



Reference and User Guide
SELCOPY/i Release 3.20

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

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

CBL Web Site - <http://www.cbl.com>

This document may be downloaded from <http://www.cbl.com/documentation.php>

Contents

Documentation Notes	1
Summary of Changes	2
First Edition (October 2013).....	2
About SELCOPY/i	5
Getting started with SELCOPY/i	6
Starting the SELCOPY/i program.....	6
Security Considerations.....	6
VSE Systems.....	6
MVS Systems.....	7
3270 Terminal Emulation.....	7
3270 Screen Sizes.....	7
Keyboard and Mouse Mapping.....	8
Window Concepts.....	8
Window Hierarchy.....	8
Manipulating Windows.....	8
Window Format.....	9
Pressing Buttons.....	10
Window Focus.....	10
Input fields.....	10
Changing the focus window.....	10
Window Names.....	10
Viewing Window Names.....	11
Window Class.....	11
System Menu.....	11
Function Keys.....	12
Default Function Keys (KEYLISTs active).....	12
Default Function Keys (KEYLISTs not active).....	12
SELCOPY/i Main Window.....	13
SELCOPY/i Main Window Menu Bar.....	14
SELCOPY/i Clipboard.....	14
SELCOPY/i Interactive Help.....	14
Window Classes	17
Storage Display Windows.....	17
Storage Window Display Format.....	17
Storage Window Resizing.....	17
Storage Display Navigation.....	18
Storage Data Manipulation.....	18
List Windows.....	18
List Window Status Bar.....	18
List Window Menu.....	18
View List Display.....	19
Field Descriptor Block (FDB).....	19
Edit View.....	20
Zoom View.....	20
Columns Displayed.....	21
Selecting, Sorting and Filtering.....	21
SELECT Clause.....	22
The Where Clause.....	23
The Sort (Order By) Clause.....	24
Sorting with the Cursor.....	25
List Entry Location.....	25
FIND Command.....	25
LOCATE Command.....	26
S Command.....	26
List Window Prefix Area.....	26
Interactive Panel Windows.....	27
Panel Window Format.....	28
Panel Window Size & Location.....	29
Panel Window Hierarchy.....	29
Panel Scrollable Display.....	29
Panel Window Views.....	30
Scrollable Input/Output Fields.....	30
Input Field Data Recall.....	31
Embedded Tables.....	31
Overview.....	31
Table Edit CLI (Primary) Commands.....	31
Table Edit Options.....	31
Table Edit Prefix (Line) Commands.....	32
Table Edit PFKeys.....	32
Table Editing Techniques.....	32
COLHEADER - SET/QUERY Option.....	34
Selection Lists.....	34
Select Dataset Name (Catalog).....	35

Contents

Window Classes

Select Dataset Name (VTOC).....	35
Select HFS Path Name.....	36
Select Member Name.....	36

Primary Option Menu (=).....	38
Menu Bar Items.....	38
Options.....	38
Panel Output Fields.....	38

Settings (=0).....	39
Options.....	39
Panel Input/Output Fields.....	39
Startup Settings (=0.1).....	39
Panel Input Fields.....	39
System Settings (=0.2).....	40
Panel Input Fields.....	40
Text Edit Settings (=0.3).....	40
Panel Input Fields.....	40
Structured Data Edit (SDE) Settings (=0.4).....	41
Options.....	41
Panel Input Fields.....	41
COBOL Compiler Options (=0.4.1).....	41
Panel Input Fields.....	42
PL/1 Compiler Options (=0.4.2).....	42
Panel Input Fields.....	42
List Window Settings (=0.5).....	42
Panel Input Fields.....	42
Batch Settings (=0.6).....	43
Panel Input Fields.....	43
DB2 Settings (=0.7).....	43
Options.....	43
Panel Input Fields.....	43
Execute SQL Settings (=0.7.2).....	44
Panel Input Fields.....	44
Program Function Keys (PF key) Settings (=0.8).....	44
Panel Input Fields.....	44
Options.....	44
Panel Input Fields.....	44
Options.....	45

Text Edit (=1).....	47
Panel Input Fields.....	47

Data Edit (=2).....	49
Structured Data Browse/Edit Panel.....	49
Menu Bar Items.....	49
Panel Input Fields.....	50
Structured Data Browse/Edit - Options Panel.....	52
Menu Bar Items.....	52
Panel Input Fields.....	52

List File Windows (=3).....	54
Options.....	54
List DASD Volumes (=3.1).....	54
Panel Input Fields.....	54
Prefix Line Commands.....	55
Columns Displayed.....	55
List VTOC Files (=3.2).....	56
Panel Input Fields.....	57
Prefix Line Commands.....	57
Columns Displayed.....	58
List VTOC Extents (=3.3).....	58
Panel Input Fields.....	59
Prefix Line Commands.....	59
Columns Displayed.....	60
Data Set List Utility (=3.4).....	61
Panel Input Fields.....	61
List Catalog Entries (=3.5).....	62
Panel Input Fields.....	63
Prefix Line Commands.....	64
Columns Displayed.....	65
List CMS Files.....	66
Panel Input Fields.....	66
Prefix Line Commands.....	67
Columns Displayed.....	67
List Dataset Details (=3.6).....	67

Contents

List File Windows (=3)	
Panel Input Fields.....	68
Prefix Line Commands.....	69
Columns Displayed.....	70
List Library Members (=3.7).....	71
Panel Input Fields.....	72
Prefix Line Commands.....	73
Columns Displayed.....	74
List MVS Allocated Files (=3.8).....	76
Panel Input Fields.....	76
Prefix Line Commands.....	76
Columns Displayed.....	77
List VSE Standard Labels.....	78
Panel Input Fields.....	78
Prefix Line Commands.....	78
Columns Displayed.....	79
List MVS Enqueues (=3.9).....	79
Panel Input Fields.....	80
Prefix Line Commands.....	80
Columns Displayed.....	80
List MVS Job Enqueues (=3.10).....	81
Panel Input Fields.....	81
Prefix Line Commands.....	81
Columns Displayed.....	81
List Associations (=3.11).....	82
Panel Input Fields.....	82
Prefix Line Commands.....	83
Columns Displayed.....	84
List HFS Path (=3.12).....	84
Panel Input Fields.....	84
Prefix Line Commands.....	85
Columns Displayed.....	85
File Copy (=5).....	88
Overview.....	88
Source and Target File Types.....	88
File Copy Panel.....	89
File Copy.....	89
Menu Bar Items.....	90
Panel Input Fields.....	90
File Reformat.....	94
Menu Bar Items.....	94
Panel Input Fields.....	94
PFKeys.....	95
File Copy - PDS Copy Statistics.....	95
Summary Format.....	95
Copy Statistics Fields.....	95
File Search/Update/Copy/Remap.....	97
Overview.....	97
Source File Types.....	97
Output Report.....	97
Unformatted File Search/Update/Copy.....	98
Formatted File Search/Update/Copy/Remap.....	99
File Search/Update/Copy/Remap Panels.....	100
Basic File Search Panel.....	101
FSU: Basic File Search Panel.....	101
Menu Bar Items.....	101
Panel Input Fields.....	102
PFKeys.....	106
Extended File Search/Update/Copy/Remap Panels.....	106
FSU: Extended File Search, Update, Copy & Remap Tasks.....	106
Menu Bar Items.....	106
Panel Options.....	107
FSU: Input File(s) Specification.....	109
Menu Bar Items.....	109
Panel Fields.....	109
FSU (unformatted): Record Selection.....	111
Menu Bar Items.....	111
Panel Fields.....	111
FSU (formatted): Record Selection.....	111
Menu Bar Items.....	112
Panel Fields.....	112
FSU (unformatted): Search records using WHERE expression.....	113
Menu Bar Items.....	113
Panel Fields.....	114
FSU (formatted): Search records using WHERE expression.....	114

Contents

File Search/Update/Copy/Remap

Menu Bar Items.....	114
Panel Fields.....	115
FSU (unformatted): Search Records using the FIND Command.....	115
Menu Bar Items.....	116
Panel Fields.....	116
FSU (formatted): Search Records using the FIND Command.....	116
Menu Bar Items.....	117
Panel Fields.....	117
FSU (unformatted): Change record data using the CHANGE command.....	118
Menu Bar Items.....	118
Panel Fields.....	119
FSU (formatted): Change record data using the CHANGE command.....	121
Menu Bar Items.....	121
Panel Fields.....	122
FSU: Copy Selected records to an Output File.....	124
Menu Bar Items.....	124
Panel Fields.....	124
FSU: Remap Fields in Selected records to an Output File.....	125
Menu Bar Items.....	126
Panel Fields.....	126
FSU: Copy Selected records to Output Library Members.....	127
Menu Bar Items.....	128
Panel Fields.....	128
FSU: Remap Fields in Selected records to Output Library Members.....	129
Menu Bar Items.....	129
Panel Fields.....	129
FSU: Report File.....	131
Menu Bar Items.....	131
Panel Fields.....	131
File Selection Panels.....	132
Multiple Fileid Masks.....	132
Table View.....	133
Single Row (Zoomed) View.....	133
Input Values.....	133
Select Files to Process.....	135
Table View.....	135
Single Row (Zoomed) View.....	136
Input Values.....	136
Select Input Members.....	137
Table View.....	137
Single Row (Zoomed) View.....	138
Input Values.....	138
Select Input Members - FIND.....	139
Input Values.....	139
Generate Formatted Record Expression Panels.....	140
Major OR Sub-Expressions.....	140
Table View.....	141
Single Row (Zoomed) View.....	141
Input Values.....	141
AND Sub-Expressions.....	142
Table View.....	142
Single Row (Zoomed) View.....	143
Input Values.....	143
Minor OR Sub-Expressions.....	144
Table View.....	144
Single Row (Zoomed) View.....	144
Input Values.....	145
SDE: Select Record-Type.....	146
SDE: Select Field Name.....	147
Multiple FIND & CHANGE Specification Panels.....	147
Multiple Find Commands (unformatted).....	147
Table View.....	147
Single Row (Zoomed) View.....	148
Input Values.....	148
Multiple Find Commands (formatted).....	149
Table View.....	149
Single Row (Zoomed) View.....	150
Input Values.....	150
Multiple Change Commands (unformatted).....	151
Table View.....	151
Single Row (Zoomed) View.....	152
Input Values.....	152
Multiple Change Commands (formatted).....	154
Table View.....	154
Single Row (Zoomed) View.....	155
Input Values.....	155

Contents

File Search/Update/Copy/Remap

File Search/Update/Copy/Remap Output.....	157
Report Format.....	157
Record Type: Command.....	158
Record Type: Summary.....	158
Record Type: Hit.....	159
Record Type: IOError.....	161
Record Type: Alias.....	161
Record Type: Record.....	161
Function Keys.....	162
File Update Undo.....	162
Overview.....	162
File Update Undo Panel.....	163
File Update Undo Output.....	163
Report Format.....	163
Report Fields.....	163
Report Data.....	164
Summary Block.....	164
Sample Terse Report Output.....	165
Sample EXTENDED Output.....	166

Compare Files/Libraries Menu (=7).....168

Options.....	168
Compare Files (=7.1).....	168
Overview.....	168
Source File Types.....	168
Output Report.....	168
Unformatted Compare.....	169
Formatted Compare.....	169
Hierarchical Compare.....	170
Record Synchronisation.....	170
Terminology.....	170
Overview.....	171
1-TO-1 Synchronisation.....	171
Read-Ahead Synchronisation.....	171
Key Synchronisation.....	172
Hierarchical Key Synchronisation.....	173
Compare Files Panels.....	175
Basic Unformatted Compare Panel.....	175
Compare Files: Basic Options.....	175
Compare Type - Formatted/Unformatted ?.....	178
Extended Unformatted Compare Panels.....	178
Compare Files (unformatted): New file details and options.....	178
Compare Files (unformatted): Old file details and options.....	180
Compare Files (unformatted): Re-synchronisation options.....	181
Compare Files (unformatted): Specify Key fields.....	184
Compare Files: Specify Key Pos/Len.....	184
Compare Files (unformatted): Output Files.....	185
Compare Files (unformatted): Options / Action.....	186
Formatted Compare Panels.....	187
Compare Files (formatted): New file details and options.....	187
Compare Files (formatted): Old file details and options.....	188
Compare Files (formatted): Re-synchronisation options.....	190
Compare Files (formatted): Specify Key fields.....	192
Compare Files: Specify Key Pos/Len.....	193
SELCOPY/i Compare Files - KEY Columns (Record Types List).....	193
SELCOPY/i Compare Files - Select from Field Names List.....	194
Compare Files (formatted): Output Files.....	195
Compare Files (formatted): Options / Action.....	196
SELCOPY/i Compare Files - Select Record Types to Compare.....	197
SELCOPY/i Compare Files - Select Field Names to Compare.....	198
SELCOPY/i Compare Files - Select Field Names List.....	198
Compare Files Output.....	199
Report Format.....	199
Record Type: Command.....	202
Record Type: Files.....	202
Record Type: Compare Compare-record_type.....	202
Record Type: Field.....	204
Record Type: Summary.....	204
Record Type: Key.....	205
Function Keys.....	205
Compare Libraries (=7.2).....	205
Overview.....	205
Compare Libraries Panel.....	205
Menu Bar Items.....	206
Panel Input Fields.....	206
Compare Libraries Output.....	207

Contents

SELCOPY Debug & Development (=8.1)	209
Invoking SELCOPY Debug.....	209
Load Library Search Chain.....	212
SELCOPY Loop Break-in.....	212
SELCOPY Debug Windows.....	213
SELCOPY Debug Main window.....	213
SYSIN Window.....	214
SYSPRINT Window.....	215
SQL Log Window.....	216
WTO Log Window.....	216
Work Area/Current Input Record Window.....	217
POS Expression Window.....	217
@ Pointer Window.....	218
Columns Displayed.....	218
Equates Window.....	218
Columns Displayed.....	219
PCB Window.....	219
TRACE Window.....	219
Point-and-Shoot Popup Menu.....	220
SELCOPY Debug Commands.....	221
BREAKPOINT.....	222
EOJ.....	222
GO.....	222
RERUN.....	223
STEPINTO.....	223
STEPOVER.....	223
TRACK.....	224
WINDOW.....	225
SELCOPY Debug Function Keys.....	225
Utilities Menu (=8)	227
Options.....	227
CBLVCAT Interactive (VCI) (=8.2).....	227
CBLVCAT Interactive Window.....	227
Prefix Commands.....	229
Columns Displayed.....	230
Raw Data Window.....	230
Prefix Line Commands.....	232
Columns Displayed.....	233
Execute IDCAMS (=8.3).....	234
Panel Fields.....	235
Prefix Commands.....	235
Columns Displayed.....	235
Execute POWER.....	236
Panel Fields.....	236
Prefix Commands.....	236
Columns Displayed.....	237
Define Catalog ALIAS (=8.4).....	238
Menu Bar Items.....	238
Panel Fields.....	238
Create Library ALIAS (=8.5).....	239
Panel Fields.....	239
Execute IEBCOPY (=8.6).....	240
Panel Fields.....	240
DB2 Dynamic SQL.....	241
Panel Fields.....	244
Favourite Datasets/Commands (=8.7).....	245
Overview.....	245
Favourite Datasets/Commands Panel.....	245
Panel Input Fields.....	245
System Information Menu (=8.8).....	247
Options.....	247
Operating System Window (=8.8.1).....	247
Menu Bar Items.....	248
LPA Modules Window (=8.8.2).....	248
Columns Displayed.....	249
Link List Window (=8.8.3).....	249
Columns Displayed.....	249
APF List Window (=8.8.4).....	250
Columns Displayed.....	250
Task List Window (=8.8.5).....	250
Columns Displayed.....	251
Allocated Storage Windows.....	251
Columns Displayed.....	251
Loaded Programs Window (=8.8.6).....	251
Columns Displayed.....	252
SELCOPY/i Storage Statistics Window.....	252

