condition returned by the "Content match" class is based on the logical relationship between each of these parameters. The value specified by the **LOGIC** parameter determines this relationship as follows:

- For **LOGIC(OR)**, a true condition will be returned for the "Content match" class if at least **one** of the specified content match parameters returns a true condition. This is the default if LOGIC is not specified.
- For **LOGIC(AND)**, a true condition will be returned for the "Content match" class if **all** of the specified content match parameters return a true condition.

Each of the "Content match" parameters (TYPE, FIND, USERID and JOBNAME) support specification of multiple, **alternate** values. If any of these alternate values match the data in the record, then a true condition will be returned for that parameter.

e.g. A "Content match" class specification of **TYPES(14,15) JOBNAME(SMF*,DCH*) LOGIC(AND)** will return a true condition if the SMF record is identified as being either type 14 or 15 **AND** it contains a value beginning "SMF" or "DCH" in the zJOBNAME field.

Parameters:

ReportDefinitionFile
The name of an data set or PDS/PDSE library member that contains the SMF report definition control statements.

Click here for details of supported report definition control statements.

If a member name only is supplied then it is assumed to be in library *userid*.**SELCOPYI.SMF.RPT**.

TYPES(*rr1,rr2 rr3-ss1 rr4#ss2 rr5:rr6*)
Specifies the SMF record types/subtypes to be selected.

A list of one or more record type (and optional subtype) numbers must be provided in brackets immediately following (no blank) the TYPES keyword, each separated by either blank or comma.

To request subtype *ss* of record type *rr* you may specify either *rr-ss* or *rr#ss*. e.g. 42-3 will specifically select SMF type 42 records of subtype 3.

Where a subtype is not included on a type specification, a colon ":" may be inserted between 2 record type values to identify a range of record types. e.g. 60:69 may be used to identify all SMF record types between 60 and 69 inclusively.

DATELO(*yyyy/mm/dd hh:mm:ss.tt*)
Specifies the minimum timestamp for record selection specified in **yyyy/mm/dd hh:mm:ss.t** format, which may be specified in as much detail as necessary. e.g. **2018/09/12** will be treated as **2018/09/12 00:00:00.0**

DATEHI(*yyyy/mm/dd hh:mm:ss.tt*)
Specifies the maximum timestamp for record selection in **yyyy/mm/dd hh:mm:ss.t** format, which may be specified in as much detail as necessary. e.g. **2018/09** will be treated as **2018/09/99 99:99:99.9**

ILIM(*nn*)
Specifies the maximum number of records that may be read from each SMF input dataset for potential selection.

See panel field Output Limit for further discussion.

OLIM(*nn*)
Specifies the maximum number of records that may be selected for output to the SMF output dataset

See panel field Output Limit for further discussion.

FIND(*FindString1, FindString2, ...*)
FIND is one of the "Content match" class parameters and specifies one or more alternate, comma separated strings to be located at any position within an SMF record.

If a record contains any of the search strings then a true condition will be returned for the **FIND** parameter. e.g.

```
FIND( SYS1.MACLIB, SYS1.MIGLIB, SYS1.MODGEN, SYS1.MSGEN)
```

Find strings may be specified as any of the following:

- An **unquoted** string.

  Alpha character matching is case-insensitive unless wildcard characters are used. The string must not match a FIND command keyword and must not contain comma or blank characters. e.g. **ABC**.

- A **quoted** string (using either apostrophes or quotation marks).

  Alpha character matching is case-insensitive unless wildcard characters are used. e.g. `'A,B C'` and `"a,b C"` are equivalent.

- A **character literal** string specified as a quoted string prefixed with "C".

  Alpha character matching is case-sensitive and may include wildcard characters. e.g. `C'A,B C'` and `C'a,b C'`. are **not** equivalent.

- A **hexadecimal** string specified as a quoted string of hex digits prefixed with "X". e.g. `X'81C2C340'`.

- A **picture** string specified as a quoted string prefixed with "P".

  Special characters represent generic groups of characters as described below. Any character in a picture string that is not one of these special characters is untranslated. e.g. `P'A##-BC'`.

String	Description
P'='	Any character.
P'¬'	Any non-blank character.
P'.'	Any non-displayable character.
P'#'	Any numeric character, 0-9.
P'-'	Any non-numeric character.
P'@'	Any uppercase or lowercase alpha character.
P'<'	Any lowercase alpha character.
P'>'	Any uppercase alpha character.
P'$'	Any non-alphanumeric special character.

- A **regular expression** string specified as a quoted string prefixed with "R".

  Regular expressions enable powerful string pattern matching at the cost of rather complex syntax and potentially extended command processing time.

  For syntax and usage see Regular Expressions in the Text Editor documentation. e.g. `R'A:d+x'` would match the upper case character "A" followed by 1 or more numeric digits followed by character "x".

- An **unquoted**, **quoted** or **character literal** string containing one or more **wildcard** characters.

  Single-character wildcard '**%**' (percent) represents exactly one occurrence of any character. Multiple-character wildcard '*****' (asterisk) represents zero or more occurrences of any character.

  Beware of the following when using wildcard characters in a find string:

  1. Alpha character matching becomes case-sensitive even in **unquoted** and **quoted** strings.
  2. For find strings containing a multiple-character wildcard, once a match has been found on the characters in the find string that precede it, the characters in the find string that follow it may be matched at any subsequent location within the SMF record. e.g. For find string 'ABC*DEF', 'ABC' may be matched in the first three characters of the record and 'DEF' in the last three characters of the record.

USERID(*uid1, uid2* etc)

USERID is one of the "Content match" class parameters and specifies one or more alternate, comma separated user id values to be matched in SMF record types known to contain a User Id field **(zUserId)** at a fixed location within the record data. This fixed position may be different for each of the SMF record types.

If a record contains any of the user id values then a true condition will be returned for the **USERID** parameter. If no match is found or the SMF record is not of a type known to contain a User Id field at a fixed location, then a false condition will be returned.

A user id value may be specified as an **unquoted**, **quoted** or **character literal** string and may contain one or more **wildcard** characters as described for the FIND parameter. e.g. A user id search string '*1' will match a user id value of any length (maximum 8) that ends with 1. A user id search string 'ABC%DEF' will match a user id beginning with 'ABC' followed by any other single character followed by 'DEF'.

Unlike a find string, a user id search value does not imply a trailing "*" (asterisk) wildcard and must start at the fixed position within the SMF record at which the zUserId field is located, for length 8 characters. If no wildcard characters are specified, a User id search is padded with blanks or truncated to 8 characters. If no wildcards are specified and the search sting is an **unquoted** or **quoted** string, then the alpha characters will be upper cased. e.g. A user id search string 'abc' will match user id 'ABC' but not 'ABC1', 'ABCXXX' or 'XABC'. Search string '%abc' would match 'XABC' only and '*abc*' would match all of these user ids.

The following SMF Record-types are known to contain a User Id field.

004 005 006 010	014 015 017	018 020 025	026 030 034	035 036 040	042 060 061	062 063 064	065 066 067	068 069 080	110 118 119

JOBNAME(*job1, job2* etc)

JOBNAME is one of the "Content match" class parameters and specifies one or more alternate, comma separated job name values to be matched in SMF record types known to contain a Job Name field **(zJobName)** at a fixed location within the record data. This fixed position may be different for each of the SMF record types.

If a record contains any of the job name values then a true condition will be returned for the **JobName** parameter. If no match is found or the SMF record is not of a type known to contain a Job Name field at a fixed location, then a false condition will be returned.

A job name value may be specified as an **unquoted**, **quoted** or **character literal** string and may contain one or more **wildcard** characters as described for the FIND parameter.

Differences between the **JOBNAME** and **FIND** parameter specifications are as described for the USERID parameter.

The following SMF Record-types are known to contain a Job Name field.

004	010	017	025	034	040	061	064	067	080
005	014	018	026	035	042	062	065	068	110
006	015	020	030	036	060	063	066	069	118

LOGIC(AND) / LOGIC(OR)

The **LOGIC** parameter defines the logical operator to be applied between conditions returned by the 4 "Content match" class parameters (TYPES, FIND, USERID and JOBNAME).

OR indicates that the "Content match" class will return a true condition when a true condition is returned from **ANY** of the "Content match" class fields.

AND indicates that the "Content match" class will only return a true condition if true conditions are returned from **ALL** "Content match" class fields for which values have been specified.

For both **OR** and **AND** logical operations, a true condition will be returned for the "Content match" class if no selection condition values are specified in any of the "Content match" class fields.

ONLINE

Indicates that the input SMF dataset(s) are in the format as written directly by SMF. These are typically **SYS1.xxxx.MAN1/2/3/etc** datasets (SELCOPYi does not support browse of SMF records directly from the System Logger).

ONLINE datasets include a **4-byte record descriptor word (RDW) prefix** at the start of each record, so record-type field mapping must be offset by this amount.

The output dataset will be created in the same format as the input.

OFFLINE

Indicates that the input SMF dataset(s) are the format as written by the SMF archiving tool (IFASMFDP) which does not include a 4-byte (RDW) record prefix. This is the default.

The output dataset will be created in the same format as the input.

FGRND | RUN

Indicates that immediate execution is required. This is the default.

If running interactively under SELCOPYi (as apposed to running in batch using PGM=SDEAMAIN) then the report output will be collected in-storage and displayed in a Text-Edit window. The output will not be saved to disk but the user may enter the CREATE or REPLACE commands to do so.

If the expected report output is larger than your available foreground region then you should choose the JCL option to generate a batch job which will be write the report to the DD name **SDEPRINT**.

BATCH | JCL

Indicates that JCL should be produced for submission to batch.

CMX

Indicates that command line interface should be produced. The SMFRPT primary command is displayed in a Text-Edit window in a format suitable for execution using the ACTION key (Shift-F4) ready to be copied into your HOME file (=4).

L

Lists all members of report deinition library *userid*.**SELCOPYI.SMF.RPT**.

ADD

This option is intended for execution via a function key (F1-F24) and assists the user creating a report definition by generating "COLUMNS:" section control statements for all "SELECTed" columns within the focus record during an SMF Browse Session.

For example:

1. Start a formatted browse of an SMF dataset containing the type of record you wish to report on, then navigate to an instance of that record-type.
   e.g. Type **"NEXT SMF014"**

2. Use the KEYS dialog or set a temporary function key definition
   e.g. Type **"PF 6 SMFRPT ADD T014RPT1"**

3. Type primary command SEL to open a dialog that will allow you to pick the columns you wish to see in your report (and the order they appear).