Contents

Utilities Menu (=8)	
Heap.....	252
Stack.....	253
Lists.....	253
SELCOPY/i Module List Window.....	253
CBLVCAT SVC window (=8.8.7).....	254
CBLNAME Window.....	254
File Search (=8.9).....	254
Panel Fields.....	255
Prefix Line Commands.....	256
Columns Displayed.....	257
Search for Library Members (=8.10).....	257
Overview.....	257
Search for Library Members Panel.....	257
Menu Bar Items.....	258
Panel Input Fields.....	258
Search for Library Members Output.....	259
Calendar Window (=8.13).....	260
Calculator Window (=8.14).....	261
Create New Datasets Menu (=8.15).....	261
Options.....	261
Allocate NonVSAM (=8.15.1).....	262
Menu Bar Items.....	262
Define VSAM KSDS/ESDS/RRDS/LDS (=8.15.2/3/4/5).....	263
Menu Bar Items.....	263
Define GDG Base (=8.15.7).....	264
Menu Bar Items.....	264
Panel Input Fields.....	264
Data Set Information.....	265
Dataset Information - Non-VSAM.....	266
Page 1.....	266
Page 2.....	268
Page 3-n.....	268
Dataset Information - VSAM.....	270
Page 1.....	270
Data Component Information.....	271
Dataset Date Information.....	271
DEFINE Options.....	271
Page 2.....	272
Page 3.....	274
Page 4.....	274
Page 5-n.....	274
Create Structure (SDO) Menu (=9)	276
Menu Bar Items.....	276
Options.....	276
Create Structure from Copybook(s).....	276
Create Structure from COBOL/PL1 copybook(s).....	276
Options.....	277
Panel Input Fields.....	277
Create Structure - Copybook Library List.....	277
Create Structure - Define Record-Types.....	278
Panel Input Fields.....	279
Create Structure from XREF File.....	280
Panel Input Fields.....	280
Display Record Layout.....	281
Display Record Layout Panel.....	281
Panel Input Fields.....	282
Display Record Layout Output.....	282
Columns Displayed.....	283
Create File Filter (=10)	284
Panel Input Fields.....	284
Unformatted Selection Criteria.....	286
Panel Input Fields.....	287
Formatted Selection Criteria.....	287
FILTER (formatted) - INCLUDE/EXCLUDE record-types.....	287
Filter (formatted) - Selection Criteria.....	288
Panel Input Fields.....	289
Print Data File (=11)	290
Print Data File - Input.....	290
Print Data File - VFMT Options.....	292
Print Data File - FMT Options.....	294
Print Data File - Select Record-Types.....	296
Print Data File - Select Field Names.....	297
Print Data File - Field Names List.....	297

Contents

Print Data File (=11)	
Print Data File - CHAR Options.....	298
Print Data File - UNFMT Options.....	299
Print Data File - Destination.....	300
DB2 Utilities.....	302
DB2 Primary Option Menu.....	302
Menu Bar Items.....	303
Options.....	303
Panel Input/Output Fields.....	303
Execute DB2 Commands.....	304
Menu Bar Items.....	305
Field Entries.....	305
Execute SQL Statements.....	305
Menu Bar Items.....	306
Field Entries.....	307
Edit Tables and Views.....	307
Menu Bar Items.....	308
Field Entries.....	308
Browse Tables and Views.....	310
Menu Bar Items.....	311
Field Entries.....	311
Create DB2 Objects.....	312
Menu Bar Items.....	312
Options.....	313
Create Storage Group.....	313
Menu Bar Items.....	313
Field Entries.....	313
Create Work File Database.....	314
Menu Bar Items.....	314
Field Entries.....	314
Create User Database.....	315
Menu Bar Items.....	315
Field Entries.....	316
Create Work File Table Space.....	316
Work File Table Space Values.....	317
Menu Bar Items.....	317
Work File Table Space Values - Panel Fields.....	317
Create User Table Space.....	318
Table Space Name & Type.....	319
Menu Bar Items.....	319
Table Space Name & Type - Panel Fields.....	319
Table Space Type Attributes.....	320
Table Space Type Attributes - Panel Fields.....	321
Table Space Partition Attributes.....	323
Table Space Partition Attributes - Panel Fields.....	323
Table Space Options (1/2).....	324
Table Options (1/2) - Panel Fields.....	324
Table Space Options (2/2).....	325
Table Options (2/2) - Panel Fields.....	326
Create Table.....	326
Table Name & Location.....	326
Menu Bar Items.....	327
Table Name & Location - Panel Fields.....	327
Table Definition.....	328
Table Definition - Panel Fields.....	329
Columns & Constraints.....	329
Columns & Constraints - Panel Fields.....	330
Column Definitions.....	331
Column Definitions - Panel Fields.....	331
Column Definition - FIELDPROC Parameters.....	338
Primary Key Definition.....	339
Primary Key Definition - Panel Fields.....	339
Unique Constraint Definitions.....	340
Unique Constraint Definitions - Panel Fields.....	340
Unique Constraint Key Columns.....	341
Unique Constraint Key Columns - Panel Fields.....	341
Referential Constraint Definitions.....	342
Referential Constraint Definitions - Panel Fields.....	343
Referential Constraint - Parent Key Columns.....	344
Referential Constraint - Parent Key Columns - Panel Fields.....	344
Referential Constraint - Foreign Key Columns.....	345
Referential Constraint - Foreign Key Columns - Panel Fields.....	346
Check Constraint Definitions.....	346
Check Constraint Definitions - Panel Fields.....	347
Model using LIKE.....	347
Model using LIKE - Panel Fields.....	348

Contents

DB2 Utilities

Model on Result Table.....	349
Model on Result Table - Panel Fields.....	349
Result Table Column Name Remap.....	350
Model on Tables/Views (Load Values).....	350
Model on Tables/Views - Panel Fields.....	351
Materialized Query Definition.....	353
Materialized Query Definition - Panel Fields.....	353
Implicit TableSpace Options.....	354
Implicit TableSpace Options - Panel Fields.....	354
Table Options (1/2).....	355
Table Options (1/2) - Panel Fields.....	355
Range Partitioning Key Columns.....	357
Range Partitioning Key Columns - Panel Fields.....	357
Tablespace Range Partitions.....	358
Tablespace Range Partitions - Panel Fields.....	358
Range Partitioning Key Values.....	359
Range Partitioning Key Values - Panel Fields.....	359
Table Options (2/2).....	360
Table Options (2/2) - Panel Fields.....	360
Generate SQL.....	361
Generate SQL - Panel Fields.....	361
Create View.....	362
Menu Bar Items.....	363
Field Entries.....	363
Create Alias.....	364
Menu Bar Items.....	364
Field Entries.....	364
Create Synonym.....	365
Menu Bar Items.....	365
Field Entries.....	365
Create Distinct Type.....	366
Menu Bar Items.....	366
Field Entries.....	366
Create Trigger.....	367
Menu Bar Items.....	367
Field Entries.....	368
Create Sequence.....	370
Menu Bar Items.....	370
Field Entries.....	370
Create Role.....	371
Menu Bar Items.....	372
Field Entries.....	372
Drop DB2 Objects.....	372
Menu Bar Items.....	372
Options.....	373
Drop Storage Group.....	373
Menu Bar Items.....	373
Field Entries.....	373
Drop Database.....	374
Menu Bar Items.....	374
Field Entries.....	374
Drop Table Space.....	374
Menu Bar Items.....	375
Field Entries.....	375
Drop Table.....	375
Menu Bar Items.....	376
Field Entries.....	376
Drop View.....	376
Menu Bar Items.....	377
Field Entries.....	377
Drop Alias.....	377
Menu Bar Items.....	378
Field Entries.....	378
Drop Index.....	378
Menu Bar Items.....	379
Field Entries.....	379
Drop Synonym.....	379
Menu Bar Items.....	380
Field Entries.....	380
Drop Type.....	380
Menu Bar Items.....	380
Field Entries.....	381
Drop Function.....	381
Menu Bar Items.....	382
Field Entries.....	382
Drop Stored Procedure.....	382

Contents

DB2 Utilities

Menu Bar Items.....	383
Field Entries.....	383
Drop Trigger.....	383
Menu Bar Items.....	384
Field Entries.....	384
Drop Global Temporary Table.....	384
Drop Sequence.....	385
Menu Bar Items.....	385
Field Entries.....	385
Drop Role.....	386
Menu Bar Items.....	386
Field Entries.....	386
Drop Trusted context.....	386
Menu Bar Items.....	387
Field Entries.....	387
List DB2 Objects.....	387
Menu Bar Items.....	388
Options.....	388
List Storage groups.....	388
Panel Input Fields.....	388
Prefix Line Commands.....	388
List Databases.....	389
Panel Input Fields.....	389
Prefix Line Commands.....	389
List Table spaces.....	389
Panel Input Fields.....	390
Prefix Line Commands.....	390
List Tables.....	390
Panel Input Fields.....	390
Prefix Line Commands.....	391
List Views.....	391
Panel Input Fields.....	391
Prefix Line Commands.....	392
List Aliases.....	392
Panel Input Fields.....	392
Prefix Line Commands.....	392
List Indexes.....	393
Panel Input Fields.....	393
Prefix Line Commands.....	393
List Synonyms.....	393
Panel Input Fields.....	394
Prefix Line Commands.....	394
List Distinct Types.....	394
Panel Input Fields.....	394
Prefix Line Commands.....	395
List Triggers.....	395
Panel Input Fields.....	395
Prefix Line Commands.....	395
List Global Temporary Tables.....	395
Panel Input Fields.....	395
Prefix Line Commands.....	396
List Sequences.....	396
Panel Input Fields.....	396
Prefix Line Commands.....	396
List Roles.....	396
Panel Input Fields.....	397
Prefix Line Commands.....	397
List Trusted Contexts.....	397
Panel Input Fields.....	397
Prefix Line Commands.....	397
List Columns.....	397
Panel Input Fields.....	398
Prefix Line Commands.....	398
List Volumes.....	398
Panel Input Fields.....	398
Prefix Line Commands.....	398
List Table space parts.....	399
Panel Input Fields.....	399
Prefix Line Commands.....	399
Select DB2 Objects.....	399
Select Storage Group.....	400
Panel Input Fields.....	400
Prefix Line Commands.....	400
Select Database.....	400
Panel Input Fields.....	401
Prefix Line Commands.....	401

Contents

DB2 Utilities

Select Tablespace.....	401
Panel Input Fields.....	402
Prefix Line Commands.....	402
Select Table.....	402
Panel Input Fields.....	403
Prefix Line Commands.....	404
Select Distinct Type.....	404
Panel Input Fields.....	405
Prefix Line Commands.....	405
Select Unique Key Constraint.....	405
Panel Input Fields.....	406
Prefix Line Commands.....	407
Select Foreign Key Column.....	407
Panel Input Fields.....	407
Prefix Line Commands.....	408
Audit Trail Functions.....	408
Menu Bar Items.....	408
Options.....	408
Audit Log Dataset Options.....	408
Panel Input Fields.....	408
Print Audit Report.....	409
Menu Bar Items.....	410
Panel Input Fields.....	410
List Audit Datasets.....	410

Window List (=W).....	411
-----------------------	-----

SELCOPY/i Command Reference.....412

ALIAS.....	414
AMS.....	414
AMSDIALOG.....	415
APE.....	415
AUDPRINT.....	416
BACKWARD.....	416
BOTTOM.....	417
BROWSE.....	417
CALENDAR.....	417
CALC.....	418
CBLI.....	418
CBLICANCEL.....	418
CBLNAME.....	419
CFOUT.....	419
CLOSE.....	419
COMMANDLINE.....	420
COMPFILE.....	421
COMPLIB.....	429
CURSORSELECT.....	430
DB2.....	430
DCMD.....	431
DOWN.....	432
DSINFORMATION.....	433
DSQL.....	433
DRAGBORDERMINUS.....	434
DRAGBORDERPLUS.....	434
EDIT.....	435
EO.....	436
ERASE.....	437
EXECSQL.....	438
FAV.....	440
FCOPY.....	441
FORWARD.....	444
FS.....	444
FSU.....	445
FSUEND.....	456
FSUOUT.....	456
FSUUNDO.....	457
HELP.....	458
HOME.....	459
IEBCOPYDIALOG.....	459
ISPF.....	459
ISPFUTIL.....	460
KEYS.....	460
KEYS Dialog.....	461
KEYS Dialog (KEYLISTs ON).....	461
KEYS Dialog (KEYLISTs OFF).....	461
KEYLIST.....	462

Contents

SELCOPY/i Command Reference	
KEYLIST Dialog.....	462
Prefix Line Commands.....	463
LA.....	463
LAS.....	463
LC.....	465
LD.....	467
LEFT.....	469
LJQ.....	470
LL.....	470
LLX.....	471
LP.....	473
LQ.....	474
LV.....	475
LVOL.....	476
LVR.....	476
LX.....	477
MAXIMISE.....	477
MDINEXT.....	478
MDIPREV.....	478
MINIMISE.....	478
MOVEWINDOW.....	479
NEXTMAINWINDOW.....	480
NEXTWINDOW.....	480
POWER.....	480
PREVMAINWINDOW.....	481
PREVWINDOW.....	481
QUICKREF.....	482
QUIT.....	482
RENAME.....	482
RESTORE.....	483
RETRIEVE.....	483
RIGHT.....	484
SDATA.....	485
SDE.....	485
SDSF.....	485
SELCOPY.....	486
SETCOLOUR.....	487
SETFOCUS.....	488
SHOWPOPUPMENU.....	489
SHOWWATTR.....	490
SIZEWINDOW.....	490
SQL.....	492
SVC.....	493
SYSAPP.....	493
SYSCOMMAND.....	493
SYSI.....	494
SYSLL.....	494
SYSLPA.....	494
SYSTEMENU.....	495
SYSPGM.....	495
SYSSTOR.....	495
SYSTASK.....	495
TASK.....	496
TOP.....	496
UP.....	497
VCAT.....	498
VIEW.....	498
VOLSTATS.....	499
WINDOWLIST.....	500
WINDOWNAMES.....	500
XMLGEN.....	501
Unix System Services (USS) Commands.....	509
USS CHDIR.....	509
USS GETCWD.....	509
USS LINK.....	510
USS MKDIR.....	510
USS REALPATH.....	510
USS RENAME.....	511
USS RMDIR.....	511
USS STAT.....	511
USS UNLINK.....	512
SELCOPY/i VTAM commands.....	513
MESSAGE.....	513
QUERY.....	513

Contents

SELCOPY/i VTAM commands	
STOP.....	514
SELCOPY/i Dump Files.....	515
Appendix A - SELCOPY/i Window Classes.....	516
Appendix B - List File Prefix Command Summary.....	517
Command Cross-Reference.....	519
Glossary.....	521

Documentation Notes

First Edition, October 2013

Information in this document details general features and functionality of the **SELCOPY Product Suite 3.20** component, **SELCOPY/i**.

This document replaces the previous edition of *SELCOPY/i Reference and User Guide* applicable to **SELCOPY Product Suite 3.10**, which is now obsolete.

Copyright in the whole and every part of this document and of the SELCOPY Products 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, SELCOPY/i and CBLVCAT, is available for download and install from <http://www.cbl.com/selcdl.html>.

The following publications for SELCOPY Product Suite and its component products are available in Adobe Acrobat PDF format at CBL web page <http://www.cbl.com/selcdoc.html>:

- SELCOPY Product Suite Customisation Guide
- SELCOPY User Manual
- CBLVCAT User Manual
- SELCOPY/i Reference and User Guide
- SELCOPY/i Text Editor (CBL*e*) Manual
- SELCOPY/i Structured Data Editor 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:

MVS - z/OS, OS/390, MVS/ESA, MVS/XA, MVS/SP, OS.

VSE - z/VSE, VSE/ESA, VSE/SP, DOS.

CMS - z/VM, VM/ESA, VM/XA, VM/SP.

All - All MVS, VSE and CMS operating systems.

Summary of Changes

This section is a summary of significant new features provided in SELCOPY/i Release 3.20.

First Edition (October 2013)

SELCOPY/i 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 "*SELCOPY/i Training Manual*" publication.

For details, see:

◇ "*Primary Option Menu (=)*"

ISPF style KeyLists

Introduce definition and management of function keys within SELCOPY/i 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 SELCOPY/i window's window class.

For details, see:

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

ISPF SPLIT

SELCOPY/i 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 SELCOPY/i 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 organisation 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 catalog 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 amendments 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)*"

About SELCOPY/i

System Environment

SELCOPY/i 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 SELCOPY/i 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.

SELCOPY/i Environment

SELCOPY/i 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, SELCOPY/i 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

SELCOPY/i has the same functionality when executing as a VTAM application as it does when running in user's TSO/E address space. Therefore, customising SELCOPY/i 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 SELCOPY/i

This chapter introduces end users to some basic SELCOPY/i concepts.

Starting the SELCOPY/i program

How SELCOPY/i 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, SELCOPY/i 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 SELCOPY/i must have first been defined as an ISPF application. See <i>SELCOPY Products Suite Customisation Guide</i> for further information on configuring access to SELCOPY/i in ISPF.
	VTAM	Enter LOGON APPLID(SELCOPYI) on a VTAM USS screen. Note: The SELCOPY/i VTAM session controller program (CBLIVTAM) must be running and the SELCOPY/i VTAM applid, SELCOPYI, must be active. See <i>SELCOPY Products Suite Customisation Guide</i> for further information on configuring access to SELCOPY/i as a stand-alone VTAM application.
VSE	VTAM	Enter LOGON APPLID(CBLIVTAM) on a VTAM USS screen. Alternatively, SELCOPY/i 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 SELCOPY/i VTAM session controller program (CBLIVTAM) must be running in a static or dynamic partition and the SELCOPY/i VTAM applid, CBLIVTAM, must be active. See <i>CBL Software Install Guide for VSE Systems</i> for further information.
VM	CMS	Enter SELCOPYI on a CMS command line to execute the SELCOPY/i 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 SELCOPY/i 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, SELCOPY/i 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, SELCOPY/i 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 SELCOPY/i 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, SELCOPY/i may be customised to restrict its use to only a specified group of trusted users. Following logon, these users will be prompted for a SELCOPY/i 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 SELCOPY/i. 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 SELCOPY/i using their RACF, or equivalent security package, login id. Under TSO, no SELCOPY/i login is performed as the user's TSO login id is used instead.

Thereafter, the user's access privilege 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 SELCOPY/i features using RACF profiles.

Resource Name	SELCOPY/i Feature
System	Access to the z/OS Operating System information available by selecting 'Operating System' from the Utilities/System menu in the CBL main window menu bar, or via line commands SYSI , SYSLPA , SYSLL , SYSAPF , SYSTASK , SYSSTOR , SYSPGM and CBLNAME .
UserTSO	Log on to SELCOPY/i under TSO and ISPF.
UserVTAM	Log on to SELCOPY/i 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 SELCOPY/i 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 ability 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 *SELCOPY/i 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 and are available for download from the *SELCOPY/i Downloads* web page.

Keyboard and Mouse Mapping

Configuration of the 3270 emulator software keyboard and mouse is recommended to greatly enhance the user's experience of the SELCOPY/i 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 SELCOPY/i, 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 SELCOPY/i 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 [SELCOPY/i and 3270 Terminal Emulation Software](#) and [SELCOPY/i Downloads](#).

Window Concepts

The SELCOPY/i 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 SELCOPY/i windows. It also provides instructions on how to work with windows in the SELCOPY/i 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 SELCOPY/i 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.

SELCOPY/i MDI applications include the CBL 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, for CBL text edit document windows only, **SET WINPOS**.

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 CBLed 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 CBLed 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 bottom of the SELCOPY/i main window. The minimise button will change from a minus sign to a solid vertical bar (representing restore).

Also see line command **MINIMISE**, for CBLed 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 CBLed 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 CBLed text edit document windows only, **WINDOW CLOSE**.

Window Format

SELCOPY/i 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.

The Title Bar

The title bar contains the title of the window.

The 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.

The 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 SELCOPY/i main window.

The 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 SELCOPY/i main window.

The 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 SELCOPY/i main window.

The Close Button

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

The 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 SELCOPY/i application. When enabled they are displayed in white, when disabled they are displayed in blue in which case their selection will have no effect.

The 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.

The 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 in the following ways:

- 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 **CBL** 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 SELCOPY/i 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 SELCOPY/i 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 common 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 SELCOPY/i Windows Classes may be found in [Appendix A - SELCOPY/i 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 SELCOPY/i Rel 3.20 onward, on z/OS systems SELCOPY/i maintains its function key definitions in **KEYLISTs**, analogous to ISPF keylists.

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

Default Function Keys (KEYLISTs active)

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

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.
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 columns.
F23	MoveWindow by x=+5	Drag window right 5 columns.

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

F7	DragBorderMinus	Drag top/bottom border up 1 line.
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.

Default Function Keys (KEYLISTs not active)

For non-z/OS systems (or with KEYLIST OFF in effect), SELCOPY/i 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 SELCOPY/i 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 SELCOPY/i 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 SELCOPY/i window list.
F24	Retrieve +	Retrieve the next command issued and place it at the command line.

The SELCOPY/i **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 SELCOPY/i **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	

SELCOPY/i Main Window

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

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



Figure 1. SELCOPY/i 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 SELCOPY/i 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 SELCOPY/i main window menu bar items.
- The **client area**. Occupying the body of the window, it contains:
 - ◆ The SELCOPY/i logo, release, copyright notice and CBL contact details.
 - ◆ The operating system name and version.
 - ◆ The user id.
 - ◆ The build level of SELCOPY/i. 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 SELCOPY/i facilities that duplicate or extend the menu facilities.

You can launch SELCOPY/i facilities by doing the following:

- Select a SELCOPY/i 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**.

SELCOPY/i Main Window Menu Bar

The SELCOPY/i main menu bar is located at the top of the SELCOPY/i main window.

The main menu consists of the following item:

Home

Open the **CBLe Text Editor** application and edit the user's HOME command centre file. This is equivalent to executing the line command **HOME**.

All SELCOPY/i facilities may be started from the CBLe Text Editor application and run as a child window of the CBLe parent window. The exception to this is the SELCOPY Debug application which, although started from CBLe, operates as a separate, special instance of the CBLe text editor. In some cases, SELCOPY/i facilities require a CBLe text edit environment to operate successfully. e.g. Structured Edit (SDE), DB2, File Search, Update & Copy.

Because of this, menu items for all available SELCOPY/i facilities are located in the **CBLe main window menu** bar and not the SELCOPY/i main window menu bar.

SELCOPY/i Clipboard

SELCOPY/i supports a clipboard facility to allow users to Copy, Cut and Paste data between windows running in the SELCOPY/i 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 SELCOPY/i clipboard is not associated with any other clipboard facility offered by the system.

SELCOPY/i Interactive Help

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

SELCOPY/i 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 SELCOPY/i window. e.g. Executing HELP in a Library List window opens the help window for the topic List Library Members.

```

-CBLi Interactive Help.
Back Forward Home Close Source Text Help
Command>
Location>

CBL Interactive

-----

CBLi Reference and User Guide
CBL Editor
Structured Data Environment

-----

SELCOPY User Manual
CBLVCAT User Manual

SELCOPY 2.01 New Features
SELCOPY 2.02 New Features
CBLVCAT 2.12 New Features
CBLi 1.60 New Features

-----

CBL Products Install Guide (For MVS)
CBL Products Install Guide (For CMS and VM/VSE)
CBL Products Install Guide (For VSE)

CBLi Update Guide (For MVS)

-----

CBLi INI control file

Line 1 of 28 Col 1 of 77 File: CBL.DIST.CBLI.HELP.HTML(CBLINDEX)

```

Figure 2. SELCOPY/i Main Help Menu window.

```

-SELCOPY Interactive Main Window
Back Forward Home Close Source Text Help
Command>
Location>

previous next contents

-----

SELCOPY Interactive Main window

Like the CBL text editor, SELCOPY Interactive 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 Interactive Main (frame) Window supports all MDI child windows
supported by the CBL frame window (including SDE Edit). The SELCOPY
Interactive frame window is actually a CBL frame window with additional
features and characteristics specifically relating to SELCOPY execution.
These features are discussed in this section whereas details on CBL frame
window features may be found in the CBL Text Edit documentation.

The SELCOPY Interactive Main window must always contain the Control Cards,
Output Listing and TRACE Windows. Closing any of these windows will quit the
SELCOPY main window and so end the Interactive session.

When a session is started, these 3 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

Line 1 of 106 Col 1 of 84 File: CBL.DIST.CBLI.HELP.html(winXSUMA)

```

Figure 3. 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 SELCOPY/i suite of help files.

Window Classes

Every SELCOPY/i 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 existence 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 SELCOPY/i window classes may be found in *Appendix A - SELCOPY/i Window Classes*.

Characteristics of CBLLe Text Edit window classes, including Structured Data Edit document views, are detailed in the *SELCOPY/i Text Editor* documentation.

Many window classes are one-to-one with an individual SELCOPY/i 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 SELCOPY/i 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.
- CBLLe 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 CBLLe 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 SELCOPY/i 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 or WINWIPO0.

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).

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

--Name--	--Type--	Key	Offset	Length	-----Title-----	DispLen	Digits	Sc
DataClas	Char	No	100	8	SMS Data Class	8	0	
DevC	Char	No	52	4	Device Class	4	0	
DSType	Char	No	66	14	Dataset type	14	0	
Entry	Char	Yes	0	44	Entry name	36	0	
EType	Char	No	59	7	Catalog entry type	7	0	
FSeq	UInt	No	56	2	File Sequence number	3	0	
MgmtClas	Char	No	116	8	SMS Management Class	8	0	
StorClas	Char	No	108	8	SMS Storage Class	8	0	
T	Char	No	58	1	Catalog entry type code	1	0	
UnitName	Char	No	84	16	Unit name	16	0	
UnitType	Hex	No	80	4	Unit type	8	0	
Vol	Char	Yes	44	6	Volume serial number	6	0	
VSeq	UInt	Yes	50	2	Volume Sequence number	3	0	

Line 1 of 13 | Col 1 of 80 | Views 1 | select * sort Name

Figure 4. 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.)

```

Field Descriptor Element
View Back Forward FDB Edit Refresh Help
Command>

-Name-- --Type-- Key Offset Length -----Title----- Displen Di
Digits  Int    No     8     1  Decimal digits (precision)      2
Displen Int    No    24     4  Field display length             5
Key     BitFlag No     5     1  Field is a key                   3
Length  Int    No    20     4  Field length                      5
Name    VChar  Yes    0     18 Field name                        8
Offset  Int    No    16     4  Field offset                      5
Scale   Int    No     9     1  Decimal scale (fraction digits)  2
Title   VChar  No    12    30  Field title                       23
Type    Enum   No     4     1  Field type                         8
Line 1 of 9 | Col 1 of 87 | Views 1 | select * sort Name

```

Figure 5. 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** on the list window command line.

A CBL text editor window is opened, the list is loaded into CBL 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 CBL editor window is already open, then the list will be edited in a new document window of the existing CBL 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.)

```

SELCPY/i - Dataset List: NBJ.CBLI 2011/12/12 1
View Refresh Back Forward FDB Text Help WS wR
Command> Scroll> Csr

---Name--- ---Value--- ---Description-----
Entry      NBJ.CBLI.INI  CAT - Catalog Entry Name (usually a DSN)
Org        PS          DS - Data Set Organiz'n PS|PO|DA|VS|etc
Trks       1          Alc - Total Tracks
Pri        1          Alc - Primary units
Alu        T          Alc - Allocation units: C|T|B=Cyl|Trk|Blk
Sec        0          Alc - Secondary units
Nxt        1          Alc - Number of extents
Alt        1          Alc - Total units
Blksz     32760     DS - Block Size
Lrecl     32756     DS - Logical record length
RecFm      VB          DS - Record format
PDSE       N          SMS - Partitioned dataset Extended Y|N
DsnPcu    100         Alc - Percent of allocated space used
DsKb       59         Alc - Data space used in Kilobytes
VSeq       1          Vol - Sequence number (Cat) 1 = First
Vol        CBLM09   Vol - Valid
Referenced 2011/12/12  Date - Last Referenced
Created    2011/11/11  Date - Created
Expires    1          Date - Of Expiry
BlkTrk     1          Alc - Blocks per track
CKDKeyL    0          Alc - CKD Physical Key Length
RKP        0          Alc - CKD Relative Key Position
DevC       DASD     Alc - Device Class DASD|TAPE|etc
UnitName   3390       Alc - Device Unit Name
UnitType   3010200F   Alc - Device Unit type
FSeq       0          Alc - Tape File Sequence number
EType      NONVSAM    CAT - Entry Type blank=NONVSAM
T          A          CAT - Entry Type Code
DSInd      82         DS - Dataset flags (DS1DSIND) (hex)
RACF       N          DS - Dataset is RACF defined Y|N
DSType     1          DS - Dataset type PDSE|KSDS|ESDS|RRDS|etc
LastVol    Y          DS - Last vol holding data for dset Y|N
Tbal      4760     DS - TrackBalance: Bytes free on last trk
DCOB       N          SMS - DASDM CREATE originated blksize Y|N
DataClas   1          SMS - Data Class
XATTR      N          SMS - Extended attributes exist Y|N
XFD        N          SMS - Extended format dataset Y|N
Line 1 of 49 | Col 1 of 65 | Views 1 | select *
    
```

Figure 6. 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 omitted. The general syntax is:

```

>>-----+-----+-----+----->>
|         |         |         |         |
| select_clause | where_clause | sort_clause |
    
```

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

Syntax:

```

+----- ALL -----+
>>-- SElect +-----+
|          | |      | |
| +-----+ | | +---+ |
| v         | | |     |
| +---+ columnname +---+ +---+ * ---+
|          | | |      |
>-----+-----+-----+-----><
|    | WHERE Clause |    |    | SORT Clause |    |

```

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.

columnname 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.

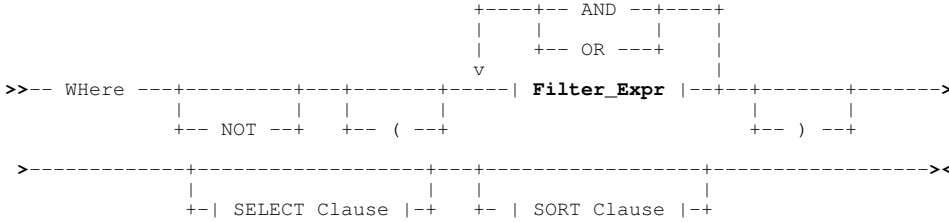
***** Display all remaining column names that have not already been selected, in their default order of display. This is the same as ALL if no other column names are specified.