   Alternatively, if you know the names (or ref numbers) of the fields you require just type **"SEL *fieldname1,fieldname2*,etc"**

4. On exit from the dialog (your browse view will have updated accordingly), **press function key F6**

5. Library member *userid*.**SELCOPYI.SMF.RPT**(*T014RPT1*) will be displayed with a new "COLUMNS:" style control statement added for each selected field.

   For each entry, a **default heading** is generated using up to 30 characters from the field's description. A **width** value is also included (set to the actual field width) ready for user modification.

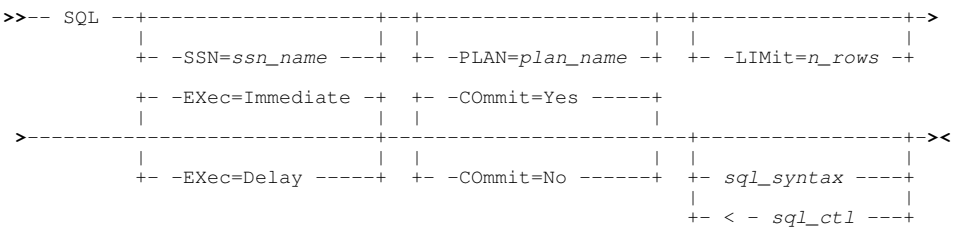
6. Repeat this process for any secondary segments if required.

See Also:

SMF Report Utility panel
SMFB primary command.
SMFEXTRC primary command.

SQL

Syntax:



Description:

Use the SQL command to open the Dynamic SQL window.

The Dynamic SQL window may also be opened via the File menu of the SELCOPY/i main window menu bar.

The Dynamic SQL window connects the user to the DB2 subsystem using the DB2 subsystem name and plan specified in the **CBLNAME** load module.

Note: Not implemented for CMS or VSE.

Parameters:

`-SSN=`*`ssn_name`*
: The DB2 sub-system to be the target of the CONNECT.
This parameter is placed in the "DB2 Subsystem>" field of the Dynamic SQL window.
Default is that defined by the SELCOPY/i INI option, DB2.SSN, otherwise the sub-system name specified in the DB2SubSys field of the CBLNAME load module is used.

`-PLAN=`*`plan_name`*
: The SELCOPY DB2 plan name which has been bound to the DB2 sub-system.
This parameter is placed in the "Plan>" field of the Dynamic SQL window.
Default is that defined by the SELCOPY/i INI option, DB2.Plan, otherwise the plan name specified in the DB2Plan field of the CBLNAME load module is used.

`-LIMIT=`*`n_rows`*
: Limit the number of rows to be displayed in the Dynamic SQL window following a SELECT transaction. Once the limit threshold has been reached, a pop-up message window is displayed and no further attempt is made to retrieve selected rows of data.
The *n_rows* value is placed in the "Select Limit>" field of the Dynamic SQL window.
The default limit is that defined by the SELCOPY/i INI option, DB2.SelectLimit, otherwise no limit is implied.

`-EXEC=IMMEDIATE|DELAY`

Determine whether the SQL command is to be executed immediately when the Dynamic SQL window is opened or simply placed on the SQL Statement command line.
The default is IMMEDIATE.

`-COMMIT=YES|NO`
Determine whether a COMMIT is to be automatically issued following every transaction (AutoCommit). If COMMIT=NO, then the user should issue COMMIT manually to commit any changes made to the data. A commit is executed automatically when the Dynamic SQL window is closed, regardless of the AutoCommit field setting.
The commit value is reflected in the "AutoCommit>" field of the Dynamic SQL window.
The default is YES.

`sql_syntax`
Valid SQL syntax to be executed when the Dynamic SQL widow is opened.
The *sql_syntax* string is placed in the first SQL Statement line field of the Dynamic SQL window.

`< sql_ctl`
Input contol file containing one or more valid SQL statements to be executed when the Dynamic SQL widow is opened.
The "< *sql_ctl*" string is placed in the first SQL Statement line field of the Dynamic SQL window.

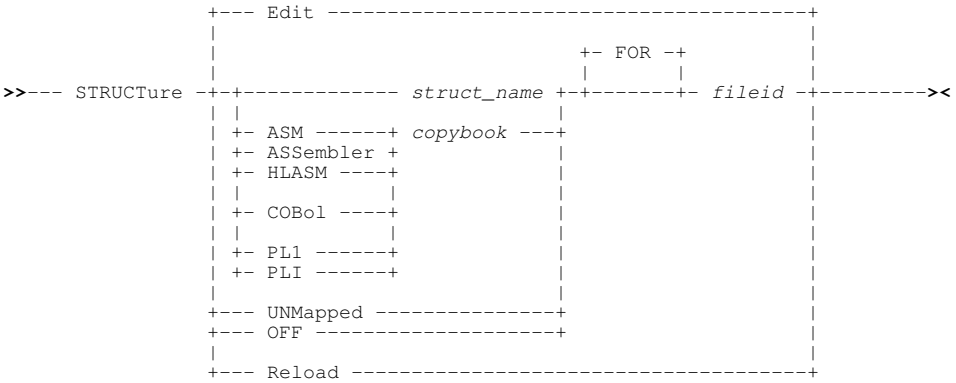
Examples:

```
SQL -SSN=DB8G  select * from dsn810.emp
```
Display all entries in the table DSN810.EMP of DB2 subsystem DB8G.

```
SQL -SSN=CBLA  -LIMIT=200  < CBL.SQL.CTL(TAB0326)
```
Execute in subsystem CBLA, all SQL statements provided via library mamber CBL.SQL.CTL(TAB0326) and limit the number list rows displayed to 200.

STRUCTURE

Syntax:



Description:

STRUCTURE performas the same functionality as the SDE SET STRUCTURE primary command.

SVC

Syntax:

Description:

For MVS only, use the SVC command to display the CBLVCAT SVC window containing the current status of the CBLVCAT Interactive (VCI) SVC required for LISTVCAT operations.

The CBLVCAT SVC window may also be opened via the System menu of the SELCOPY/i main window menu bar.

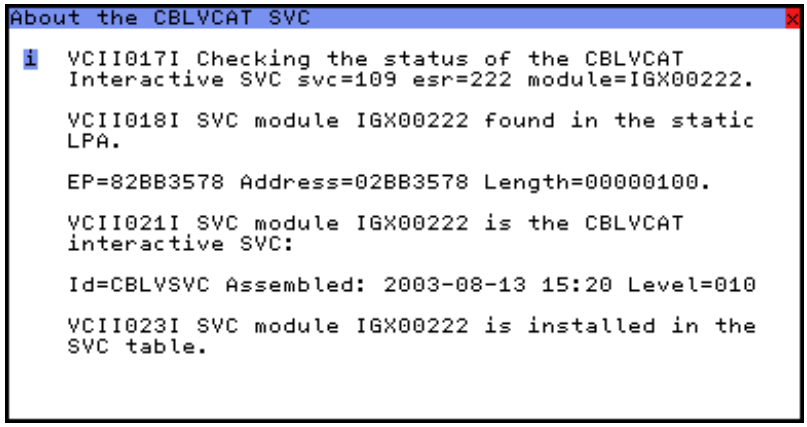


Figure 320. CBLVCAT SVC Window.

SYSAPF

Syntax:

```
>>-- SYSAPF ------------------------------------------------------------------><
```

Description:

Use the SYSAPF command to open the APF List window. (Not applicable to CMS and VSE systems.)

The APF List window may also be opened via the System menu of the SELCOPYi main window menu bar.

SYSCOMMAND

Syntax:

```
>>--+- SYSCOMMAND -+--- command ----------------------------------------------><
    |               |
    +- SYS ---------+
    |               |
    +- SYSTEM ------+
    |               |
    +- TSO ---------+
    |               |
    +- CMS ---------+
    |               |
    +- DOS ---------+
```

Description:

Pass the command directly to the local CMS or TSO environment for execution.

When a command is issued in a SELCOPY/i window, the following occurs:

1. If the command is recognised as a SELCOPY/i command it is executed by SELCOPY/i.
2. If the command is not recognised as a SELCOPY/i command, it is passed to the CMS or TSO environment.

Parameters:

command
: Valid CMS or TSO command or expression.

Example:

`cms query dasd`
: Pass the command "query dasd" to CMS.

SYSI

Syntax:

```
>>--+- SYSI -----+-------------------------------------------------------------><
    |            |
    +- SYSINFO -+
```

Description:

Use the SYSI command to open the Operating System window.

The System Information window may also be opened via the System menu of the SELCOPY/i main window menu bar.

SYSLL

Syntax:

```
>>-- SYSLL ----------------------------------------------------------------><
```

Description:

Use the SYSLL command to open the Link List window. (Not applicable to CMS and VSE systems.)

The Link List window may also be opened via the System menu of the SELCOPY/i main window menu bar.

SYSLPA

Syntax:

```
>>-- SYSLPA ---------------------------------------------------------------><
```

Description:

Use the SYSLPA command to open the LPA Modules window. (Not applicable to CMS and VSE systems.)

The LPA Modules window may also be opened via the System menu of the SELCOPY/i main window menu bar.

SYSMENU

Syntax:

```
>>-- SYSMENU --+--------------+-------------------------------------------><
               |              |
               +- windowname -+
```

Description:

Use the SYSMENU command to open the System Menu for the specified window.

The System Menu may also be opened via the System Menu button of a window.

Parameters:

windowname

The window name of the window for which the system menu is to be opened. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

SYSPGM

Syntax:

```
>>-- SYSPGM ------------------------------------------------------------><
```

Description:

Use the SYSPGM command to open the Loaded Programs window.

The Loaded Programs window may also be opened via the System menu item of the SELCOPY/i main window menu bar.

SYSSTOR

Syntax:

```
>>-- SYSSTOR -----------------------------------------------------------><
```

Description:

Use the SYSSTOR command to open the Storage Statistics window.

The Storage Statistics window may also be opened via the System menu of the SELCOPY/i main window menu bar.

SYSTASK

Syntax:

```
>>-- SYSTASK -----------------------------------------------------------><
```

Description:

Use the SYSTASK command to open the Task List window. (Not applicable to CMS and VSE systems.)

The Task List window may also be opened via the System menu of the SELCOPY/i main window menu bar.

TASK

Syntax:

```
>>-- TASK --- pgmname ---+------------------+---+----------------+---><
                         |                  |   |                |
                         +-- -LIB libpath --+   +-- -PARM parm --+
```

Description:

For MVS only, use the TASK command to start a program as a sub-task of SELCOPY/i.

TASK commands are generated by the CBLe REXX macro, JCLCMX, to run non-SELCOPY job steps of an MVS batch job in the environment in which SELCOPY/i is being executed (i.e. TSO or VTAM).