WHERE Clause Any valid **WHERE clause** used to filter list rows.

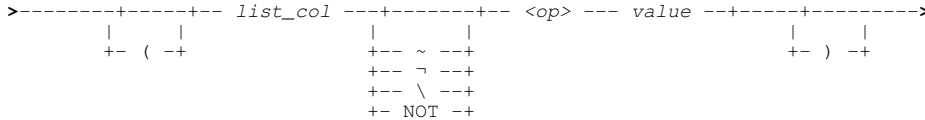
SORT Clause Any valid **SORT clause** used to sort list rows.

The 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 recommended 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. This operator applies to character columns only and returns TRUE if <i>value</i> is a sub-string of <i>list_col</i> .
>>	Begins. This operator applies to character columns only and returns TRUE if <i>value</i> is a sub-string at the start of <i>list_col</i> .

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:

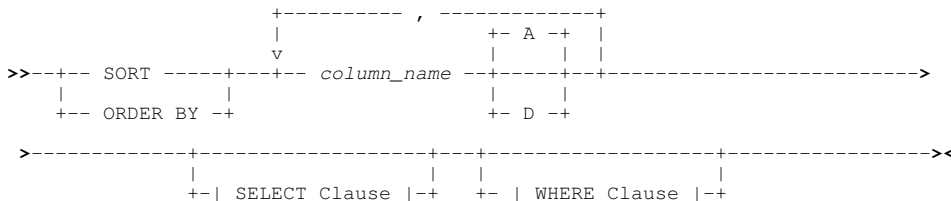
```
where col1 = 'A' and (col2 << 'C' or col3 > 4)
```

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.

The 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:

```
>>-- Find ----- string -----><
>>-- RFIND -----><
```

Description:

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

Following a successful FIND operation, **RFIND** (assigned to PF5 by default) may be used to repeat the search for the remaining list entries.

FIND *string* 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).

Note that key fields area highlighted and remain at a fixed position within the display when scrolling left and right. If multiple key field columns exist within the list, then changing the order of the key fields using a **SELECT** clause, will allow the user to execute

FIND/RFIND on the contents of an alternate key field column.

Parameters:

string

A character search string used in the compare against data within fields in the first column of the display.

LOCATE Command

Syntax:

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

Description:

Starting at the first entry and proceeding 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 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 description 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 CBL text editor, if the list window is a CBL List document window. (i.e. a CBL synonym, command or macro name.)
2. The SELCOPY/i environment command processor. (i.e. a SELCOPY/i command.)
3. TSO, if SELCOPY/i 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 an even number of block markers, each block having an accompanying command.

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 SELCOPY/i INI variable when focus is removed from the panel window, and in the SELCOPY/i User INI file when the SELCOPY/i 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 display 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 SELCOPY/i **window manipulation** techniques. The last customised position and size of an IPO panel window is stored in the SELCOPY/i 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 SELCOPY/i 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 VIEW command may also be used to scroll between panel views. By default, "VIEW +" and "VIEW -" to scroll forwards and backwards through the panel views are assigned to <PF19> and <PF20> respectively.

Closing a secondary panel view using command, END (assigned to <PF3> by default), will return panel focus back to the primary panel view. If END is executed at the primary panel view, then the IPO panel window is closed.

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 <PF10> to scroll left and <PF11> to scroll right.

The entire contents of the field may be expanded into a CBL 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 <PF2> 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.

```

- NBJ2.SELCOPYI.D2011237.T1612370.EXPAND 50 F SEQ -+x
Command> Scroll> Csr

Expanded Character String Edit
Panel: ZS2CV10 Field: SELECT Max Length: 1024
|.....1.....+.....2.....+.....3.....+.....4.....+.....>
000001 SELECT CUSTREF, INVOICE_NUM, DATE
000002 FROM ZSINV.MONTH4
000003 WHERE DATE BETWEEN '04-01-2010' AND '04-30-2010'
000004 UNION ALL
000005 SELECT CUSTREF, INVOICE_NUM, DATE
000006 FROM ZSINV.MONTH5
000007 WHERE DATE BETWEEN '05-01-2010' AND '05-31-2010'
000008 UNION ALL
000009 SELECT CUSTREF, INVOICE_NUM, DATE
000010 FROM ZSINV.MONTH6
000011 WHERE DATE BETWEEN '06-01-2010' AND '06-30-2010'
000012 UNION ALL;
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022 * * * End of File * * *

```

Figure 7. 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 SELCOPY/i 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, SELCOPY/i 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 analogous 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 display 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.
MORE	Filter (include) additional table rows.
QUERY	Query current table edit options.
REDO	Redo changes to the table that have been undone.
RIGHT	Scroll right through the display of table columns.
SELECT	Select table columns and their order of display.
SET	Set table edit options.
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

<i>.name</i>	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 <i>n</i> rows of an excluded row group. F* include all excluded rows.
I(n)	Insert a new row or a block of <i>n</i> new rows.
L(n) L*	Include the last <i>n</i> 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 <i>n</i> 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	REMINDE. (Reinstate previous value for focus entry field.)
F14	EXPAND. (Display focus entry field in a Text-Edit window.)
F15	END
F16	SHOWPOPMENU
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 panel 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 editible 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 explanatory 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 and 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 the row before 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.

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 SELCOPY/i utility panels. (e.g. Compare Files, Create Filter.)

Select Dataset Name (Catalog)

```

SELPCOPY/i - Select SELPCOPY/i Structure Definition Object (SDO) Dataset name
View Refresh Back Forward FDB Text Help
Command>
ZZSGSLD0
Select SELPCOPY/i Structure Definition Object (SDO) Dataset name:
Name> CBL.%%.SDO. (PF1=HELP)
Place the cursor on a list entry and press ENTER to select a dataset.
---Entry--- Org -Trks- -Pri- Alu -Sec- Nxt -Alt- Blksz Lrecl RecFm PDSE DsnP
CBL.CBLI.SDO PO 90 1 C 1 6 6 32760 16380 VB Y
CBL.FT.CSV.SDO PO 15 1 C 1 1 1 32760 16380 VB Y
CBL.PL1.SDO PO 15 1 C 1 1 1 32760 16380 VB Y
CBL.RRD.SDO PO 120 1 C 1 8 8 32760 16380 VB Y
CBL.SEG.SDO PO 1 1 T 2 1 1 32760 16380 VB Y
    
```

Figure 8. SELPCOPY/i - 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.

Select Dataset Name (VTOC)

```

SELPCOPY/i - Select (Print) File to Print Dataset name
View Refresh Back Forward FDB Text Help
Command>
ZZSGSLV0
Select (Print) File to Print Dataset name:
Name> CBL.% Volume> CBLM15 (PF1=HELP)
Place the cursor on a list entry and press ENTER to select a dataset.
-Vol-- Dsn-----
CBLM15 CBL.AON#US.MAPITALL.DATA.IQ003229 PS VB 32756 32760 C
CBLM15 CBL.CBLI.INI PS VB 259 27998 T
CBLM15 CBL.CBLI.MBRLIST.SMALL.KSDS.DATA VS U 0 4096 T
CBLM15 CBL.CBLI.MBRLIST.SMALL.KSDS.INDEX VS U 0 4096 T
CBLM15 CBL.CBLIDUMP.JGE.D2012172.T1634040 PS VB 256 27998 C
CBLM15 CBL.CBLIDUMP.JGE.D2012172.T1634083 PS VB 256 27998 C
CBLM15 CBL.CBLI150.HTML.EDTCONTE PS VB 256 6233 T
CBLM15 CBL.CBLI150.HTML.EDTCOPY PS VB 256 6233 T
CBLM15 CBL.CBLI150.HTML.EDTEDITY PS VB 256 6233 T
CBLM15 CBL.CBLI150.HTML.EDTLOCAT PS VB 256 6233 T
CBLM15 CBL.CBLI150.HTML.EDTQUEUE PS VB 256 6233 T
CBLM15 CBL.CBLI190.QI00014.JCL PO FB 80 32720 T
CBLM15 CBL.CBLI190.RS00007.ADA PO VB 8188 32760 T
CBLM15 CBL.CBLI190.RS00007.LSA PO FBA 121 32670 T
CBLM15 CBL.CBLI190.RS00009.ADA PO VB 8188 32760 T
    
```

Figure 9. SELPCOPY/i - 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

```

SELPCOPY/i - Select (Compare Files) New file HFS path name
View Refresh Back Forward FDB Text Help          wS wR
Command>
ZZSGSLP0
Select (Compare Files) New file HFS path name:
Path> /u/cbl/ige
Place the cursor on a list entry and press ENTER to select an HFS path.
-----Name-----Size-----Modified-----Permis
fc01                d                8192    2012/04/27    16:06:08    rwxr--
fc02                d                8192    2012/04/30    13:40:10    rwxr--
Create.tmp          f                 102     2012/08/31    15:00:07    rwxr--
CBL.AON#US.MAPITALL.DATA  f            75264    2012/03/01    11:09:33    rwxr--
CBL.CBLI310.CM\     f                 73     2011/11/08    12:40:20    rwxr--
CBL.CBLI310.XXX     f                 73     2012/04/26    10:00:49    rwxr--
CBL.SSC.HFSW01      f                 48     2010/12/22    14:31:42    rwxr--
JGE.CompFw.D2011178.170607.CN f                235    2011/11/04    16:45:02    rwxr--
JGE.CompFw.D2011178.170607.CO f                215    2011/08/05    10:22:23    rwxr--
JGE.CompFw.D2011178.170607.DO f            14418    2011/10/26    12:12:35    rwxr--
JGE.CompFw.D2011178.170607.IN f            51539    2011/10/26    12:12:35    rwxr--
JGE.CompFw.D2011178.170607.MN f            97686    2011/10/26    12:12:35    rwxr--
JGE.CompFw.D2011178.170607.MO f            97686    2011/10/26    12:12:35    rwxr--
JGE.CompFw.D2011178.170612.CN f                 0     2011/06/27    16:35:12    rwxr--
JGE.CompFw.D2011178.170612.CO f                 0     2011/06/27    16:35:12    rwxr--
Line 1 of 41 | Col 1 of 643 | Views 1 | select * sort Name,T

```

Figure 10. SELPCOPY/i - 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

```

SELPCOPY/i - Select (Compare Files) New file Member name
View Refresh Back Forward FDB Text Help          wS wR
Command>
ZZSGSLLO
Select (Compare Files) New file Member name:
Library> CBL.JCL
Member>
Place the cursor on a list entry and press ENTER to select a member.
-Member- Alias VV MM -Created-- --LastMod-- CurSize IniSize -Mods- --User-
ABBEY    N      1 27 2006/09/14 2007/12/07 15:15      18      16      18  JGE
ABBEYC   N      1 1 2007/02/16 2007/02/16 09:47      17      17      0  JGE
ABBEYX   N      1 27 2006/09/14 2007/12/07 15:15      18      16      18  JGE
ACB      N      1 2 2008/09/29 2009/07/14 11:11      18      18      0  JGE
ADACOPYT N      1 14 2004/04/26 2009/07/14 11:11      51      18      0  JGE
ADACOPYV N      1 3 2003/08/21 2009/07/14 11:11      21      16      0  JGE
AIXCOPY  N      1 8 2009/08/12 2009/08/19 16:45     152     101      0  JGE
AMAINLK  N      1 39 2010/02/01 2010/12/10 13:11      28      22      0  NBJ
AMBLIST  N      1 4 2005/11/11 2009/07/14 11:11      8        8      0  JGE
AMEBCDIC N      1 2 2008/09/29 2009/07/14 11:11      20      20      0  JGE
AMEX     N      1 3 2009/10/05 2009/10/05 16:31      7        7      0  JGE
AMSDDEL  N      1 3 2010/09/23 2010/09/23 15:11      81      82      0  JGE
AMSUPPM  N      1 6 2008/01/09 2009/07/14 11:11      24      20      0  JGE
APEWTOR  N      1 4 2004/10/25 2009/07/14 11:11      9        8      0  JGE
ASMAPE   N      1 1 2009/08/06 2009/08/06 12:30      17      17      0  JGE
Line 1 of 624 | Col 1 of 87 | Views 1 | select * sort Member

```

Figure 11. SELPCOPY/i - 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 (=)

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

This panel may be started by entering "=" (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 SELCOPY/i 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 SELCOPY/i session.

Help

Open the general help for the Primary Option menu panel.

Options

0	Settings	Set SELCOPY/i 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
T	Training	Setup SELCOPY/i Training Material
WL	Window List	Display active windows, select with cursor to switch focus
X	Exit	Exit SELCOPY/i

Panel Output Fields

User:

An output field displaying the user's logon id.

Version:

An output field displaying the version of SELCOPY/i.

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 output field displays the VM userid of the guest machine or CMS system.

Settings (=0)

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

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

Options

Individual option entries relate to specific functions of the SELCOPY/i 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	

Panel Input/Output Fields

REXX Macro Library Path:

Fields that together establish the library search chain used to locate a SELCOPY/i REXX macro.

User Library>

An input field allowing the user to enter the fully qualified name of one or more PDS/PDSE SELCOPY/i REXX macro libraries to appear first in the macro path search chain.

This field may be expanded (using <PF2<) 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 (<PF3<) 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-existent 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 SELCOPY/i users.

CBL Supplied Library>

A non-enterable (output) field identifying the DSN of the last library in the macro search path.

This library (SZZSDIST.CBLE) contains macros that have been distributed by CBL as part of the SELCOPY Product Suite and contains macros available to all SELCOPY/i 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 SELCOPY/i **Settings** panel.

This panel specifies options relating to SELCOPY/i general operation.

Panel Input Fields

Command Line>

Specifies the location of the command line in all SELCOPY/i 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.

Abend Trap>

Specifies whether the SELCOPY/i abend trap is activated to trap any SELCOPY/i internal system abends and, if possible, recover from the abend.

If set ON, the abend will be trapped and a formatted dump written to a SELCOPY/i 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 SELCOPY/i 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 SELCOPY/i **Settings** panel.

This panel specifies options relating to the SELCOPY/i text editor.

Panel Input Fields

Interface>

Specifies the default edit interface.

The SELCOPY/i 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 SELCOPY/i 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 SELCOPY/i 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 SELCOPY/i text edit. Setting this option to "YES" has the following effect:

◇ ISREDIT (ISPF Edit) macro commands and assignment statements are interpreted by the SELCOPY/i text editor.

◇ Following search of the SELCOPY/i macro path libraries, libraries in the SYSUEXEC and SYSEXEC concatenations are searched for a member name that matches any unrecognised command passed to the SELCOPY/i text editor.

See **ISPF Edit Macros** for details.

Action 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.

Action Delim>

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.

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 SELCOPY/i **Settings** panel.

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

Options

1 **COBOL**
2 **PL/1**

COBOL Compiler and Replacing Options
PL/1 Compiler Options

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 be 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.

Aux Dsn HLQ>

Specifies the dataset name prefix (maximum length 26 characters) to be used by SELCOPY/i SDE when allocating the default Auxiliary Edit data set. Auxiliary edit occurs when editing a non-KSDS data set that is larger than the Max Storage value or the calculated amount of free private area storage above the 16MB line.

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

Default value is %USER%.CBLI.SDEAUX.

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.

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 SELCOPY/i Structured Data (SDE) COBOL copybook support.

Panel Input Fields

COBOL Compiler>

Specifies the full DSN and member name of the COBOL Compiler module (e.g. IGY330.SIGYCOMP(IGYCRCTL).) SELCOPY/i 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 SELCOPY/i 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 member used by SELCOPY/i 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

PL/1 Compiler>

Specifies the full DSN and member name of the PL1 Compiler module (e.g. EL330.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.