Parameters:

pgmname
: The name of the program load module to be executed.

-LIB *libpath*
: A list of load libraries to be included before the current environment's search library chain. This is equivalent to supplying a JCL STEPLIB statement in a batch job and so may be used to define the location of the program module to be executed plus any modules called by the program.

  Libpath may be one of the following:

  - ◊ A DDname which has been pre-allocated to one or more load libraries.
  - ◊ One or more load library DSNs separated by '**,**' (commas), '**;**' (semi-colons) or ' ' (blanks).
    Note that if blanks are used, quotes must also be used to delimit the list of DSNs, not the individual DSNs.

-PARM *parm*
: Parameter string to be passed to the program. This is equivalent to supplying the PARM parameter on an JCL EXEC statement in a batch job.

  If the parm string contains blanks, then quotes must be used to delimit the parm string.

Examples:

```
TASK  TRSMAIN   PARM='UNPACK'
```
: Start program TRSMAIN to unpack a tersed data set.
  Relevant INFILE and OUTFILE ddnames must be allocated before executing this command. (See the CBLe command ALLOCATE.)

```
TASK  MYPROG  -LIB "SYS7.DEV.MYLIB.LOAD  SYS4.USER.ROUTINES.X01323"
```
: Include the specified libraries at the start of the load library search chain then execute program MYPROG.

TOP

Syntax:

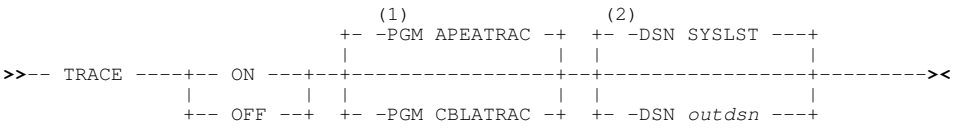
```
>>--- TOP ------------------------------------------------------------------><
```

Description:

Use the TOP command to display the top lines of the data in the focus window. The first line of the data becomes the first line of the display area.

TRACE

Syntax:



Notes:

1. Default -PGM program name is APEATRAC unless INI option variable TRACE.PROGRAM is assigned. If so the value of option TRACE.PROGRAM is used as the default.
2. Parameter -DSN is mandatory for z/OS systems unless INI option variable TRACE.DATASET is assigned. If so the value of option TRACE.DATASET is used as the default. For z/VSE systems, output trace records are always written to SYSLST.

Description:

TRACE will start or stop the SELCOPYi trace facility.

The trace facility is used to generate formatted or unformatted trace output for trouble shooting internal SELCOPYi issues and, therefore, should only be started if requested to do so by CBL support. TRACE executes program modules ZZSCTRAC or ZZSATRAC (alias CBLATRAC and APEATRAC respectively) to output trace records to a specified data set (or SYSLST fo z/VSE).

The trace facility is primarily used to generate formatted and/or unformatted trace output for trouble shooting internal SELCOPYi issues and, therefore, should only be started if requested to do so by CBL support. The trace facility is started and stopped using SELCOPYi primary command, TRACE.

TRACE ON will start output of SELCOPYi trace records to the nominated output dataset, whose DSN is specified as *outdsn*. Furthermore, in z/OS systems, if DDname CBLILIBT is allocated then additional trace records relating to SELCOPYi's internal function calls to library activation/deactivation, are written to the dataset allocated to CBLILIBT.

Both *outdsn* and CBLLIBT must represent existing, physical sequential datasets which have been allocated as LRECL=256, RECFM=VB.

TRACE OFF will stop output of SELCOPYi trace records.

Parameters:

`ON | OFF`
: Set the trace facility on or off.

`-PGM APEATRAC | CBLATRAC`
: Determines which of the SELCOPYi trace programs will be used to write trace output.

  APEATRAC will output trace records as text whereas CBLATRAC will output formatted records which are to be viewed using a SELCOPYi SDO structure. Note that CBLATRAC output is not supported in z/VSE.

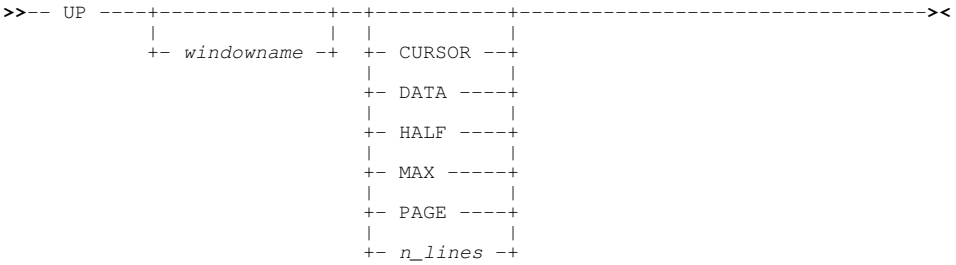
  The default is the value assigned to SELCOPYi INI variable TRACE.PROGRAM. Otherwise, if this variable is not set, the default is APEATRAC.

`-DSN outdsn`
: Applicable only to z/OS, *outdsn*, specifies the DSN of the cataloged data set to which trace output records will be written.

  The default for z/VSE is SYSLST. The default for z/OS is the value assigned to SELCOPYi INI variable TRACE.DATASET. Otherwise, if this variable is not set, an error will occur.

UP

Syntax:



Description:

Scroll the view of the data within the specified window upwards towards the top of the displayable data.

The extent by which data is scrolled is determined by the CURSOR, DATA, HALF, PAGE, MAX or *n_lines* parameter which may be specified using any one of three methods determined in the following order of precedence:

1. The scrolling command verb, UP, and one of these scrolling parameters is explicitly specified on the command line.
2. The scrolling parameter is specified on the command line and a PFKey assigned to UP is actioned.
   Note that the contents of a command line are appended to the command stream assigned to a PFKey when that PFKey is actioned.
3. No scrolling parameter is specified, so the current value of the "Scroll>" field is used.
4. No scrolling parameter is specified and no "Scroll>" field is present, so a defualt of one line is used.

By default this command is assigned to function key **PF7**.

Parameters:

windowname
: The window name of the window in which the display is to be scrolled. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

CURSOR
: The line on which the cursor is positioned becomes the last line of the scrolled display.
If the cursor is positioned outside the display area or on the first line within the display area, then UP PAGE is executed instead.

DATA
: Scroll up to display one page (display window depth) less one line of data.
The first line in the current display area becomes the last line of the scrolled display.

HALF
: Scroll up half a page of data.
The line that is half way down the page of data in the current display area becomes the last line of the scrolled display.

MAX
: Scroll up to display the first page of data.
The first displayable line becomes the first line of the scrolled display.

PAGE
: Scroll up to display the next whole page of data.
The line before the first line of the current display area becomes the last line of the scrolled display.

n_lines
: Scroll up a specified number of lines.
The line that is *n_lines* lines above the current line becomes the first line of the scrolled display.

VCAT

Syntax:

Description:

Use the VCAT command to open the Execute CBLVCAT window and optionally execute CBLVCAT control statements.

The Execute CBLVCAT window may also be opened via the File menu of the SELCOPY/i main window menu bar.

Parameters:

`cblv_syntax`
: Valid CBLVCAT syntax to be executed when the Execute CBLVCAT window is opened. Refer to the CBLVCAT User Manual for command reference.

  Note: The separator character, which by default is "!" (exclamation mark), may be used to enter multiple CBLVCAT operations on a single control statement.

  Currently, CBLVCAT control statements are restricted to a maximum length of 71 characters.

  This parameter is placed in the first VCAT command line field of the Execute CBLVCAT window.

`<`
: CBLVCAT input control statements will be passed from the data set referenced by cblv_ctl, to the Execute CBLVCAT window for CBLVCAT execution.

`cblv_ctl`
: The data set name of an MVS sequential data set or PDS and member, VSE LIBR lib.sublib and member or CMS fileid containing CBLVCAT control statements.

  This parameter, together with <, is placed in the first VCAT command line field of the Execute CBLVCAT window.

Examples:

`V q cblname`
: Generate CBLVCAT Query CBLNAME report.

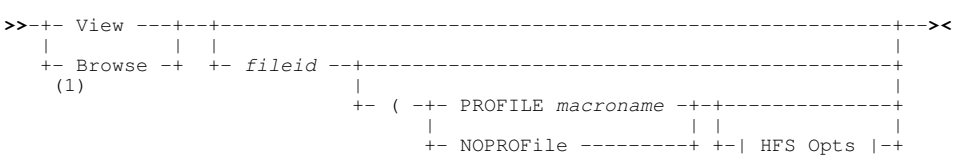
`V option pw 133 !report vcat dsn type component alloc3 !lc key=cbl type=c`
: Generate CBLVCAT catalog report.

`V < CBL.VVC.CTL(REPVTOC)`
: Generate CBLVCAT report by executing control statements located in PDS member REPVTOC of CBL.VVC.CTL.

`V < PRD2.CBLVCAT.REPVTOC.CTL`
: Generate CBLVCAT report by executing control statements located in VSE LIBR member REPVTOC of sublibrary PRD2.CBLVCAT.

VIEW

Syntax:



Notes:

1. BROWSE is a synonym of VIEW for VSE and CMS systems only.

Description:

Use the VIEW command to open a CBLe text edit window view to edit a file in read only mode.

If the CBLe text editor main window has been stopped, VIEW will start the CBLe main window before opening the read only edit view of the requested file.

A file edited in read only mode has **(Read Only)** in the title bar of the document window view.

Read-write edit may be invoked using the SELCOPYi EDIT command.

Parameters:

fileid
The fileid of the file to be viewed.

For **z/OS**, *fileid* may be any of the following:

- ◊ The DSN of a physical sequential data set.
- ◊ The DSN of a VSAM (KSDS, ESDS, RRDS, VRDS or LDS) data set.
- ◊ The library DSN and parenthesised member name of a PDS or PDSE library member.
- ◊ The library DSN, parenthesised member name and absolute or relative number of a PDSE library member generation as described under z/OS PDSE Library Member Generations. (PDSE version 2 with MAXGENS only.)
- ◊ The name of a member to be edited from the same PDS or PDSE library as the member displayed in the current text edit window view. If no text edit window view is open and *fileid* is an IDCAMS alias name, then data belonging to the aliased dataset is viewed. Otherwise, it is treated as an HFS/ZFS file name in the user's current working directory.
- ◊ The name of an HFS/ZFS file system fileid.

For **VSE**, *fileid* is the member name of a LIBR library in lib.sublib.mn.mt format.

For **CMS**, *fileid* is a CMS fileid.