List Window Settings (=0.5)

The List Window Settings panel (ZZSGSET5) is an **interactive panel window**, opened on selection of option 5. in the SELCOPY/i **Settings** panel.

This panel specifies options relating to the SELCOPY/i 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 CBL text edit window to edit the file.
View	Open a CBL 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 SELCOPY/i **Settings** panel.

This panel specifies options relating to SELCOPY/i 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 SELCOPY/i, 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 SELCOPY/i.

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 SELCOPY/i'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 SELCOPY/i 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 SDE 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 SELCOPY/i **Settings** panel.

Options

- | | |
|------------------|---------------------------|
| 1 Audit | Audit Log Dataset Options |
| 2 ExecSQL | ExecSQL Output Options |

Panel Input Fields

DB2 SubSystem>

Identifies the DB2 subsystem to which a connection will be made. SELCOPY/i DB2 functions and panels will operate on objects defined in this subsystem.

Current SQLID>

The default SQL authorisation ID for any 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.

The default value for Current SQLID is the user's TSO or SELCOPY/i VTAM logon id.

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 SELCOPY/i **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.Dyyyyddd.Thhmmssx.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.Dyyyyddd.Thhmmssx.SDO**.

Program Function Keys (PF key) Settings (=0.8)

The Program Function Keys Settings panel (ZZSGSET8) is an **interactive panel window**, opened on selection of option 8. from the SELCOPY/i **Settings** panel.

Panel Input Fields

Use **KEYLISTs**
KEYLIST Mode

YES/NO
PRIVATE/SHARED/PRODUCT

Options

- | | |
|-----------------------|--|
| 1 KeyList | Display/Modify loaded KEYLISTs |
| 2 Private | Edit your personal Function Key list table |
| 3 Shared | Edit the site-wide Function Key list table |
| 4 Product | Browse the product supplied Function Key list table |
| 5 Reload | Reload function key definitions from the saved tables. |
| 6 Title Bars | Display/Modify the Window Title-Bars KeyList |
| 7 Borders | Display/Modify the Window Borders KeyList |
| 8 Show All | Show all Function Keys |
| 9 Show Partial | Show partial Function Keys |
| 10 Show None | Remove Function Key display |

Panel Input Fields

Use KEYLISTs>

YES

Use **KeyLists** to control function key definitions and enable **PFSHOW** to display current key settings. KEYLISTs were introduced with SELCOPYi 3.20 for closer compatibility with ISPF and include default key definitions to consistently match ISPF standards
e.g. F1=HELP, F2=SPLIT, F9=SWAP, etc.

NO

Revert to the pre-SELCOPYi Rel 3.20 system whereby function keys are defined by window class, and default key definitions are not specially designed to fit in with ISPF standards. e.g. F9=WINDOW.

KEYLIST Mode>

If **KeyLists** are active, then this setting determines where they are loaded from and whether or not they are modifiable by the individual user.

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.

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.

Options

1. KeyList

This is the equivalent of typing the **KEYLIST** primary command, which starts a dialog that lists all loaded keylists. The dialog allows the user to select each keylist for edit, or for use by the focus window.

2. Private

Edits the user's own personal keylist table as stored on disk. This table contains the definitions for each key in every keylist that is defined at the user level, and is automatically updated when updates are made to a KeyList using the **KEYS** dialog. Standard *Data Edit* techniques may be used to display and/or modify the data. e.g. to show definitions for the F12 only, type:

```
ONLY PF12 #3
```

3. Shared

Edits the installation specific SHARED keylist table as stored on disk. This table should be maintained by your systems programmer in order to define site specific global overrides and will require RACF authority if updates are to be saved.

4. Product

Browses the SELCOPYi PRODUCT supplied keylist table as stored on disk. It should not be modified, but you may find it helpful to cut/paste from this table into either your PRIVATE or SHARED keylist table, before making global changes.

5. Reload

Reloads active KeyList from the saved tables, as determined by the "*KEYLIST Mode*" option below. This option is handy following global updates to the saved tables using Options 1/2, or to revert to previously saved definitions following a temporary override or corruption. Note that PRIVATE function overrides are written to disk only at the end of the SELCOPYi session, or if "*KEYLIST Mode*" is changed from PRIVATE to either SHARED or PRODUCT.

6. Title Bars

Starts the KEYS dialog for the "@WINTITL" KeyList which defines the action of Function Keys pressed while the cursor is placed within the SELCOPYi window "Title-Bar", or caption area. Defaults include:

F7/F8

Drag the location of the window UP/DOWN x 1 position.

F10/F11

Drag the location of the window LEFT/RIGHT x 1 position.

F19/F20 (i.e. Shift-F7/F8)

Drag the location of the window UP/DOWN x 5 positions.

F22/F23 (i.e. Shift-F10/F11)

Drag the location of the window LEFT/RIGHT x 5 positions.

F15/F16 (i.e. Shift-F3/F4)

Shrink/Stretch window width x 20 positions.

F17/F18 (i.e. Shift-F5/F6)

Shrink/Stretch window depth x 20 positions.

7. Borders

Starts the KEYS dialog for the "@WINBORD" KeyList which defines the action of Function Keys pressed actions while the cursor is placed within the SELCOPYi window borders (top/bottom/left/right edges or any of the four corners). Defaults include:

F7/F8

Stretch or shrink the window by dragging the border UP/DOWN x 1 position.

F10/F11

Stretch or shrink the window by dragging the border LEFT/RIGHT x 1 position.

F19/F20 (i.e. Shift-F7/F8)

Stretch or shrink the window by dragging the border UP/DOWN x 5 position.

F22/F23 (i.e. Shift-F10/F11)

Stretch or shrink the window by dragging the border LEFT/RIGHT x 5 position.

F15/F16 (i.e. Shift-F3/F4)

Shrink/Stretch window width x 20 positions.

F17/F18 (i.e. Shift-F5/F6)

Shrink/Stretch window depth x 20 positions.

8. Show All

This is the equivalent of typing the **PFSHOW LONG** primary command, which mimics ISPF's feature to display Function Key definitions at the bottom of the screen.

9. Show Partial

This is the equivalent of typing the **PFSHOW SHORT** primary command, which mimics ISPF's feature to display Function Key definitions at the bottom of the screen. This option will suppress the display of function keys defined as *TYPE=LONG*.

10. Show None

This is the equivalent of typing the **PFSHOW OFF** primary command, which switches off the display of Function Key definitions at the bottom of the screen.

Text Edit (=1)

```
SELCPY/i - Text Edit Entry Panel
File Help
Command>
ZZSGEDIT
PDS/PDSE member, Sequential, VSAM or HFS path:
Name>
Volume> _____ If dataset is uncataloged. + Member> _____

Action:
 / Edit Records. All records are loaded in available storage.
- Read Only Edit (DISP=SHR). Preserve data. (Use SET FILEID before SAVE.)
- Browse Records. Records are loaded/dropped when scrolled.

HFS Options:
 / Undefined - HFS records terminated by End-Of-Line characters.
- Fixed - HFS records are of Fixed Length.
Record Length> 0 0=>default. (Undefined: 32752, Fixed: 80)
EOL Characters> STD EOL user string> _____ (2-byte char/hex)

Profile Macro:
 / Use Default.
- Use Specified Macro. Macro Name> PROFILE
- Do not use a Profile Macro.
```

Figure 12. SELCPY/i - Text Edit Entry Panel.

The Text Edit Entry panel (ZZSGEDIT) is an **interactive panel window**, opened on selection of option 1. in the SELCPY/i Primary option menu.

This panel is used to open an existing cataloged or uncataloged data set PDS/PDSE library member or HFS file for CBL text edit. For full documentation on text edit sessions, please refer to publication "*SELCPY/i Text Editor (CBL)*".

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 either action the data edit or display the **Structured Data Browse/Edit - Options** panel view.

```
SELCOPY/i - Structured Data Browse/Edit
File Command Structure Replace Help
Command>
ZZSGSDE0
PDS/PDSE member, Sequential, VSAM or HFS path:
Name>
Volume> If dataset is uncataloged. + Member>

Action:
Z Browse Data.
Edit Full. (Insert/Update/Delete)
Edit In-Place. (Update only)
Edit Full Auxiliary. (AUX File)
Edit Full Read-Only. (DISP=SHR)
Edit Full Read-Only & Auxiliary.

Structure/Copybook overlay:
Dsn>
Type: SDO AData Cobol PL1
(PFS=Edit Copybook)
Member>

Record Selection:
Start>
For> @ # records + Record Key RBA
Filter> Q Filter selected records. (F=File; Q=Quick)
File> Member>

Additional Options: (Enter "/" to display HFS and Profile options.)
```

Figure 13. 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 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.

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 successfully 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 (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 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> Y/N

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. 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 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 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.

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.

```

SELCOPY/i - Structured Data Browse/Edit - Options
File Command Structure Replace Help          wS wR      Scroll> Csr
Command>                                     Lines 1-20 of 20
ZZSGSDE0

HFS Options:      (Ignored for edit/browse of non-HFS data set.)
< Undefined - HFS records terminated by End-Of-Line characters.
- Fixed - HFS records are of Fixed Length.
- Variable - HFS records are of Variable Length.
  Max Record Length> 0          0=> default. (Und/Var: 32752, Fix: 80)
  EOL Characters>  STD          EOL user string> (2-byte char/hex)
  Var Length Field>  0 (Offset) / 2 (Length) / 0 (Data Origin)

Profile Macro:
< Use Default. (SDEPROF)
- Use Specified Macro.
- Do not use a Profile Macro.          Macro Name> SDEPROF
  
```

Figure 14. 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>

Identifies 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)

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 Associations
12 HFS	LP - List HFS Paths
13 DB2	LDxx - 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 **CBL** 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.

The screenshot shows a window titled "DASD Volumes: CBL*" with a menu bar (View, Back, Forward, FDB, Edit, Refresh, Help) and a command prompt. The main display is a table of volume attributes. The status bar at the bottom indicates "Line 1 of 10", "Col 1 of 241", "Views 1", and "select * sort UNIT,VOL".

UNIT	-VOL--	FREECYL	FREETRK	FREEXTN	FREEDSCB	MAXCYL-	MAXTRK-	VOLPCU	VTOCPC
0AA0	CBLMCT	2074	31134	9	645	1865	27975	38	
0AA1	CBLM01	137	2128	32	366	30	450	96	4
0AA2	CBLM02	554	8350	14	182	490	7362	84	7
0AA3	CBLM03	136	2099	16	219	126	1890	96	6
0AA4	CBLM04	90	1565	48	237	30	450	97	6
0AA5	CBLM05	110	1824	57	250	81	1215	97	6
0AA6	CBLM06	445	6899	70	310	209	3142	87	5
0AA7	CBLM07	1478	22274	27	7211	1472	22080	86	
0AA8	CBLM08	929	13972	15	7242	783	11745	91	
0AEF	CBLM09	3337	50055	1	697	3337	50055	1	

Figure 15. 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.
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.
?	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 PF2 by default.

Columns Displayed

The data displayed for MVS 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

The data displayed for 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 [CBL 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.

```

VTOC File List: CBLM01
View Back Forward FDB Edit Refresh Help
Command>
Volume> CBLM01
Filter> CBL.*.*
-----Vol-----Dsn-----Org RecFm Lrecl Blksz
CBLM01 CBL.APFLIB PO U 32000 4096
CBLM01 CBL.ASM.TEST.JCL PO FB 80 23440
CBLM01 CBL.ASM.TEST.LSA PO FBA 121 23474
CBLM01 CBL.ASM.TEST.MAC PO FB 80 23440
CBLM01 CBL.ASM.TEST.OBJ PO FB 80 3120
CBLM01 CBL.CA.CAESDR.OBJ PS FB 80 27920
CBLM01 CBL.CA.CS11.CAESDR.REP PS VBA 240 27998
CBLM01 CBL.CA.CS11.ESD PS U 0 4096
CBLM01 CBL.CAI.CAIPROC PO FB 80 3120
CBLM01 CBL.CAI.CASCRN PO V 4100 4104
CBLM01 CBL.CAI.CLJ43SLD PO FB 80 3120
CBLM01 CBL.CAI.CLU43ETL PO FB 80 3120
CBLM01 CBL.CAI.CS11.LST PO FB 133 32718
CBLM01 CBL.CAI.F331.CF331MLD PO FB 80 3120
CBLM01 CBL.CAI.LIBR.MAST.CBL1 DA F 0 1086
CBLM01 CBL.CAI.LIBR.SAMPJCL PO FB 80 3120
CBLM01 CBL.CAI.SMPSCDS PO FB 80 3120
Line 1 of 217 | Col 1 of 246 | Views 1 | select * sort Vol,Dsn

```

Figure 16. 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.

Filter> **Note:** The filter parameter is not supported for 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.
AS	Open an Associations list window to list associated objects for this entry.
AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.
B	Open the CBL 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 CBL text editor to edit this entry.
EU	Open the SDE structured data editor to edit the entry in update mode only.
F	For MVS only, 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.
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 MVS 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 .
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBL 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 PF2 by default.

Columns Displayed

Name	Type	Description
Vol	Char	Volume serial number
Dsn	Char	Dataset name
Org	Enum	Data set organisation
RecFm	Enum	Record format
Lrecl	UInt	Logical record length
Blksz	UInt	Block Size
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
Created	VTOCDate	Creation date
Referenced	VTOCDate	Last referenced date
Expires	VTOCDate	Expiry Date
SMS	BitFlag	SMS managed dataset
PDSE	BitFlag	PDS extended
HFS	BitFlag	HFS dataset
UnCat	BitFlag	Uncataloged dataset
XFD	BitFlag	Extended format dataset
XAttr	BitFlag	Extended attributes exist
ReBlk	BitFlag	May be reblocked
IsICF	BitFlag	ICF catalog BCS dataset
InICF	BitFlag	Cataloged in an ICF catalog
DSInd	Hex	Dataset indicators (DS1DSIND)
KyL	UInt	Dataset key length
RKP	UInt	Relative key position
TBal	UInt	Bytes remaining on last track
BIKTrk	UInt	Blocks per track
F1Vol	Char	Format 1 DSCB volume serial
VSeq	UInt	Volume sequence number

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 **CBL** 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 VSE.

```

VTOC Extent List: CBLM01
View Back Forward FDB Edit Refresh Help
Command>
Volume> CBLM01
-Vol-- -CC-- -HH-- Seq -----Dsn----- Org Al
--- CBLM01      0    0    1  **Label Area**
--- CBLM01      0    1    0  CBL.MODEL
--- CBLM01      1    0    1  **VTOC**
--- CBLM01      1    0    1  SYS1.VTOCIX.CBLM01      PS  T
--- CBLM01      2    0    1  CBL.JCL.ORIG            PO  C
--- CBLM01      3    0    1  SYS1.VVDS.VCBLM01      VS  T
--- CBLM01      3    10   1  CBL.S200.LONG.CTL.FILE.DATAS  PS  T
--- CBLM01      3    11   1  CBL.SSC.@ZDS.CBL.SSC.@      VS  T
--- CBLM01      3    12   1  CBL.SQ11181.PDS          PO  T
--- CBLM01      3    13   1  CBL.SQ11180.TEMP        PS  T
--- CBLM01      3    14   1  LAC.MULTIVOL.KSDS.DATA     VS  T
--- CBLM01      4    0    1  CBL.ISPMLIB             PO  C
--- CBLM01      5    0    1  CBL.EXEC                PO  C
--- CBLM01     10    0    1  CBL.CBL200.OBJ          PO  T
--- CBLM01     10    1    2  CBL.CBL200.OBJ          PO  T
--- CBLM01     10    2    3  CBL.CBL200.OBJ          PO  T
--- CBLM01     10    3    4  CBL.CBL200.OBJ          PO  T
--- CBLM01     10    4    5  CBL.CBL200.OBJ          PO  T
Line 1 of 727 | Col 1 of 107 | Views 1 | select * sort Vol,CC,HH

```

Figure 17. VTOC Extent List window displaying all extents on volume 'CBLM01.'

Panel Input Fields

Volume>

The 1-6 character volume id containing the required VTOC.

Filter>

Note: The filter parameter is not supported for 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.
AS	Open an Associations list window to list associated objects for this entry.
AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.
B	Open the CBL 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 CBL 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.
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 .
UT	Opens the general file utilities menu to ultimately generate specific line commands in a temporary CMX file.
V	Open the CBL 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 PF2 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)

```

Open File
File Help
Command>
ZZSGOPEN
Scroll> Csr
Lines 1-13 of 20

Open File: PDS(E) member, Sequential, VSAM or HFS path
Dsn/Path> _____ +
Volume> _____ If dataset is uncataloged.

_ Use (TSO) Prefix
ENTER Key Action> Edit
(Leave blank for a list of available options)
From a file-type list-window e.g. LISTDATASET (LD)
LISTVOLUME (LV), LISTLIBRARY (LL) etc, this is the
"Select" action used when the ENTER key is pressed
with the cursor on a list-row, or the "S"
line-command is issued.

```

Figure 18. SELCOPY/i - Data Set List Utility.

The Data Set List Utility panel (ZZSGOPEN) is an **interactive panel window**, opened on selection of option 4. 'Dslst' 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 CBL text edit.

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 CBL text editor with Read/Write authority.
View	Edit using the CBL 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 [CBL main window menu bar](#).
- Enter command **LC** 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](#).

For 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 MVS, 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.

```

Catalog List: CBL.A*.**                               2011/12/07 14:17 -+x
View Refresh Back Forward FDB Text Help
Command>
Entry> CBL.A*.**
Catalog> USERCAT.CBLCAT
Types> BA
AllVols> N
-----Entry----- VSeq -Vol-- VTot DevC FSeq T -EType- ----
CBL.ACS.TRAN.LST          1 CBLM09    1 DASD  0 A NONVSAM
CBL.ADCD.CBLI.CMX        1 CBLM03    1 DASD  0 A NONVSAM PDSE
CBL.ADCD.TEST            1 CBLM06    1 DASD  0 A NONVSAM
CBL.AIRPORTS.BIN        1 CBLM07    1 DASD  0 A NONVSAM
CBL.AIRPORTS.CSV        1 CBLM08    1 DASD  0 A NONVSAM
CBL.AM.G1465.TXT        1 CBLM05    1 DASD  0 A NONVSAM
CBL.AM.G1621.TXT        1 CBLM07    1 DASD  0 A NONVSAM
CBL.AM.G1645.TXT        1 CBLM10    1 DASD  0 A NONVSAM
CBL.AM.LOAD              1 CBLM04    1 DASD  0 A NONVSAM
CBL.AM.LOAD.SQ10152     1 CBLM04    1 DASD  0 A NONVSAM
CBL.AMALL.DA             1 CBLM02    1 DASD  0 A NONVSAM
CBL.AMALL.EBCDIC.DA     1 CBLM07    1 DASD  0 A NONVSAM
CBL.AMALL.G1465.DA      1 CBLM08    1 DASD  0 A NONVSAM
CBL.AMCUST.G1465.DA     1 CBLM07    1 DASD  0 A NONVSAM
CBL.AMCUST.G1516.DA     1 CBLM02    1 DASD  0 A NONVSAM
CBL.AMCUST.G1586.DA     1 CBLM07    1 DASD  0 A NONVSAM
CBL.AMCUST.G1621.DA     1 CBLM10    1 DASD  0 A NONVSAM
CBL.AMCUST.G1645.DA     1 CBLM10    1 DASD  0 A NONVSAM
CBL.AMCUST.G1647.DA     1 CBLM11    1 DASD  0 A NONVSAM
CBL.AMEX.CTL             1 CBLM03    1 DASD  0 A NONVSAM PDSE
Line 1 of 93 | Col 1 of 141 | Views 1 | select * sort Entry,VSeq,Vol

```

Figure 19. MVS Catalog List Window.

```

Catalog List: *                                       -+x
View Back Forward FDB Edit Refresh Help
Command>
Entry> *
Catalog> CBLUCT2
Types> AC
-----DSN----- --TYPE-- --NRECS-- --PCNT-- --ALLOCT-- ALLOCU -AL
CBL.DBXRRDS.RRDS        RRDS (R)          5          0.5          1
CBL.LIBR.CBLLIB1       SAM              28800+    ** ALL**      C=160
CBL.LIBR.CBLLIB2       SAM              36000+    ** ALL**      C=200
CBL.SQ11473.SAM        SAM (R)           1+        16.7          1
CBL.SQ11564.SAM        SAM (R)           0( 240)    TEMP          1
CBL.SQ11630.KSDS       KSDS (R)          0( 972)    1
CBL.SQ11637.KSDS       KSDS (R)          5          0.6          1
CBL.SQ11641.ESDS       ESDS (R)          0( 972)    1
CBL.SYSADATA.APEEINIT  SAM (R)          27+        73.0          25
CBL.SYSADATA.APEETERM SAM (R)          13+        72.3          12
CBL.SYSADATA.CBLAVARL SAM (R)          27+    **90.0**      20
CBL.SYSADATA.CNVFPRTF SAM (R)          28+        75.7          25
CBL.SYSADATA.CNVPTLE0 SAM (R)          18+        75.0          16
CBL.SYSADATA.CVHNBJ2   SAM (R)           6+         3.4          C=2
CBL.SYSADATA.CVHTEST   SAM (R)           8+         4.5          C=2
CBL.SYSADATA.CVHTEST2 SAM (R)           6+         3.4          C=2
Line 1 of 147 | Col 1 of 567 | Views 1 | select * sort DSN

```

Figure 20. VSE Catalog List Window.

Panel Input Fields

Entry>

Specify the fileid mask.

◇ 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.

Catalog>

Nominate a specific catalog in which to search for the requested entry.

For **MVS** systems, this is a catalog DSN. Specifying a catalog DSN is unnecessary 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.

For **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	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.

Prefix Line Commands

For **MVS** 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.
AS	Open an Associations list window to list associated objects for this entry.
AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.
B	Open the CBL 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 CBL 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.
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 CBL 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 PF2 by default.

For **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 PF2 by default.

Columns Displayed

For **MVS** systems, the data displayed is:

Name	Type	Description
Entry	Char	Entry name
Vol	Char	Volume serial number
VSeq	UInt	Volume Sequence number
DevC	Char	Device Class
FSeq	UInt	File Sequence number
T	Char	Catalog entry type code
EType	Char	Catalog entry type
DSType	Char	Dataset type
UnitType	Hex	Unit type
UnitName	Char	Unit name
DataClas	Char	SMS Data Class
StorClas	Char	SMS Storage Class
MgmtClas	Char	SMS Management Class

For **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 Message 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.
TIMESTAMP	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 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 CMS, the **File List** window is opened in place of the MVS/VSE Catalog List or Dataset List windows and displays information about files residing on accessed disks.

```

File List: * * A
View Back Forward FDB Edit Refresh Help
Command>
File> * * A
  ---Fn---  ---Ft---  Fm  --LRecL--  Fmt  --nRecs--  --nBlks--  ---TimeStamp--
---EIPLESA  PROC      A5      72  V      51      1  1997-06-04  16:00
---##NFS##  #NAMES#   A1      64  F      64      1  2004-08-03  17:29
---cvea-djh  tab       A1      256 F      1      1  2000-04-07  13:24
---hello    MODULE    A1     3704 V      3      1  2006-07-19  15:50
---t_vm01    MODULE    A1     5144 V      3      2  2006-07-25  17:50
---A        ADMP#N    A1     134  V      24      1  2004-08-03  17:29
---A        MACRO     A1      80  F     4497     88  2007-04-05  15:09
---ABC      ZAP       A5      80  F      10      1  2004-03-30  16:18
---ADDLBL   JCL       A5      71  V     127     2  2007-03-14  16:30
---AM       HIST      A5      70  V     16      1  1999-08-27  15:40
---AMPMEML  EXEC      A5      64  V     31      1  2002-05-08  11:09
---AMPMEML  EXEC      A1      66  V     31      1  2006-09-06  17:01
---AMSITE   EXEC      A5      64  V     53      1  2001-12-11  14:54
---AMSITE   EXEC      A1      66  V     53      1  2006-09-06  17:01
---AMUPDC   EXEC      A5     108  V     117     1  2002-02-14  15:06
---AMUPDC   EXEC      A1     110  V     117     1  2006-09-06  17:01
---APEAVDBT RUN       A1      80  V     1615    33  2002-10-04  12:51
---ASMCBLN  EXEC      A1      80  F     818    16  2007-05-08  10:57
Line 1 of 652 | Col 1 of 108 | Views 1 | select * sort Fn,Ft,Fm

```

Figure 21. 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 than 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 CBL 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 CBL 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 PF2 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 **CBL** 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.

```

Dataset List: NBJ.*.** 2011/12/07 15:43
View Refresh Back Forward FDB Text Help
Command>
Entry> NBJ.*.**
Catalog> USERCAT.CBLCAT
Types> CG
AllVols> N
-----Entry----- Org -Trks- -Pri- Alu -Sec- Nxt -Alt-
NBJ.CBL.EMP.D2010190.KSDS VS 0 0 0 0 0 0
NBJ.CBL.EMP.D2010190.KSDS.DATA VS 5 5 T 5 1 5
NBJ.CBL.EMP.D2010190.KSDS.INDEX VS 1 1 T 1 1 1
NBJ.CBLIDEMO.KSDS VS 0 0 0 0 0 0
NBJ.CBLIDEMO.KSDS.DATA VS 30 2 C 4 1 2
NBJ.CBLIDEMO.KSDS.INDEX VS 1 1 T 1 1 1
NBJ.CBLIDEMO.V0000.KSDS VS 0 0 0 0 0 0
NBJ.CBLIDEMO.V0000.KSDS.DATA VS 84 28 T 7 5 84
NBJ.CBLIDEMO.V0000.KSDS.INDEX VS 1 1 T 1 1 1
NBJ.CBLINST.CBL11091.CSI VS 0 0 0 0 0 0
NBJ.CBLINST.CBL11091.CSI.DATA VS 315 21 C 17 1 21
NBJ.CBLINST.CBL11091.CSI.INDEX VS 75 5 C 9 1 5
NBJ.CBLINST.CBL11091.SILOG VS 0 0 0 0 0 0
NBJ.CBLINST.CBL11091.SILOG.DATA VS 15 1 C 1 1 1
NBJ.CBLINST.CBL11091.SILOG.INDEX VS 1 1 T 1 1 1
NBJ.DATASET.BASIC01.KSDS VS 0 0 0 0 0 0
NBJ.DATASET.BASIC01.KSDS.DATA VS 1 1 T 1 1 1
NBJ.DATASET.BASIC01.KSDS.INDEX VS 1 1 T 1 1 1
NBJ.DATASET.COPY.KSDS VS 0 0 0 0 0 0
NBJ.DATASET.KSDS VS 0 0 0 0 0 0
NBJ.DATASET.KSDS.DATA VS 1 1 T 1 1 1
NBJ.DATASET.KSDS.INDEX VS 1 1 T 1 1 1
Line 1 of 61 Col 1 of 361 Views 1 select * sort Entry,VSeq,Vol

```

Figure 22. Dataset List window displaying all Cluster and AIX entries beginning 'CBL.'

Panel Input Fields

Entry>

Specify the fileid mask.

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.)

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 unnecessary 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> 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.

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, using the entry name as the Audit DSN field entry.
B	Open the CBL 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 CBL 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.
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 CBL 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 PF2 by default.

Columns Displayed

Name	Type	Description
Entry	Char	CAT - Catalog Entry Name (usually a DSN)
Org	Enum	DS - Data Set Organiz'n PS PO DA VS etc
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
Blksz	UInt	DS - Block Size
Lrecl	UInt	DS - Logical record length
RecFm	Enum	DS - Record format
PDSE	BitFlag	SMS - Partitioned dataset Extended Y N
DsnPcu	UInt	Alc - Percent of allocated space used
DsKb	UInt	Alc - Data space used in Kilobytes
VSeq	UInt	Vol - Sequence number (Cat) 1 = First
Vol	Char	Vol - Volid
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
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)
VTot	UInt	Vol - Total no of volumes
F1Vol	Char	Vol - Valid of 1st volume (Format 1 DSCB)
VOLSq	UInt	Vol - Sequence number (VTOC) 1 = First

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 (MVS) or LIBR (VSE) library.

```

Library List: PRD
View Back Forward FDB Edit Refresh Help
Command>
Library> PRD
Lib-----DSN-----CreDate- CreTime- --SubLi
PRD1 VSE.PRD1.LIBRARY 07-07-06 15:25.27
PRD2 VSE.PRD2.LIBRARY 07-07-06 15:25.29
Line 1 of 2 | Col 1 of 158 | Views 1 | select * sort Lib
  
```

Figure 23. Library List window displaying VSE libraries beginning 'PRD'.

```

Library List: CBLLIB2.CB*
View Back Forward FDB Edit Refresh Help
Command>
Library> CBLLIB2.CB*
--Lib-- -SubLib- CreDate- CreTime- --Members-- -BlksUsed-- ---Size--- --Lo
CBLLIB2 CBLI1150 08-05-21 14:47.02 1254 21493 0
CBLLIB2 CB070128 07-04-12 17:06.58 1154 11843 0
Line 1 of 2 | Col 1 of 84 | Views 1 | select * sort Lib,SubLib
  
```

Figure 24. Library List window displaying VSE library 'CBLLIB2' sub-libraries beginning 'CB'.

```

Library List: CBLLIB2.CB070128.*.*
View Back Forward FDB Edit Refresh Help
Command>
Library> CBLLIB2.CB070128.*.*
--Lib--  -SubLib-  -Member-  --Type--  Recfm  --Records--  ---Lrecl---  --Blocks--
CBLLIB2  CB070128  ABTRAP01  HTML      S      1      4525
CBLLIB2  CB070128  ACCESSED  HTML      S      1      1642
CBLLIB2  CB070128  ADABAS00  HTML      S      1      38972      4
CBLLIB2  CB070128  ADABAS01  HTML      S      1      4924
CBLLIB2  CB070128  ADD000001 HTML      S      1      8874
CBLLIB2  CB070128  AGENTS    HTML      S      1      3754
CBLLIB2  CB070128  AGENTS00  HTML      S      1      7567
CBLLIB2  CB070128  ALIAS     HTML      S      1      7203
CBLLIB2  CB070128  ALLFILES  HTML      S      1      2330
CBLLIB2  CB070128  ALLOC     HTML      S      1      21445      2
CBLLIB2  CB070128  ALL000001 HTML      S      1      2843
CBLLIB2  CB070128  AND000001 HTML      S      1      13300      1
CBLLIB2  CB070128  APPEND01  HTML      S      1      4530
Line 1 of 1154 | Col 1 of 151 | Views 1 | select * sort Lib,SubLib,Member,Typ

```

Figure 25. Library List window displaying all members of VSE sub-library 'CBLLIB2.CB070128'.

```

Library List: CBL.JCL(S*)
View Back Forward FDB Edit Refresh Help
Command>
Library> CBL.JCL(S*)
-Member-  Alias  VV  MM  -Created--  ---LastMod---  CurSize  IniSize  -Mods-  --
SELCLCTL  N      1  1  2002/07/16  2002/07/17  11:09    12      7      0  LA
SELCLKED  N
SELCMJ01  N
SELCNAMT  N      1  3  2006/11/28  2007/04/16  17:31    17      18      0  JG
SELCPDSX  N      1  5  2004/02/11  2006/03/27  15:57    57      49      0  JG
SELDBIMS  N      1  0  2002/04/22  2002/04/22  09:52    10      10      0  LA
SELPDSEU  N      1  14 2008/11/14  2008/11/20  11:16    27      21      0  JG
SMPE0001  N      1  28 2005/09/08  2006/08/03  16:40    29      25      0  JG
SMPE0002  N      1  2  2005/09/09  2005/09/09  15:34    25      25      0  NB
SMPE0003  N      1  5  2005/09/12  2005/09/12  12:50    14      25      0  NB
SMPE0004  N      1  1  2005/09/12  2005/09/12  12:52    14      14      0  NB
SMPE0005  N      1  16 2005/09/12  2005/09/12  16:51    17      36      0  NB
SMPE0006  N      1  3  2005/09/12  2005/09/12  16:51    17      17      0  NB
SMPE0007  N      1  13 2006/03/01  2006/03/01  11:18    17      16      0  NB
SMPE0008  N      1  2  2006/03/09  2006/03/09  14:38    26      26      0  NB
Line 9 of 84 | Col 1 of 90 | Views 1 | select * sort Member

```

Figure 26. Library List window displaying members of MVS 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.