Note that if *fileid* is not specified, then EDIT is executed for the fileid specified by the SELCOPYi INI variable System.CmdTEXT. If System.CmdTEXT has not been set, then no action is taken.

PROFILE *macroname*
The REXX edit macro to be executed as the profile when editing the file.

This macro overrides use of the default profile macro defined by the SELCOPY/i INI option Edit.DefProfile=*macroname* and/or the CBLe command SET DEFPROFILE (default PROFILE.)

The macro name must exist in a library within the CBLe macro path.

The PROFILE option only affects the profile for the file currently being added to the ring, and does not affect the profile to be used when additional files are added to the edit ring later in your edit session.

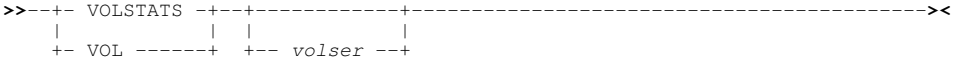
NOPROFILE
Suppresses use of a profile macro when editing the file.

The NOPROFILE option only affects the file currently being added to the ring, and does not affect the profile to be used when additional files are added to the ring later in your edit session.

HFS Opts
See Text Edit primary command EDIT for supported HFS parameters.

VOLSTATS

Syntax:


>>--+- VOLSTATS -+--+------------+---><
| | | |

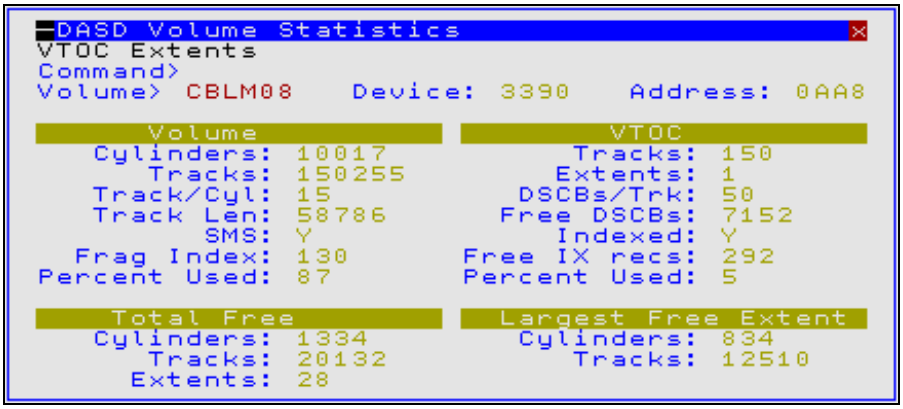


Figure 321. DASD Volume Statistics Window.

Parameters:

volser
The volume serial number of the volume for which statistics are to be displayed.
This parameter is placed in the Volume field of the DASD Volume Statistics window.

Examples:

```
 VOLSTATS  Z2RES1
```

WINDOWLIST

Syntax:

```
>>--+- WINDOWLIST -+-------------------------------------------------------><
    |               |
    +- WL ----------+
```

Description:

Use this command to open the Window List window which lists all open windows.

The Window List window may also be opened via the **Window** item of the SELCOPY/i main window menu bar.

Position the cursor on an entry in the window list and hit <Enter> or, if configured, double-click the left mouse button on the list entry, to close the window list and make the selected window the focus window.

WINDOWNAMES

Syntax:

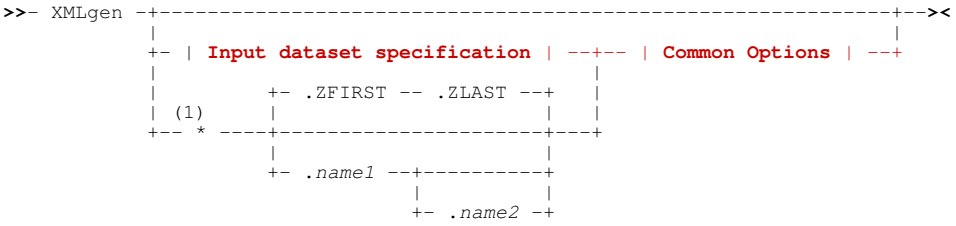
```
>>--+- WINDOWNAMES -+-------------------------------------------------------><
    |                |
    +- WN -----------+
    |                |
    +- NAMES --------+
```

Description:

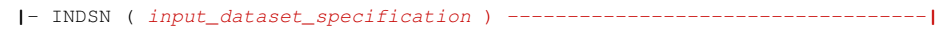
This command toggles the display of window names in the title bar. When the window names are displayed, they are shown left justified in the title bar followed by a colon.

XMLGEN

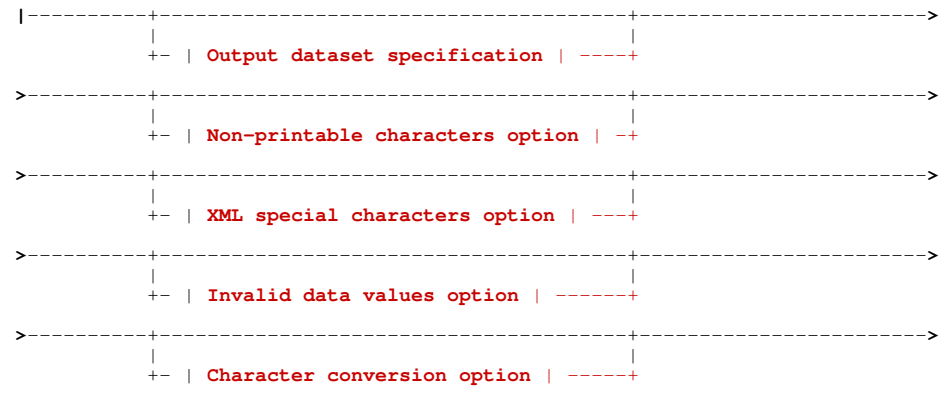
Syntax:



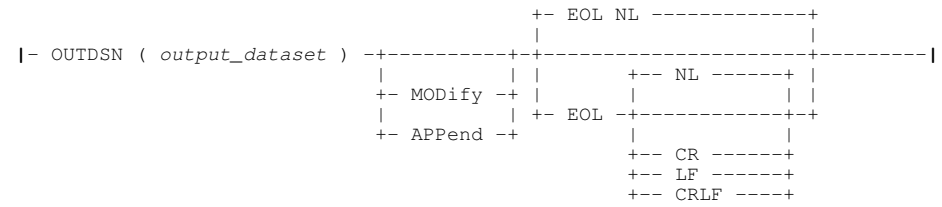
Input dataset specification:



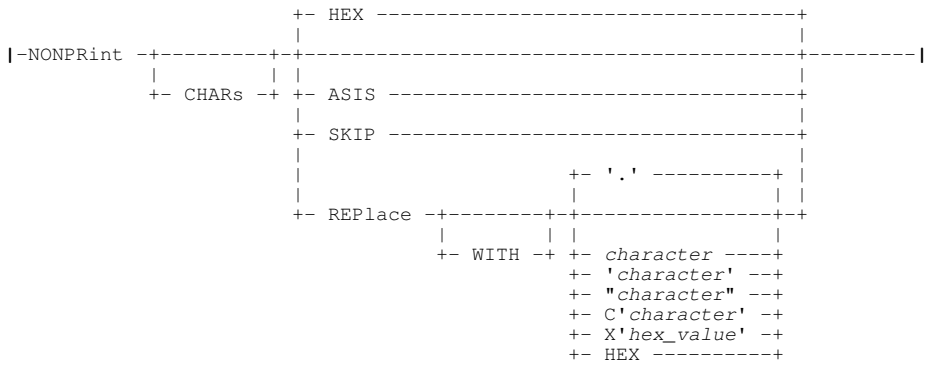
Common Options:



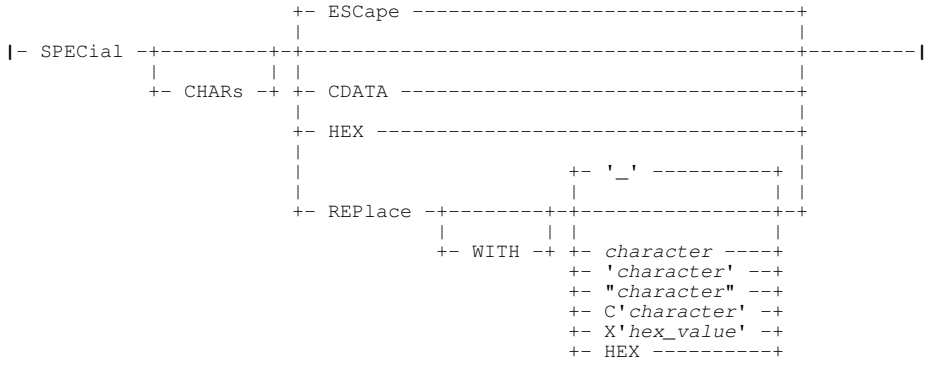
Output dataset specification:



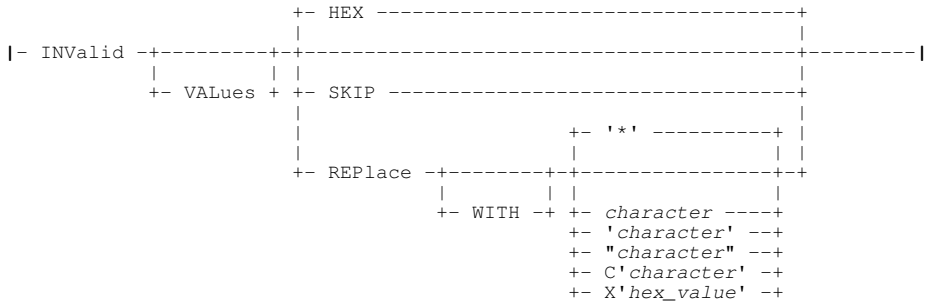
Non-printable characters option:



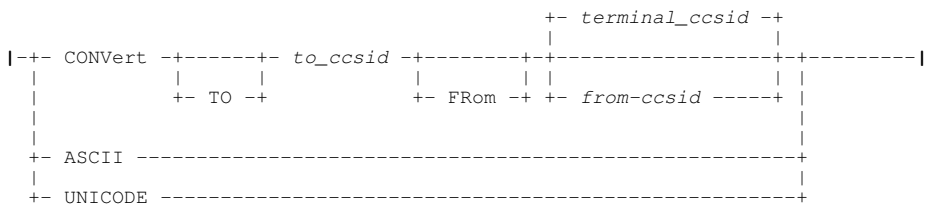
XML special characters option:



Invalid data values option:



Character conversion option:



Notes:

1. Asterisk (*) is optional if any other parameter is specified.

Description:

XMLGEN is used to open the general XML Generation panel or to immediately produce an exportable copy of structured data as extensible markup language (XML) text. The structured data may belong to a specific structured data set or the current SDE view.