- ◇ For **MVS** the library parameter is a PDS (or PDSE) dataset name and optionally one or more member name mask.

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)
```

- ◇ For **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
```

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 CBL 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 CBL 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.
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 CBL CLI SUBMIT command using the entry name as input. (A CBL frame window must be active for this operation to succeed.)
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 CBL 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 PF2 by default.

For **VSE** systems, the following prefix line commands are available when a list of 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 VSE LIBR member. (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 PF2 by default.

For **VSE** systems, the following prefix line commands are available when a list of 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 CBL e 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 CBL
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 CBL e 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 PF2 by default.

Columns Displayed

The data displayed for MVS 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 MVS 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
Fvl	BitFlag	Only 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 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 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 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 [CBL main window menu](#) bar.
- Enter command [LA](#) on the command line of any window.

The resultant Allocated Datasets window displays all current file allocations defined to the environment running SELCOPY/i. i.e.

MVS TSO	DDnames allocated to the TSO Userid.
MVS VTAM	Files allocated to the SELCOPY/i VTAM application.

```

Allocated Datasets
View Back Forward FDB Edit Refresh Help
Command>
DDName>
- DDName- -CSeq- -----DsN----- -Member- -Vol
--- AOFPRINT      1  NBJ2.NBJ2.TSU00119.D0000101.?
--- AOFTABL       1  AUT310.AOFTABL
--- DITPLIB       1  DIT130.SDITPLIB
--- IHVCONF      1  AUT310.IHVCONF
--- ISPEXEC       1  ISP.SISPEXEC
--- ISPEXEC       2  SYS1.SBPXEXEC
--- ISPEXEC       3  CSQ600.SCSQEXEC
--- ISPEXEC       4  EUV.SEUVEXEC
--- ISPLLIB       1  CBL.SSC.EXE
--- ISPLLIB       2  CBL.ISPLLIB
--- ISPLLIB       3  DSU.ISPLLIB
--- ISPLLIB       4  GEN.ISPLLIB
--- ISPLLIB       5  GDDM.SADMMOD
--- ISPLLIB       6  FMN810.SFMNMOD1
--- ISPLLIB       7  CSQ600.SCSQAUTH
--- ISPLLIB       8  AUT310.SINGMOD1
--- ISPLLIB       9  DSN910.SDSNLOAD
--- ISPLLIB      10  CBL.CAI.CAILIB
Line 1 of 162 | Col 1 of 109 | Views 1 | select * sort DDName,CSeq
  
```

Figure 27. 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 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.
AS	Open an Associations list window to list associated objects for this entry.

AP	Open the DB2 Print Audit Report panel for this entry, using the entry name as the Audit DSN field entry.
B	Open the CBL 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 CBL 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.
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 .
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 CBL 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 PF2 by default.

Columns Displayed

Name	Type	Description
DDName	Char	DD name
CSeq	Int	Concatenation 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 **CBL** 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.

```

Standard Labels
View Back Forward FDB Edit Refresh Help
Command>
DDName>
  PN PT -File-- -----DSN----- 0 VSAMCat -Vol--
    SYSUCT2 USER.DL1.CAT.SYSWK2      A VSAMCat  SYSWK1
    SYSUCT7 USER.CAT.SYSWK7        A VSAMCat  SYSWK1
    TRFILE  VTAM.TRACE.FILE          S VSAMCat  SYSWK1
    VSEJMGR VSESP.JOB.MANAGER.FILE   S VSAMCat  SYSWK1
    VSESPUC VSESP.USER.CATALOG      A VSAMCat  SYSWK1
  F2 T IJSYS01 %DOS.WORKFILE.SYS001.RECOVER A VSESPUC  SYSWK1
  F2 T IJSYS02 %DOS.WORKFILE.SYS002.RECOVER A VSESPUC  SYSWK1
  Z1 T BB      VSESP.USER.CATALOG    A BB       SYSWK1
  Z1 T CATWK1  VSESP.USER.CATALOG    A CATWK1   SYSWK1
  Z1 T CBLMULT CBL.MULT.EXT.FILE.BG.VERY.LONG.DSN S VSAMCat  SYSWK2
    CBL.MULT.EXT.FILE.BG.VERY.LONG.DSN S VSAMCat  SYSWK3
    CBL.MULT.EXT.FILE.BG.VERY.LONG.DSN S VSAMCat  SYSWK1
  Z1 T CBLTEMP CBL.TEMP.LABEL.BG        S VSAMCat  SYSWK1
  Z1 T CBLVSAM CBL.VSAM.LABEL.BG      A VSAMCat  SYSWK1
  Z1 T IJSYSRS CBL.IJSYSRS.WITH.NO.LUB  S VSAMCat  SYSWK1
  Z1 T IJSYSRX CBL.IJSYSRX.WITH.NO.LUB  S VSAMCat  SYSWK1
  Z1 T IJSYSR6 CBL.IJSYSR6.WITH.NO.LUB   S VSAMCat  SYSWK1
  Z1 T IJSYSXX CBL.IJSYSIN.WITH.NO.EXT    S VSAMCat  SYSWK1
Line 82 of 103 | Col 1 of 139 | Views 1 | select * sort PN,PT,File

```

Figure 28.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 PF2 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 [CBL 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.

```

Enqueues: SYSDSN
View Back Forward FDB Edit Refresh Help
Command>
Queue Name> SYSDSN
Resource Name> SYS1
  --JOB--  --QNAME--  --STATUS--  --SCOPE--  RSV  MC  OWN-  WTEX  WTSH  RNAMEL  -----R
-----
JGE      SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SBLS
LAC      SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SBLS
LAC3     SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SBLS
JGE2     SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SBLS
NBJ2     SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SBLS
NBJ      SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SBLS
JGE      SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SBLS
LAC      SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SBLS
LAC3     SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SBLS
JGE2     SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SBLS
NBJ2     SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SBLS
NBJ      SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SBLS
JGE      SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SCBD
LAC      SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SCBD
LAC3     SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SCBD
JGE2     SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SCBD
NBJ2     SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SCBD
NBJ      SYSDSN    SHR OWN    SYSTEM    N    N    6     0     0     13    SYS1.SCBD
Line 1 of 301 | Col 1 of 90 | Views 1 | select *

```

Figure 29. 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 PF2 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.

```

Job Enqueues: LAC
View Back Forward FDB Edit Refresh Help
Command>
JobName> LAC
--JOB--  --QNAME--  --STATUS--  --SCOPE--  RSV  MC  OWN-  WTEX  WTS  RNAME
-----
LAC     SPFUSER   EX  OWN   STEP   N   N   1     0     0     7     LAC
LAC     SYSDSN   SHR  OWN   SYSTEM N   N   6     0     0     8     DSU.EXEC
LAC     SYSDSN   EX   OWN   SYSTEM N   N   1     0     0     35    SYS09035.
LAC     SYSDSN   SHR  OWN   SYSTEM N   N   6     0     0     12    EOY.SEOYC
LAC     SYSDSN   SHR  OWN   SYSTEM N   N   6     0     0     15    CSQ600.SC
LAC     SYSDSN   EX   OWN   SYSTEM N   N   1     0     0     16    LAC.ISPF.
LAC     SPFEDIT  EX   OWN   SYSTEMS N  N   1     0     0     52    LAC.ISPF.
LAC     SYSDSN   EX   OWN   SYSTEM N   N   1     0     0     35    SYS09035.
LAC     SYSDSN   SHR  OWN   SYSTEM N   N   6     0     0     14    AUT310.AO
LAC     SYSDSN   SHR  OWN   SYSTEM N   N   6     0     0     13    SYS1.SBLS
LAC     SYSDSN   SHR  OWN   SYSTEM N   N   6     0     0     14    FAN140.SF
LAC     SYSDSN   EX   OWN   SYSTEM N   N   1     0     0     35    SYS09035.
LAC     SYSDSN   SHR  OWN   SYSTEM N   N   6     0     0     14    AUT310.IH
LAC     SYSDSN   SHR  OWN   SYSTEM N   N   6     0     0     13    SYS1.SBLS
LAC     SYSDSN   EX   OWN   SYSTEM N   N   1     0     0     35    SYS09035.
LAC     SYSDSN   SHR  OWN   SYSTEM N   N   7     0     0     15    CSQ600.SC
LAC     SYSDSN   SHR  OWN   SYSTEM N   N   6     0     0     13    SYS1.SCBD
LAC     SYSDSN   SHR  OWN   SYSTEM N   N   6     0     0     12    IOE.SIOEE
Line 1 of 200 | Col 1 of 120 | Views 1 | select *

```

Figure 30. 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 PF2 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 [CBL 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 Alternate Index objects with which it is associated.

Note: Not implemented for CMS and VSE.

```

Associations: CBL.OPNEMPTH*.**
View Back Forward FDB Edit Refresh Help
Command>
Entry> CBL.OPNEMPTH*.**
Catalog> USERCAT.CBLCAT
Types>
-----Assoc----- A -----Entry----- T
CBL.OPCODES.DATA      D CBL.OPNEMPTH      R
CBL.OPCODES.INDEX    I CBL.OPNEMPTH      R
CBL.OPNEMAIX         G CBL.OPNEMPTH      R
CBL.OPNEMAIX.DATA    D CBL.OPNEMPTH      R
CBL.OPNEMAIX.INDEX   I CBL.OPNEMPTH      R
Line 1 of 5 | Col 1 of 76 | Views 1 | select * sort Assoc,Entry
  
```

Figure 31. 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 unnecessary 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.
AS	Open an Associations list window to list associated objects for this entry.
B	Open the CBL 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 CBL 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 CBL 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 PF2 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 [CBL 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.

```

HFS Path: /etc
View Back Forward FDB Edit Refresh Help
Command>
HFS Path> /etc
Recurse> NO
CaseIgn> NO
-----Name----- T ---SzL--- ----Modified----- Permission --Path--- -Owner
---.nfsc----- f 8 2005/06/03 14:07:45 rw-r--r-- /ADCD/etc START2
---booksrv----- d 8192 1999/05/12 17:43:37 rwxr-xr-x /ADCD/etc START2
---bpa----- d 8192 1999/01/19 16:08:04 rwxr-xr-x /ADCD/etc 2134
---cmx----- d 8192 1999/01/19 16:08:04 rwxr-xr-x /ADCD/etc 2134
---csh.login.nbj f 1119 2008/06/23 15:21:41 rwxrwxr-x /ADCD/etc NBJ
---dce----- d 8192 1999/08/24 14:47:53 rwxr-xr-x /ADCD/etc 2134
---dfs----- d 8192 1999/08/24 14:47:53 rwxr-xr-x /ADCD/etc 2134
---hostsx----- f 34 2005/05/12 23:16:38 rwxrwxrwx /ADCD/etc START2
---httpd.conf----- f 127910 2000/05/03 14:09:04 rwxr-xr-x /ADCD/etc START2
---httpd.envvars f 536 2000/05/03 14:06:22 rw-r--r-- /ADCD/etc START2
---ics_pics.conf f 3132 2000/05/03 14:09:17 rw-r--r-- /ADCD/etc START2
---imolsinf----- f 330 1999/08/13 13:51:25 rwxr-xr-x /ADCD/etc START2
---inetd.conf----- f 1505 2008/06/17 15:04:52 ----- /ADCD/etc START2
---inetd.pid----- f 10 2009/04/27 09:26:12 rw-r--r-- /ADCD/etc START2
---init.options f 2587 1999/10/21 18:52:50 ----- /ADCD/etc START2
---ioepdcf----- l 22 1999/10/23 22:43:24 rwxrwxrwx /ADCD/etc START2
---javelin.conf f 13573 2000/05/03 14:09:29 rw-r--r-- /ADCD/etc START2
Line 1 of 39 | Col 1 of 601 | Views 1 | select * sort Name,T

```

Figure 32. 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: <ul style="list-style-type: none"> • For a directory entry or a symbolic link to a directory, open a new List HFS list window to display the contents of the directory. • For all other entries, a CBL text editor view is opened to edit the data. (Equivalent to prefix command "E").
B	Open the CBL text editor 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 CBL 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 ultimately generate specific line commands in a temporary CMX file.
V	Open the CBL 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 PF2 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
AcIAccess	BitFlag	Access ACL exists
AcIFModel	BitFlag	File Model ACL exists
AcIDModel	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

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.

```

SELCOPY/i - File Copy
File Help JCL Command
Command>
ZZSGFC00
Input PDS/PDSE Library, Sequential, VSAM DSN mask or HFS path mask:
  DSN/Path Mask> _____ + (All mask matches will be copied, type
  Member Mask> _____ + SELECT to view/modify the selection list)
  Volume Mask> _____

Output PDS/PDSE Library, Sequential, VSAM DSN or HFS path:
  DSN/Path > _____ +
  Member > _____ - Replace existing members
  Volume > _____ (If output file is uncataloged)
  Strip/Pad Char> _____ e.g. X'FF' (If copying fixed<->var length records)

Record Selection: For each input file, copy only selected records.
- Start> _____ + / Record - Key - RBA
- For> _____ # records
- Filter> Q Select records to copy. (F=File; Q=Quick) Type FILT to view/mod
  File> _____ + Member> _____

Options:
  / Reformat using structure/copybook layouts - Recurse HFS Sub-directories
  - Append to existing Output - Ignore HFS fileid case
  Member Delim> @
  
```

Figure 33. 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 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 constitute 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 belonging to that library. <PF5> 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 utility. 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.

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 entries 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 "," (comma) 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.

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 libraries, then the output library will contain the last member copied.

Strip/Pad Char>

Specifies a single byte pad or strip character implemented 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 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 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.

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) <<<<<<*"

Reformat using structure/copybook layouts

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 **Reformat using structure/copybook layouts** 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.

```

SELCOPY/i - File Reformat
File Help JCL Command
Command>
ZZSGFC00
Input Structure/Copybook overlay:
Dsn> Type: / SDO _ AData _ COBOL _ PL1 Member>
PF10=Browse Input Dataset
PF22=Browse Input Copybook

Output Structure/Copybook overlay:
Dsn> Type: / SDO _ AData _ COBOL _ PL1 Member>
PF11=Browse Output Dataset
PF23=Browse Output Copybook

1. Help (PF1) 2. Execute (ENTER) 3. Back (PF3) 4. Exit (PF15)
  
```

Figure 34. 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> Y/N

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.