If asterisk (*), or any other parameter other than INDSN, is specified, then data in the current SDE edit or browse view will be used as input for XML generation. If no parameters are specified, the **XML Generation** panel is opened.

XMLGEN reads an input structured data and writes an output text dataset consisting of XML tags and tag content. The tag names correspond to the field names of the copybook/structure applied to the input dataset and the tag content to the field values

expressed in character format.

The hierarchy of record types, group fields and elementary fields in the structure is represented by a corresponding nesting of XML tags, with the output enclosed within **<INPUT>** and **</INPUT>** tags.

The XMLGEN command may be executed in the foreground or via SDEIN input to program SDEAMAIN for batch processing.

During foreground execution a progress window is displayed showing input and output record counts, updated every second, which allows the user to interrupt processing before completion using the attention key.

Parameters:

Input dataset specification (INDSN)

Use of an XMLGEN input data set nominates a specific data set from which records are to be selected for XML generation.

The XMLGEN input dataset may be specified explicitly in the command as the argument of the **INDSN** keyword. If the **INDSN** keyword is not present in the command then the contents of the current SDE structured dataset browse or edit view are used. If there is no current structured dataset the XML Generation panel is opened.

INDSN (*input_dataset_specification*)

The input dataset specification is in the form of a structured edit BROWSE command (the BROWSE command verb is not required) which must be enclosed in parentheses following the INDSN keyword.

BROWSE keyword options such as **FROM**, **FOR**, **FILTER** and **VIEW** may be specified to limit the records from the input dataset which will be copied to the output dataset.

*

Required only if no other XMLGEN parameters are specified in order to immediately generate XML for data from the current SDE view (using defaults) as opposed to opening the general purpose XML Generation panel or the SDE XML Generation Panel as appropriate.

Start/End Line labels (.name1/.name2)

Applicable only to XML generation from data in the current SDE view, start and end line labels may be used to select a range of data records to be processed.

.name1

Corresponds to a label name *.name1* that identifies the first line in a range of SDE edit/browse lines. The preceding "." (period/dot) is mandatory.
Default is .ZFIRST.

.name2

Corresponds to a label name *.name2* that identifies the last line of a range of SDE edit/browse lines. The preceding "." (period/dot) is mandatory.
.name2 may occur on a line with a lower line sequence number than *.name1*. This is functionally equivalent to specifying *.name2* before *.name1* on the XMLGEN command.
Default is .ZLAST.

Output dataset specification (OUTDSN)

The XMLGEN output dataset may be specified explicitly in the command as the argument of the **OUTDSN** keyword. If the **OUTDSN** keyword is not present in the command then the value of the INI file variable **SDE.XMLGENOUTDSN** is used if it exists, otherwise a default dataset name **userid.ZZS.XMLGEN** is used.

OUTDSN (*output_dataset*)

The output dataset name. Parentheses around the dataset name are accepted but not required. If this dataset exists its organisation may be sequential, a partitioned dataset member, VSAM (except LDS and KSDS) or HFS (ZFS). If it does not exist and XMLGEN is executing in batch the command terminates with an error message. If it does not exist and XMLGEN is executing interactively the user will be asked to allocate it (unless the name represents an HFS file in which case it will be implicitly defined).

This parameter may also be a DD name. If **output_dataset** consists of 8 or fewer characters and represents an allocated DD name then this allocated dataset is used for output.

MODIFY | **APPEND**

The output will be appended to the dataset if it exists (and is not partitioned). If this keyword is not specified the output will overwrite any existing dataset content.

EOL NL|CR|LF|CRLF

HFS file end of line specification. This parameter is accepted but ignored if the output dataset is not an HFS file. The values here are specified in EBCDIC, but if the output is subject to character conversion, the line end characters will also be converted. Note that ASCII does not have a NL (newline) character so if the output is being converted to a non-EBCDIC CCSID NL is changed to CR.

NL	X'15'	EBCDIC New Line. This is the default for EBCDIC output to an HFS file.
CR	X'0D'	EBCDIC Carriage Return.
LF	X'25'	EBCDIC Line Feed.
CRLF	X'0D25'	EBCDIC Carriage Return Line Feed.

Non-printable characters option (NONPRINT)

Since XMLGEN output is supposed to be in a portable character format, this option is required to specify how non-printable characters are dealt with.

NONPRINT

This keyword starts the specification of the non-printable characters option. The default option is **HEX**.

HEX

If a character field contains a non-printable character output the whole field in hex string format. For example a character field length 4 containing X'FFFFFFFF' would have its value represented as

```
X&apos;FFFFFFFF&apos;
```

If a non-printable character is found in a character field and this option is in effect the field XML tag will have the attribute **NONPRINT_CHAR="HEX"**.

ASIS

No special action is taken. All input bytes are copied to the output XML tag value. If a non-printable character is found in a character field and this option is in effect the field XML tag will have the attribute **NONPRINT_CHAR="ASIS"**.

SKIP

The field value is skipped. If a non-printable character is found in a character field and this option is in effect the field XML tag will have the attribute **NONPRINT_CHAR="SKIP"** and no content.

REPLACE

Each non-printable character in a character field is replaced with the specified value. If a non-printable character is found in a character field and this option is in effect the field XML tag will have the attribute **NONPRINT_CHAR="REPLACE"**. The default replace character is the period (full-stop) ".". The replacement character can be specified as:

character

The actual replacement character. If no character follows REPLACE then period (.) is assumed. If the character is a lower case letter it will be changed to upper case.

'character' | *"character"*

The actual replacement character in single quotes (apostrophes) or double quotes. If the character is a lower case letter it will be translated to upper case.

c'*character*' | c"*character*"

The actual replacement character in single quotes (apostrophes) or double quotes with a **c** or **C** prefix. No case translation takes place.

x'*hex_value*' | x"*hex_value*"

The replacement character specified as a hexadecimal value.

HEX

This keyword does not represent a replacement character but requests that any substring of non-printable characters found in a character field is replaced with its value in hexadecimal format inside **<HEX> </HEX>** tags. For example a character field length 4 containing X'C1C2FFC3' would have its value represented as

```
AB<HEX>X&apos;FF&apos;</HEX>C
```

In this case the field XML tag wiill have the attribute **NONPRINT_CHAR="REPLACE_HEX"**.

XML special characters option (SPECIAL)

XML specifies 5 characters as of special syntactical significance. These characters are used to delimit XML constructs and must not appear as themselves in tag values. XML provides an escape sequence (character reference) which can be used to represent these special characters in tag values.

The XML special characters are:

Character	Name	Escape sequence
<	Less than	<
>	Greater than	>
'	Apostrophe	'
"	Double quote	"
&	Ampersand	&

This option provides a way of dealing with any of the XML special characters found in character data fields.

SPECIAL

This keyword starts the specification of the XML special characters option. The default option is **ESCAPE**.

ESCAPE

If a character field contains an XML special character replace it with its XML escape sequence. For example a character field length 4 containing 'A<>B' would have its value represented as:

```
A&lt;&gt;B
```

If an XML special character is found in a character field and this option is in effect the field XML tag will have the attribute **SPECIAL_CHAR="ESCAPE"**.

HEX

If a character field contains an XML special character output the whole field in hex string format. For example a character field length 4 containing 'A<>B' would have its value represented as

```
X&apos;C14C6EC2&apos;
```

If an XML special character is found in a character field and this option is in effect the field XML tag will have the attribute **SPECIAL_CHAR="HEX"**.

CDATA

If a character field contains an XML special character output the whole field asis in an XML character data (CDATA) section. CDATA sections in an XML document represent unparsed character data. For example a character field length 4 containing 'A<>B' would have its value represented as

```
<![CDATA[A<>B]]>
```

If an XML special character is found in a character field and this option is in effect the field XML tag will have the attribute **SPECIAL_CHAR="CDATA"**.

REPLACE

Each XML special character in a character field is replaced with the specified value. If an XML special character is found in a character field and this option is in effect the field XML tag will have the attribute **SPECIAL_CHAR="REPLACE"**. The default replace character is the underscore "_". The replacement character can be specified as:

character
: The actual replacement character. If no character follows REPLACE then underscore (_) is assumed. If the character is a lower case letter it will be changed to upper case.

'character' | *"character"*
: The actual replacement character in single quotes (apostrophes) or double quotes. If the character is a lower case letter it will be translated to upper case.

c'*character*' | c"*character*"
: The actual replacement character in single quotes (apostrophes) or double quotes with a **c** or **C** prefix. No case translation takes place.

x'*hex_value*' | x"*hex_value*"
: The replacement character specified as a hexadecimal value.

HEX
: This keyword does not represent a replacement character but requests that any substring of XML special characters found in a character field is replaced with its value in hexadecimal format inside **<HEX> </HEX>** tags. For example a character field length 4 containing 'A<>B' would have its value represented as:

```
A<HEX>X&apos;4C6E&apos;</HEX>B
```

In this case the field XML tag wiill have the attribute **SPECIAL_CHAR="REPLACE_HEX"**.

Invalid data values option (INVALID)

Non-character fields in structured data files may have invalid values which cannot be converted to character format. For example, a field defined as containing packed decimal (COBOL COMP-3) data may not contain a valid packed decimal value. This option provides a way of specifying how such fields are represented in the XML output.

INVALID

This keyword starts the specification of the invalid data values option. The default option is **HEX'**.

HEX

If a non-character field contains an invalid data value output the whole field in hex string format. For example a packed decimal field length 4 containing X'00000000' would have its value represented as

```
X&apos;00000000&apos;
```

If an invalid value is found in a non-character field and this option is in effect the field XML tag will have the attribute **INVALID_DATA="HEX"**.

SKIP

The field value is skipped. If an invalid data value is found in a non-character field and this option is in effect the field XML tag will have the attribute **INVALID_DATA="SKIP"** and no content.

REPLACE

The invalid field data value is replaced with the specified value. If an invalid field data value is found in a non-character field and this option is in effect the field XML tag will have the attribute **INVALID_DATA="REPLACE"**. The default replace character is the asterisk "*". The replacement character can be specified as:

character
: The actual replacement character. If no character follows REPLACE then asterisk (*) is assumed. If the character is a lower case letter it will be changed to upper case.

'character' | *"character"*
: The actual replacement character in single quotes (apostrophes) or double quotes. If the character is a lower case letter it will be translated to upper case.

c'*character*' | c"*character*"
: The actual replacement character in single quotes (apostrophes) or double quotes with a **c** or **C** prefix. No case translation takes place.

x'*hex_value*' | x"*hex_value*"
: The replacement character specified as a hexadecimal value.

Character conversion option (CONVERT/ASCII/UNICODE)

Since the purpose of XMLGEN is to produce a portable export version of the data in a z/OS mainframe structured data file, and the output is character data, the coded character set identifiers (CCSIDs) of the input, output and of the XMLGEN internal constants themselves are of significance.

Even if the input and output is coded in an EBCDIC CCSID, these may differ, and both may differ from the CCSID of the XMLGEN internal constants. Since some of the special characters used in XML have different code points in different EBCDIC CCSIDs (for example square brackets) these must be dealt with consistently to produce correct XML output.

XMLGEN uses the z/OS character conversion support supplied by IBM modules CUNLINFO (for obtaining CCSID information) and CUNLCNV (for character conversion from one CCSID to another).

The internal XMLGEN CCSID (that of the constants used to build the XML syntax) is CCSID 285 (EBCDIC, SBCS UNITED KINGDOM).

XMLGEN assumes a default CCSID as follows:

Interactive
: When executed interactively XMLGEN uses as default input CCSID that of the user's 3270 terminal.

Batch
: When executed in batch XMLGEN uses as default input CCSID the value of the INI file variable **SDE.CCSID**. This variable is set automatically to the user's 3270 terminal CCSID (if not already set) during an interactive session. It can also be set using the structured data SET CCSID command.

If no explicit conversion is specified the XML output dataset is produced using the default CCSID and the input dataset character fields are assumed to be in the same CCSID. The internal XMLGEN constants are converted from internal CCSID 285 to the default CCSID.

CONVERT
: Use this keyword to request character CCSID conversion.

to_ccsid
: The CCSID of the output XML text dataset. Internal XMLGEN character literals and input character data fields (and HFS line end characters if used) are converted to this CCSID.

from_ccsid
: The input character data fields are converted from this CCSID. If this parameter is not supplied the default input CCSID is used.

ASCII
: Convert the output to ASCII. This is equivalent to specifying **CONVERT TO 819**. CCSID 819 is ISO 8859-1 ASCII.

UNICODE
: Convert the output to UNICODE (UTF-16). This is equivalent to specifying **CONVERT TO 1200**. CCSID 1200 is the IBM bigendian UTF-16 CCSID which is automatically transformaed to the most recent UTF-16 standard.

Ouput line splitting option (SPLIT)

For each elementary input field XMLGEN builds one output record containing the field start tag, the field value (possibly with embedded HEX tags and special character escape sequences), and the field end tag. Depending on the options chosen and the nature of the input data, relatively long output records may result. If an output record is longer than the allocated logical record length of the output dataset this option controls how XMLGEN deals with the long output record.

NOSPLIT
: Do not split the output record. Rather than truncate the output record XMLGEN terminates with an error message. This is the default.

SPLIT
: Split the output record breaking it up into as many logical records as necessary. Records are split at the logical record length irrespective of the record content.

Redefined field selection option (REDEFINES)

If the structure defined for the input dataset contains redefined fields this option controls whether the field redefinitions are output.

NOREDEFINES
Do not output the field redefinitions. This is the default.

REDEFINES
Output all field redefinitions.

Unnamed field option (FILLER)
If the structure defined for the input dataset contains unnamed or FILLER fields this option controls whether these fields are output.

NOFILLER
Do not output unnamed or FILLER fields. This is the default.

FILLER
Output all unnamed or FILLER fields.

Output comment header block option (HEADER/NOHEADER)
This option controls whether an XML style comment block is generated at the top of the output dataset. This contains information about the host operating system, the id of the creator of the output file and the creation date and time, and details of any character conversion performed on the output character data.

COMMENTS
Output a comment block. This is the default.

NOCOMMENTS
Do not output a comment block.

Ignore group hierarchy option (GROUP/ELEMENTARY)
This option controls whether elements of a group field are output as children of their parent group tag.

GROUP
Group field tags are included with elements as children. This is the default.

ELEMENTARY
Group field tags are not included. All elements are output at the top level within the record-type.

XML nested tag indentation option (INDENT)
Nested output XML tags corresponding to the hierarchy of group and elementary data fields in the input structure are indented by a default of one column for each data item level. This option allows the specification of a different indentation value.

INDENT *n_cols*
The indentation value (default 1).

Limit number of input lines (LIMIT)
The number of input records, record segments or DB2 table rows processed may be restricted using this option.

LIMIT *n_lines*
The maximumm number of input lines. Default is all lines.

Output view option (BROWSE/EDIT/NOVIEW)
When XMLGEN is run interactively this option allows the user to request to view the output when the command completes.

BROWSE
Browse the output XML dataset. This is the default when run interactively.

EDIT
Edit the output XML dataset using the SELCOPY/i text editor.

NOVIEW
Do not view the output XML dataset. This option is forced when run in batch.

Uppercase tag name option (TAGUPPER/NOTAGUPPER)
This option controls whether tag names generated from the structure field names are upper cased. The default action is controlled by the global SDE option, NAMECASE.

TAGUPPER
Tag names are uppercased.

NOTAGUPPER
Tag names are not uppercased. Tags match the field names with no upper case translation.

Examples:

In the following example the COBOL copybook CBL.COB(XMLEXAMP) is used to map the 3 records in the file CBL.XMLEXAMP

The content of the COBOL copybook CBL.COB(XMLEXAMP) is shown as it would appear in text edit:

```
-CBL.COB(XMLEXAMP)     80 F PDSE   Size=10   Alt=0,0;1
      <---+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000000 * * * Top of File * * *
000001       ** CBL.COB(XMLEXAMP) ***     L=001 --- 2013/10/08 14:45:56  (CBL)
000002
000003        01  XMLExample.
000004            05  Employee.
000005                07  FirstName      Pic x(20).
000006                07  LasttName      Pic x(20).
000007            05  Age                Pic s9(3) comp.
000008            05  Salary             Pic s9(7)v99 comp-3.
000009 * * * End of File * * *
```

The content of the file CBL.XMLEXAMP is shown as it would appear in structured data edit with HEX ON in effect to show the content of numeric fields:

```
-Edit   CBL.XMLEXAMP using CBL.COB(XMLEXAMP)          47 F SEQ
00000000 *** Top of Data ***
Record type: XMLEXAMPLE   Fixed(47) Offset=0 Data elements=6

         FIRSTNAME            LASTTNAME               AGE      SALARY
         #3                   #4                       #5          #6
         <---+----1----+----> <---+----1----+----> <---+> <---+----1>
00000001 John                 Doe                      52    33000.00
         D9894444444444444444 C9844444444444444444     03       03000
         1685000000000000000 4650000000000000000      04       0300C

00000002 Amy                  Johnston                 28    41500.00
         C9A4444444444444444 D989AA99444444444444     01       04500
         1480000000000000000 1685236500000000000      0C       0100C

00000003 Freda                Bloggs                   39    27800.00
         C9888444444444444444 C9988A44444444444444     02       02800
         6954100000000000000 2367720000000000000      07       0700C

00000004 *** End of Data ***
```

The following XMLGEN command then produces the output file CBL.XMLGEN:

```
   xmlgen  indsn(cbl.xmlexamp using cobol cbl.cob(xmlexamp))
          outdsn(cbl.xmlgen)
          nocomment indent 3 edit
```

The XMLGEN output file is edited as a result of the EDIT keyword parameter in the command. Note that each record has an associated record type (level 1) tag XMLEXAMPLE with the SEQ_NUMBER attribute identifying the record number:

```
-CBL.XMLGEN     27990 V SEQ   Size=26   Alt=0,0;0
      <---+----1----+----2----+----3----+----4----+----5----+----6----+--
000000 * * * Top of File * * *
000001 <INPUT FILE="CBL.XMLEXAMP" FORMAT="CBL.COB(XMLEXAMP)">
000002    <XMLEXAMPLE SEQ_NUMBER="1">
000003       <EMPLOYEE>
000004          <FIRSTNAME>John</FIRSTNAME>
000005          <LASTTNAME>Doe</LASTTNAME>
000006       </EMPLOYEE>
000007       <AGE>52</AGE>
000008       <SALARY>33000.00</SALARY>
000009    </XMLEXAMPLE>
000010    <XMLEXAMPLE SEQ_NUMBER="2">
000011       <EMPLOYEE>
000012          <FIRSTNAME>Amy</FIRSTNAME>
000013          <LASTTNAME>Johnston</LASTTNAME>
000014       </EMPLOYEE>
000015       <AGE>28</AGE>
000016       <SALARY>41500.00</SALARY>
000017    </XMLEXAMPLE>
000018    <XMLEXAMPLE SEQ_NUMBER="3">
000019       <EMPLOYEE>
000020          <FIRSTNAME>Freda</FIRSTNAME>
000021          <LASTTNAME>Bloggs</LASTTNAME>
000022       </EMPLOYEE>
000023       <AGE>39</AGE>
000024       <SALARY>27800.00</SALARY>
000025    </XMLEXAMPLE>
000026 </INPUT>
000027 * * * End of File * * *
```

Unix System Services (USS) Commands

The CBLe text editor and SDE (Structured Data Environment) Edit support HFS files and the concept of a current working directory. This enables users to reference HFS files by an HFS path relative to the current working directory.

To fully support this functionality and assist with HFS file management for data edit, certain UNIX System Services commands are supported as part of the SELCOPY/i CLI command set. These commands are prefixed by "USS".

USS prefixed SELCOPY/i commands may only affect HFS path names and so specification of "/" (slash) within the path name or a leading "." (dot/period) in order to identify the fileid as an HFS path name is unnecessary.

Command	Description
USS CHDIR	Change the current working directory.
USS GETCWD	Display the current working directory.
USS LINK	Define a new HFS hard link to a file.
USS MKDIR	Define a new HFS directory.
USS REALPATH	Display the absolute HFS file path for a given relative HFS path.
USS RENAME	Rename an existing HFS file, hard link, symbolic link or directory.
USS RMDIR	Remove an existing, empty HFS directory.
USS STAT	Display status of a specified HFS path.
USS UNLINK	Remove an existing HFS file, hard link or symbolic link.

USS CHDIR

Syntax:

```
>>-- USS ----- CHDIR ------ hfs_path ----------------------------------------><
```

Description:

Change the current working directory.

USS CHDIR is equivalent to the USS shell command CD but without the additional options.

Parameters:

hfs_path
: An HFS path name representing a directory.

USS GETCWD

Syntax:

```
>>-- USS --+-- GETCwd --+-------------------------------------------------><
           |            |
           +--- PWD ----+
```

Description:

Display the current working directory. If executed from a CBLe or SDE edit view, output is displayed on the message line. Otherwise, output is displayed in a popup message window.

USS GETCWD is equivalent to the USS shell command PWD.

Parameters:

USS GETCWD has no parameters.

USS LINK

Syntax:

```
>>-- USS ----- LINK ----- old_hfs_path ---- new_hfs_path --------------------><
```

Description:

Create a hard link to an existing HFS file.

USS LINK is equivalent to the USS shell command LINK.

Parameters:

old_hfs_path
: An HFS path name representing a file. This may be the HFS file name, another hard link or a symbolic link. If *old_hfs_path* is a symbolic link, a hard link is created to the file that results from resolving the symbolic link.

new_hfs_path
: The HFS path name of the new hard link to the file data.

USS MKDIR

Syntax:

```
>>-- USS ----- MKDIR ------ hfs_path ----------------------------------------><
```

Description:

Create a new HFS directory.

USS MKDIR is equivalent to the USS shell command MKDIR but without the additional options.

Parameters:

hfs_path
: An HFS path name representing a directory.

USS REALPATH

Syntax:

```
>>-- USS ----- REALPATH --- hfs_path ----------------------------------------><
```

Description:

Display the absolute HFS path name for the specified (relative) HFS path name.

Parameters:

hfs_path
: Any HFS path name.

USS RENAME

Syntax:

```
>>-- USS ----- RENAME --- old_hfs_path ---- new_hfs_path --------------------><
```

Description:

Rename an existing HFS file, hard link, symbolic link or directory name.

USS RENAME is equivalent to the SELCOPY/i RENAME command except that rename arguments are always treated as HFS path names.

Parameters:

old_hfs_path
: An HFS path name representing a file, hard link, symbolic link or directory name.

new_hfs_path
: The new HFS path name.

USS RMDIR

Syntax:

```
>>-- USS ----- RMDIR ------ hfs_path ----------------------------------------><
```

Description:

Remove an existing, empty HFS directory.

USS RMDIR is equivalent to the USS shell command RMDIR except that, currently, no option exists to remove intermediate directory components.

Parameters:

hfs_path
: An HFS path name representing a directory.

USS STAT

Syntax:

```
>>-- USS ----- STAT ------- hfs_path ----------------------------------------><
```

Description:

Display the status of the specified HFS path name.

This includes the absolute HFS path name, type, file size, blocksize, format and permissions (octal).

Parameters:

hfs_path
: An existing HFS path name.

USS UNLINK

Syntax:

```
>>-- USS ----- UNLINK ----- hfs_path ----------------------------------------><
```

Description:

Unlink the specified HFS path name.

USS UNLINK is equivalent to the USS shell command UNLINK.

Parameters:

hfs_path
: An existing HFS path name representing a file name, hard link or symbolic link. Alternate path names to the same data are unaffected.

SELCOPYi VTAM commands

Commands may be passed to the SELCOPYi VTAM application via the system operator console.

In MVS, this is achieved using the MODIFY (F) JES command and the appropriate job name as follows:

```
MODIFY  CBLIVTAM,command
```

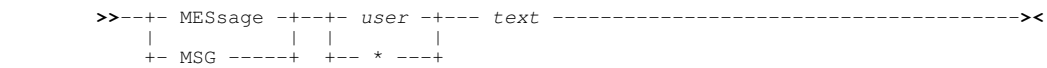
In VSE, this is achieved via an operator communications (OC) exit using the attention routine (AR) command MSG for the partition running SELCOPYi VTAM. e.g.

```
MSG     F8,DATA=command
```

Command	Description
MESSAGE	Send a text message to one or more users logged on to SELCOPYi VTAM.
QUERY	Query the SELCOPYi VTAM environment.
STOP	Stop SELCOPYi VTAM.

MESSAGE

Syntax:



Description:

Send a text message to a single user or all users logged on to SELCOPY/i VTAM. The message text will appear in a pop-up window at the user's terminal when a 3270 AID key is hit. (e.g. <Enter>, any PFKey, etc.)

Parameters:

user
: The user id of the user to whom the message is to be sent.
If "*" (asterisk) is specified, then the message is sent to all users who are logged on to SELCOPY/i VTAM.

text
: The message text.

Examples:

```
F CBLIVTAM,MSG JOHNB Please browse CBL.CMX(SKEL).

MSG F8,DATA=MESSAGE * Please logoff. SELCOPY/i VTAM will be stopped at 10:00.
```

QUERY

Syntax:

Description:

Query information about the SELCOPY/i VTAM environment. SELCOPY/i currently supports only one parameter (i.e. USERS) which identifies all users logged on to SELCOPY/i VTAM.

Parameters:

USERS
: Display information about users who are logged on to SELCOPY/i VTAM.

Examples:

```
MSG F8,DATA=Q
 ZZSV021I Applid CBLIVTAM has 2 active sessions
         User     Terminal Session
         JGE1     D20001   30000002
         NBJ1     D20101   12000003
```

STOP

Syntax:

```
>>-- STOP ------------------------------------------------------------------><
```

Description:

Stop the SELCOPY/i VTAM job.

Examples:

```
F CBLIVTAM,STOP

MSG F8,DATA=STOP
```

SELCOPYi Dump Files

SELCOPYi dump files are supported for SELCOPYi running in MVS environments only.

In order to assist CBL software engineers to correct any defects encountered in the SELCOPYi system and programs, SELCOPYi dump files exist to store formatted storage dumps.

By default, the **System.AbendTrap** variable is set **ON** in the SELCOPYi System INI file. Therefore, in the event of a program check or program abend occuring which ultimately halts the CBL interactive environment, a message is sent to the user and control is passed to SELCOPYi's abend handler routines.
If AbendTrap is set **OFF**, any abnormal program end is handled by the operating system.

Each time the SELCOPYi abend handler is called, a new dump file is allocated with DSN prefix qualifier(s) determined by the **System.DumpDSNPrefix** variable in the SELCOPYi User or System INI file.
The remainder of the dump file DSN is of the form **.Dyyyyddd.Thhmmssx**, representing the current local date and time.

Dump files are allocated as physical sequential data sets with DCB=(RECFM=VB,LRECL=256,BLKSIZE=0) and SPACE=(CYL,(9,5)).

If an abend is encountered in SELCOPYi, then please contact the CBL support desk via telephone on +44 1656 650692 or via email at support@cbl.com. A request to email the SELCOPYi dump file to CBL file is likely.

Appendix A - SELCOPYi Window Classes

The following table identifies the SELCOPYi Windows Classes.

Window Class Name	Window Class Description
VCIWMAIN	SELCOPYi main window
EDTWMAIN	CBLe main window
EDTWEDIT	Text Edit document window
EDTWFIND	Text Edit FIND dialog window
EDTWCHNG	Text Edit CHANGE dialog window
EDTWSORT	Text Edit SORT dialog window
EDTWFILL	Text Edit FILL dialog window
EDTWEMSG	Text Edit message window
EDTWHEXE	Text Edit line Hex Dump view
SDEWVIEW	Structured Data Edit document view
LISTFRAM	List window
LISTFILE	List File window
VCIWEXEC	CBLVCAT Interactive window
SDBWDBUG	SELCOPY Debug main window
SYSIN	SELCOPY Debug SYSIN Text Edit document view
SYSPRINT	SELCOPY Debug SYSLST/SYSPRINT Text Edit document view
WTOLOG	SELCOPY Debug WTO LOG Output Text Edit document view
SQLLOG	SELCOPY Debug DB2 SQL LOG Output Text Edit document view
TRACE	SELCOPY Debug TRACE Output Text Edit document view
STORAGE	SELCOPY Debug POS Hex Dump view
HTMWMAIN	Help window
VCIWDEFA	Allocate New NonVSAM Cataloged Dataset dialog window
VCIWDEFC	Define VSAM Cluster dialog windows
VCIWDFAL	Define VSAM Catalog ALIAS dialog window
WINWIPO0	Interactive Panel windows
WINWALID	Define PDS/PDSE Member ALIAS dialog window
WINWIEBC	Execute IEBCOPY dialog window
WLDIALOG	Window List window
SDEWFCOP	File Copy dialog window
CALCULAT	Rexx Calculator window
CALENDAR	Calendar window
HEXDUMP	Storage display window
SYSINFO	System Information window

Appendix B - List File Prefix Command Summary

See List Window Prefix Area for a description of the list window class prefix area and its features.

The following table is a summary of all the standard prefix commands supported by list data object windows, the Execute CBLVCAT window and File Search window.

A Open the Create Library Alias dialog window for Library Lists or the Define Catalog ALIAS for all other file lists.

AP Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.

AS Open an Associations list window for the entry.

B Open the CBLe text editor to to perform SDATA BROWSE on the entry.

C Open the File Copy dialog panel to copy the entry.

CF Open the Compare Files Panel for this entry, using the entry name as the New File field entry.

CL Open the Compare Libraries Panel for this entry, using the entry name as the New DSN field entry.

D Delete the entry. User will be prompted to verify the deletion.

E Open the CBLe text editor to edit the entry.

EU Open the SDE structured data editor to edit the entry in update mode only.

EX Execute the library member entry. (Invokes the TSO command, EXECUTE, using the entry name as input.
Supported in MVS TSO or ISPF environments only.

F Open the FSU - File Search/Update Window to perform an advanced search and optionally update the contents of the entry. Supported for MVS SELCOPY licensees only on all types of data set.

FO Open an SDE view to display (browse) the entry as output from the FSU - File Search/Update Window.
Supported for MVS SELCOPY licensees only.

FS Open the File Search window to search the contents of the entry.
Supported for MVS PDS/PDSE, CMS fileid, VSE LIBR sub-library and member entries only.

G Open the Library Member Generations List for the entry. Supported for MVS PDSE version 2 libraries defined with MAXGENS.

I Display detailed information for the entry. For file or DB2 object lists, opens a Data Set Information panel and, for Volume lists, opens the DASD Volume Statistics window.

IC Open the Execute IEBCOPY panel for this entry, using the entry name as the PDSIn field entry.

ID Open an IDCAMS Command window and issue an IDCAMS LISTCAT for the entry.

J Submit the library member entry to batch. Executes the CBLe CLI SUBMIT command using the entry name as input. (A CBLe frame window must be active for this operation to suceed.)
Supported in MVS and VSE environments only.

K Delete (Kill) the entry without prompting for verification.

L Open a Dataset List window for the entry.
Supported for Execute CBLVCAT windows only.
For **VSE LIBR Library member** list windows only, lock the LIBR member.

M Open a Library List window for the entry.
Supported for MVS PDS/PDSE, VSE LIBR library and sub-library entries only.

NX Include all excluded lines represented by a single shadow line. (Applies to **all** list types)

NXF Include the **first** line in a group of excluded lines represented by a single shadow line. (Applies to **all** list types)

NXL Include the **last** line in a group of excluded lines represented by a single shadow line. (Applies to **all** list types)

Q List dataset enqueues (major name SYSDSN) for the entry. Supported for MVS only.

R Rename the entry.

SD

Open the SDE BROWSE/EDIT Dialog Window to edit or browse the entry's data within a Structured Data Environment window view.
Supported for MVS SELCOPY licensees only.

T Issue a LISTVCAT operation against the entry with parameters TUNE and DEFINE.
For DASD Listwindows only, open the VTOC list window for the volume entry.

U Unallocate the MVS DD name or UNLOCK the VSE LIBR member entry.
Entries may only be unallocated or unlocked by the user that originally allocated or locked it.

UT Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.

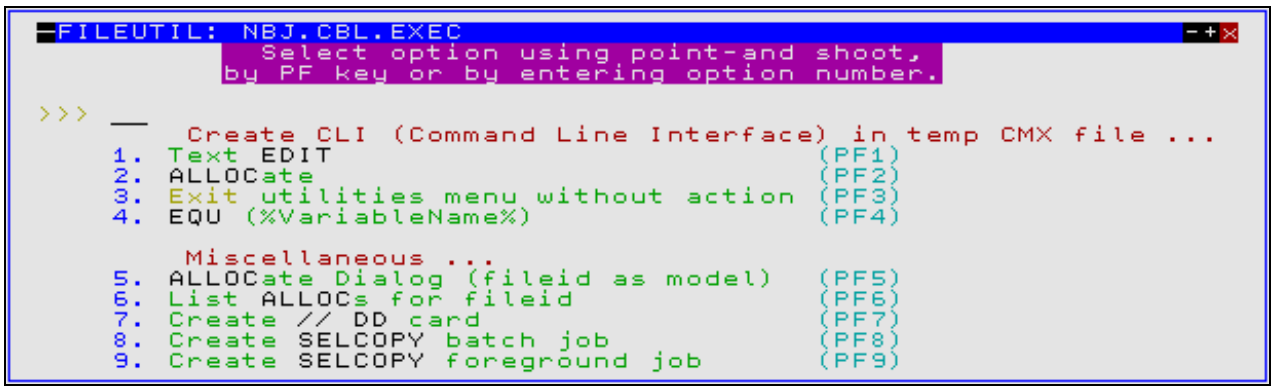


Figure 322. File Utilities Menu.

Options are selected by entering the required option number at the command prompt or executing the equivalent PFKey.

On selecting one othese options, a dialog panel or edit view containing generated syntax for the selected entry *entry_name* is opened as follows:

1. Text Edit	`<edit 'entry_name'`
2. ALLOCate	`<alloc f(MYDDNAME) reuse shr dsn('entry_name')`
4. EQU (%VariableName%)	`<equ MyFile 'entry_name'`
5. ALLOCate Dialog (fileid as model)	Opens the Allocate NonVSAM dialog.
6. List ALLOCs for fileid	`LA; where DsN=entry_name`
7. Create // DD card	`//MYDDNAME DD DISP=SHR,DSN=entry_name`

8. Create SELCOPY batch job	`//FILEUTIL EXEC PGM=SELCOPY`
	`//MYDDNAME DD DISP=SHR,DSN=entry_name`
	`//SYSPRINT DD SYSOUT=*`
	`//SYSIN DD *`
	`option worklen=65536 NoRdw`
	`read MYDDNAME`
	`print len=100 type=b stopaft=22`
	`/*`

9. Create SELCOPY foreground job	`** %USER%.FILEUTIL.Tnnnnnn.SLC *** L=001 --- yyyy/mm/dd HH:MM:SS (%USER%)`
	`*<RunSelc | Use this command to run SELCOPY in the foreground.`
	
	`option NoRdw * worklen=65536`
	`read MYDDNAME dsn='entry_name' dirdata`
	`print type=b stopaft=22`

VOpen the CBLe text editor to View (edit read/only) the entry.**VC**Open an Execute CBLVCAT window and issue a LISTVCAT and/or LISTVTOC operation (as appropriate) for the entry.**X**Exclude the line from the list display. (Applies to **all** list types)**Z**Perform a compress of an MVS PDS library to reclaim disk space occupied by replaced (back-level) members. This action performs an IEBCOPY to itself. No action is taken for PDSE entries, however, the IEBCOPY dialog is opened with an error message if executed against any non-PDS(E) entry.
Supported in MVS environments only./Open a drop down menu containing valid prefix command functions for the list window entry. Position the cursor on the required function and hit <Enter> to action the command.
Assigned to F16 by default.**?**Same as "/".>Open a new window containing a zoomed vertical display of the entry's fields. Particularly useful for list windows that have a large number of displayed columns.
Assigned to PF17 (Shift-F5) by default.

Command Cross-Reference

	Prefix Commands																															
	A	AS	AP	B	C	CF	CL	D	E	EU	EX	F	FO	FS	G	I	IC	ID	J	K	L	M	Q	R	SD	T	U	UT	V	VC	Z	?	/	>
VCAT	Y	Y	-	Y	Y	-	-	Y	Y	-	-	Y	Y	Y	-	-	-	Y	-	Y	Y	Y	Y	Y	Y	Y	-	-	Y	Y	Y	Y	-	Y
LVR	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	Y	Y	Y	-	Y	-	Y	-	Y	-	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	-	Y	Y
LVOL	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	-	Y	-	-	-	-	-	-	-	-	-	Y	-	-	-	Y	-	Y	Y	Y
LC	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	-	Y	-	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y
LD	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	-	Y	-	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y
LV	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	-	Y	-	Y	Y	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y
LVX	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	-	Y	-	Y	Y	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y
LA	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	-	Y	-	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	Y
Lab	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	Y	Y
LL	**2**	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	**1**	**1**	-	Y	Y	-	**1**	Y	Y	-	-	-	Y	Y
LQ	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	Y
LJQ	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	Y
LAS	Y	Y	-	Y	Y	Y	-	Y	Y	Y	-	Y	Y	-	-	Y	-	Y	-	Y	-	Y	Y	Y	Y	Y	-	Y	Y	Y	-	-	Y	Y
LP	-	-	-	Y	Y	Y	-	Y	Y	Y	-	Y	-	-	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	Y	-	-	-	Y	Y
FS	**2**	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	**1**	**1**	-	Y	Y	-	**1**	Y	Y	-	-	-	Y	Y

Legend:

1. VSE LIBR member list only.
2. MVS LIBR member list only.

Glossary

The following is a glossary of terms used in this document.

3270 Emulator
Third party software that emulates Mainframe 3270 hardware terminals on PC and UNIX based platforms.

CLI
A Command Line Interface is a text based method by which users can execute functions supported by the application.

CBLe
A powerful text editor that runs as an MDI application under SELCOPY/i. CBLe supports its own command line interface (CLI) and has been developed based on specifications for IBM's ISPF Edit, CMS XEDIT and Mansfield Software's KEDIT for Windows.

CBLVCAT
CBL licensable product that supports VSAM file tuning and VTOC, ICF/VSAM catalog and VSE LABEL reporting. Executes as a batch facility or interactively as a SELCOPY/i application.

Edit View or **Text Edit View**
A CBLe MDI document window that contains a display of text edited data. If the same file is displayed in multiple windows, then the user has multiple edit views of the file. Each edit view can have a different current line, ARBCHAR setting, ZONE columns, etc.

FDB
A Field Descriptor Block used to define the field column elements of a list.

List Column
A single column of text within the display area of the current list window. A list column may fall within a List Field Column or in the gap between field columns.

List Current Column
The first scrollable list column within the display area of the current list window. Key list columns (FDB field Key=Yes) are non-scrollable and so are not included

List Current Row
The first visible list row within the display area of the current list window.

List Field Column
A single column field within the current list window which has a maximum length as defined by the field's FDB entry.

List Focus Column
The list column on which the cursor is positioned within the list focus row.
If the cursor is positioned outside the list display area (e.g. the command line) or within the list prefix area, the list focus column is defined as being the list current column.

List Focus Row
The row within the current list window on which the cursor is positioned. If the cursor is positioned outside the list display area (e.g. the command line), then the list focus row is defined as being the list current row.

List Window
A class of SELCOPY/i window containing rows of associated information. List windows support point-and-shoot column sorting; select, sort and filter CLI commands; and prefix area commands.

MDI
Multiple Document Interface is a Microsoft specification for PC applications that enable the user to work with multiple documents at the same time. Each document is displayed in a separate child window within the client area of the application's main (frame) window. Typical MDI applications on PCs include word-processing and spread sheet applications.

MDI Client Area window
The MDI client area window is the display area within an MDI application's frame window. The MDI client area serves as the backround for MDI child windows.

MDI Child/Document Window
An MDI child or document window is opened in an application's client area window each time a document is opened. Each child window has a sizing border, title bar, window menu, minimise, maximise, restore and close buttons. A child window is clipped so that it is confined to the client window and cannot appear outside it.
When a child window is maximized, its client area completely fills the MDI client area window. In addition, the system automatically hides the child window's title bar, and adds the child window's window menu icon and Restore button to the MDI application's menu bar.

MDI Frame Window
An MDI frame window may be considered the main window of an MDI application. It is the parent window of the MDI client area window in which MDI child windows are opened. It has a sizing border, title bar, window menu, minimise, maximise restore and close buttons.

Modal Window
A modal window requires user interaction before further processing can occur. Window focus cannot be placed on any other window until the modal window is closed.

Edit Ring
The set of all **files** being edited within CBLe. It is not the set of all windows opened. e.g. The contents of one file may be displayed in more than one edit view (window.)

SDB
See SELCOPY Debug.

SELCOPY Debug (SDB)
An Intergrated Development Environment for SELCOPY that runs as an MDI application in SELCOPY/i.

SELCOPY/i
The interactive environment developed by CBL and supplied as part of the SELCOPY Product Suite. Requires licence key for SELCOPY and/or CBLVCAT elements of SELCOPY Product Suite.

SELCOPY/i INI
File containing configuration options for SELCOPY/i. The SELCOPY/i System INI file is processed on startup of SELCOPY/i and contains options that apply to all users. The SELCOPY/i User INI file contains options specific to each user that may, where appropriate, override options set in the SELCOPY/i System INI file.

SELCOPY/i VTAM
Name of the multi-user version of the SELCOPY/i application that executes under VTAM.

Storage Display Window
A SELCOPY/i window containing hexadecimal and character display of areas of storage.