PFKeys

In addition to the standard interactive panel key assignments for scrolling and navigation, the File Copy Search 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.
F6	FILTER	Open the FILTER dialog for the specified filter type.
F13	REMIND	Display a saved input field value.
F14	EXPAND	Expand an input/output field.
F19	BIF	Browse the input file if input fileid mask is a single file.
F20	BOF	Browse the output file.

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).

```

FSU - PDS Copy Statistics
View Refresh Back Forward FDB Text Help
Command>                               Scroll> Csr

-Member- -Action- AliasOf- Truncated RemapError
AA       Replaced
AAA     *NoAlias ABC
SDETSO  Copied           Y
SDPROF  Copied
SELECT  Copied
SLCCXX  Replaced           Y
UNCAT   Copied           Y
WHAT    Copied           Y
ZZSICBLE Replaced
ZZSIEDIT Replaced
ZZSLCXX *NoCopy  SLCCXX  Y

Line 1 of 11 | Col 1 of 47 | Views 1 | select * sort Member

```

Figure 35. 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.

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.

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.
- 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.
- 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 satisfy 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 discriminate 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.

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.

```

SELCOPY/i - Basic File Search
File Help JCL Command
Command>
ZZSGFSU0
PDS/PDSE Library, Sequential, VSAM DSN mask or HFS path mask:
DSN/Path Mask>
Member Mask>
Volume Mask>
HFS Options>
Search Options:
Op> EQ
String>
Bounds> 1 (Start Column) 0 (End Column) 0=>Start Column only.
As> / Unrestricted Word Prefix Suffix
Record Selection:
Start>
For> 0 # records
Filter> Q Filter selected records. (F=File; Q=Quick)
File>
Extended File Search/Update/Copy/Remap Tasks:
Enter "/" to display a list of extended FSU utility tasks.
  
```

Figure 36. 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 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 constitute 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> 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 belonging to that library.

Having displayed the Select Files to Process panel, a list of the selected, individual fileids will be passed to the utility 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 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 entries 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 "," (comma) 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.

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 constitute 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 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.

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.

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*.

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 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.

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.

PFKeys

In addition to the standard interactive panel key assignments for scrolling and navigation, the Basic File Search panel supports the following:

PF1	HELP	Display context sensitive help .
PF2	EXPAND	Expand an input/output field.
PF3	END	Close the panel.
F16	REMIND	Display a saved input field value.
PF5	SELFILES	Select from a list of files that match the Fileid mask.
PF6	FILTER	Open the FILTER dialog for the specified filter type.
PF17	COMMAND	Display the generated FSU command.

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.

```

SELCOPY/i - Extended File Search, Update, Copy & Remap Tasks
File Help
Command>
ZZSGFSU0

Unformatted Data:
1. File Search (FIND) Search/Report records with FIND.
2. File Search (WHERE) Search/Report records with SDE expression.
3. File Change+Update Change record data then Update-in-place.
4. File Copy Copy records to a File.
5. File Change+Copy Change & Copy records to a File.
6. Library Copy Copy Library member records to a Library.
7. Library Change+Copy Change & Copy Library member records to a Library.

Formatted Data: (Using a Copybook/Structure)
8. File Search (FIND) Search/Report formatted records with FIND.
9. File Search (WHERE) Search/Report formatted records with SDE expression.
10. File Change+Update Change formatted records then Update-in-place.
11. File Change+Copy Change formatted records then Copy to a File.
12. File Remap Remap formatted records to a File.
13. File Change+Remap Change & Remap formatted records to a File.
14. Library Change+Copy Change & Copy formatted member records to a Library.
15. Library Remap Remap formatted member records to a Library.
16. Library Change+Remap Change & Remap formatted Library member records.

```

Figure 37. 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 applies 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 applies 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 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**

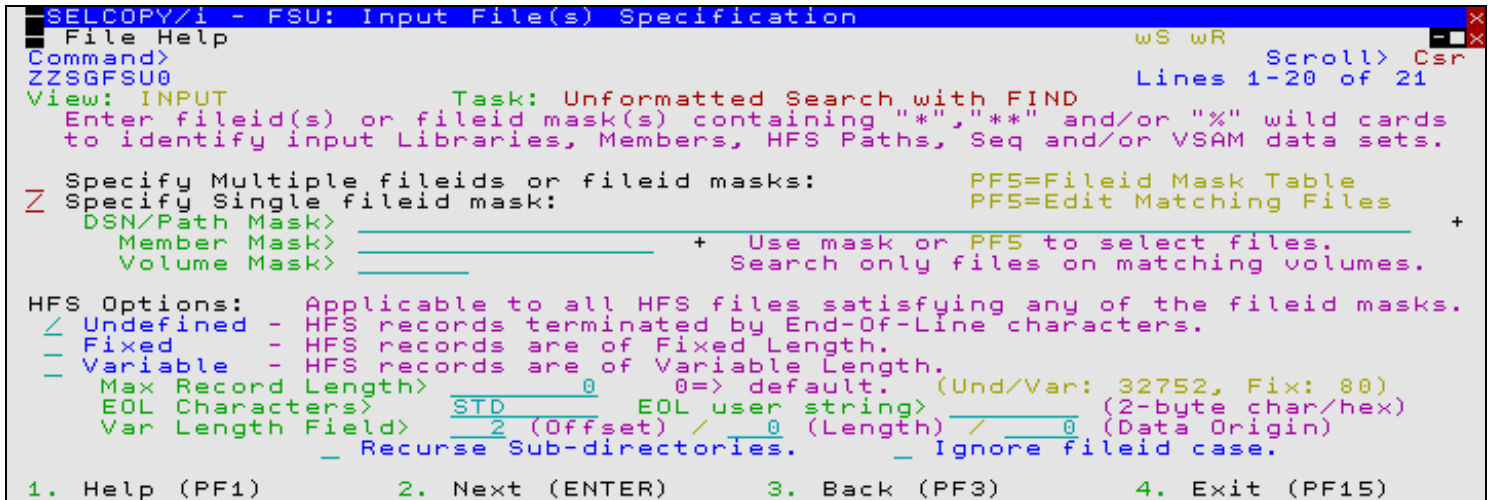
This option provides the same functionality as **Formatted Library Remap** with the added ability to change data in fields as it is remapped to the output members.

Specification of a record-type mapping is mandatory to identify 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.



```

SELCOPY/i - FSU: Input File(s) Specification
File Help
Command>
ZZSGFSU0
View: INPUT
Task: Unformatted Search with FIND
Enter fileid(s) or fileid mask(s) containing "*", "**" and/or "%" wild cards
to identify input Libraries, Members, HFS Paths, Seq and/or VSAM data sets.

Specify Multiple fileids or fileid masks:
Specify Single fileid mask:
DSN/Path Mask>
Member Mask>
Volume Mask>
PFS=Fileid Mask Table
PFS=Edit Matching Files
+
Use mask or PFS to select files.
Search only files on matching volumes.

HFS Options:
/ Undefined - HFS records terminated by End-Of-Line characters.
- Fixed - HFS records are of Fixed Length.
- Variable - HFS records are of Variable Length.
Max Record Length> 0 0=> default. (Und/Var: 32752, Fix: 80)
EOL Characters> STD EOL user string> (2-byte char/hex)
Var Length Field> 2 (Offset) / 0 (Length) / 0 (Data Origin)
- Recurse Sub-directories.
- Ignore fileid case.

1. Help (PF1) 2. Next (ENTER) 3. Back (PF3) 4. Exit (PF15)

```

Figure 38. 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 and library members 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, <PF5> or <Enter> will display the **Multiple Fileid Masks** panel. When a list of fileid masks has been entered and closed, <PF5>, 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.

Press <PF5> 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 be opened to select from a list of matching member names 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

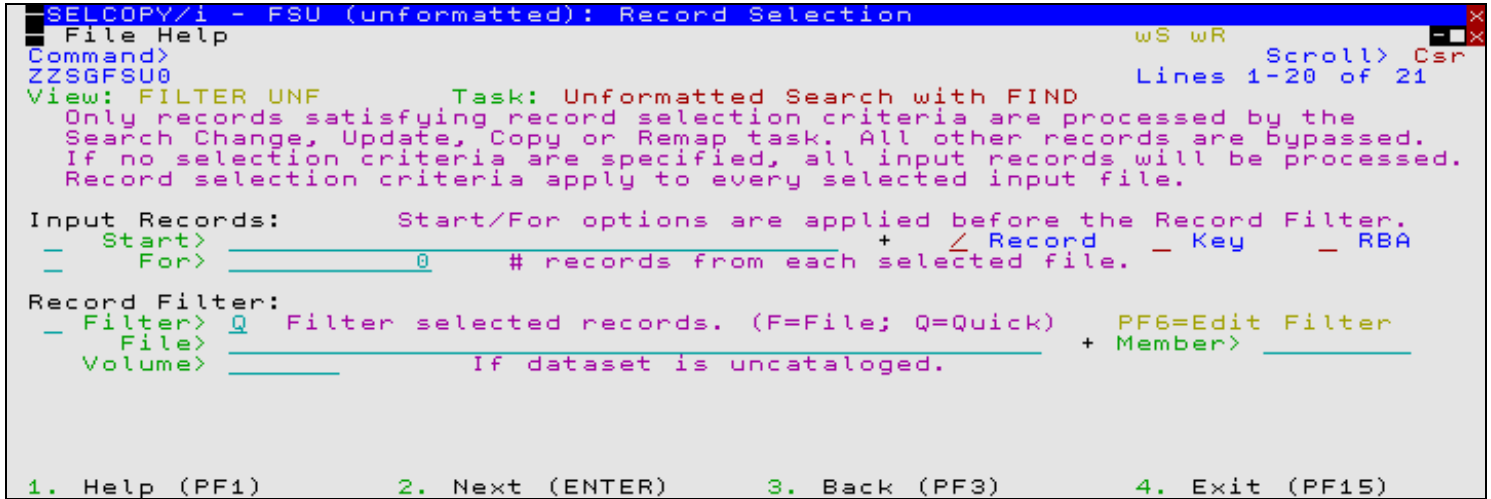
Set this option on to disable case sensitivity when matching HFS file names with the file **name** portion of the specified HFS file path mask. The name portion of the HFS file path is the character string following the last "/" (slash) of the fileid mask or the entire HFS file path if no "/" is included 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.



```

SELCOPY/i - FSU (unformatted): Record Selection
File Help
Command>
ZZSGFSU0
View: FILTER UNF Task: Unformatted Search with FIND
Only records satisfying record selection criteria are processed by the
Search Change, Update, Copy or Remap task. All other records are bypassed.
If no selection criteria are specified, all input records will be processed.
Record selection criteria apply to every selected input file.

Input Records: Start/For options are applied before the Record Filter.
- Start> _____ + / Record - Key - RBA
- For> _____ # records from each selected file.

Record Filter:
- Filter> Q Filter selected records. (F=File; Q=Quick) PF6=Edit Filter
- File> _____ + Member> _____
- Volume> _____ If dataset is uncataloged.

1. Help (PF1) 2. Next (ENTER) 3. Back (PF3) 4. Exit (PF15)

```

Figure 39. 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:

Fields which together specify the master subset of selected records on which subsequent filtering and processing will occur. These field values are applied 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.

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.

```

SELCOPY/i - FSU (formatted): Record Selection
File Help                                     wS wR
Command>                                     Scroll> Csr
ZZSGFSU0                                     Lines 1-20 of 21
View: FILTER FMT                             Task: Formatted Search with FIND
Only records satisfying record selection criteria are processed, all others
are bypassed. If no selection criteria are specified, all input records are
processed. Record selection criteria apply to every selected input file.

Input Records:                               Start/For options are applied before the Record Filter.
- Start>                                     + / Record - Key - RBA
- For> _____ # records from each selected file.

Input Structure/Copybook File & Record-Type Mapping:
  Dsn> _____ Member> _____
  Type> / SDO - AData - COBOL - PL1 Copybook Source Type
  RecType> _____ + (Used for WHERE, FIND & CHANGE only)

Record Filter:
- Filter> Q Filter selected records. (F=File; Q=Quick) PF6=Edit Filter
  File> _____ + Member> _____
  Volume> _____ If dataset is uncataloged.

1. Help (PF1)      2. Next (ENTER)      3. Back (PF3)      4. Exit (PF15)

```

Figure 40. 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:

Fields which together specify the master subset of selected records on which subsequent filtering and processing will occur. These field values are applied 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.

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> Y/N

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.



```

SELCOPY/i - FSU (unformatted): Search records using WHERE expression
File Help                               wS wR
Command>                                Scroll> Csr
ZZSGFSU0                                Lines 1-20 of 21
View: WHERE                               Task: Unformatted Search with expression
Search and report records that match the search criteria specified by an SDE
WHERE expression. If Execute FIND Command(s) is selected, only those records
which satisfy the WHERE expression are passed for FIND command processing.

For formatted record processing, the WHERE expression may reference field
names/numbers identified by the selected record-type.

Z Use generated WHERE expression (unformatted data):          PF5=Generate WHERE
Z Use WHERE expression entered below:
  WHERE> _____ +

< Execute additional FIND on records satisfying this WHERE expression.

1. Help (PF1)          2. Next (ENTER)          3. Back (PF3)          4. Exit (PF15)

```

Figure 41. 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, <PF5> or <Enter> will display the **Unformatted Selection Criteria** panel. When a WHERE expression has been entered and the panel closed, <PF5>, 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 no validation is performed on this field.

<PF2> may be pressed to expand an 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.

```

SELCOPY/i - FSU (formatted): Search records using WHERE expression
File Help                               wS wR
Command>                                Scroll> Csr
ZZSGFSU0                                Lines 1-20 of 21
View: WHERE                               Task: Formatted Search with expression
Search and report records that match the search criteria specified by an SDE
WHERE expression. If Execute FIND Command(s) is selected, only those records
which satisfy the WHERE expression are passed for FIND command processing.

For formatted record processing, the WHERE expression may reference field
names/numbers identified by the selected record-type.

/ Use generated WHERE expression (formatted data):           PF5=Generate WHERE
- Use WHERE expression entered below:
  WHERE> _____ +

_ Execute additional FIND on records satisfying this WHERE expression.

1. Help (PF1)      2. Next (ENTER)      3. Back (PF3)      4. Exit (PF15)

```

Figure 42. 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, <PF5> or <Enter> will display the **Formatted Record Expression** panel. When a WHERE expression has been entered and the panel closed, <PF5>, 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 no validation is performed on this field.

<PF2> may be pressed to expand an 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.



```

SELCOPY/i - FSU (unformatted): Search Records using the FIND Command
File Help                               WS WR
Command>                               Scroll> Csr
ZZSGFSU0                                Lines 1-20 of 21
View: FIND UNF                          Task: Unformatted Search with FIND
Enter one or more FIND commands used to search, then report matching input
records. All specified FIND commands are joined by a logical AND or OR.

Specify Multiple FIND commands:          PF5=FIND Command Table
Specify Single FIND command:
Op> EQ Relational operator. (Enter "/" for list)
String>
Bounds> 1 (Start Column) 0 (End Column) 0=>Start Column only.
As> / Unrestricted _ Word _ Prefix _ Suffix

1. Help (PF1)      2. Next (ENTER)      3. Back (PF3)      4. Exit (PF15)

```

Figure 43. 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, <PF5> or <Enter> will display the **Multiple Find Commands** panel. When a list of FIND commands has been entered and closed, <PF5>, may be used to redisplay the panel.

Specify Single FIND command:

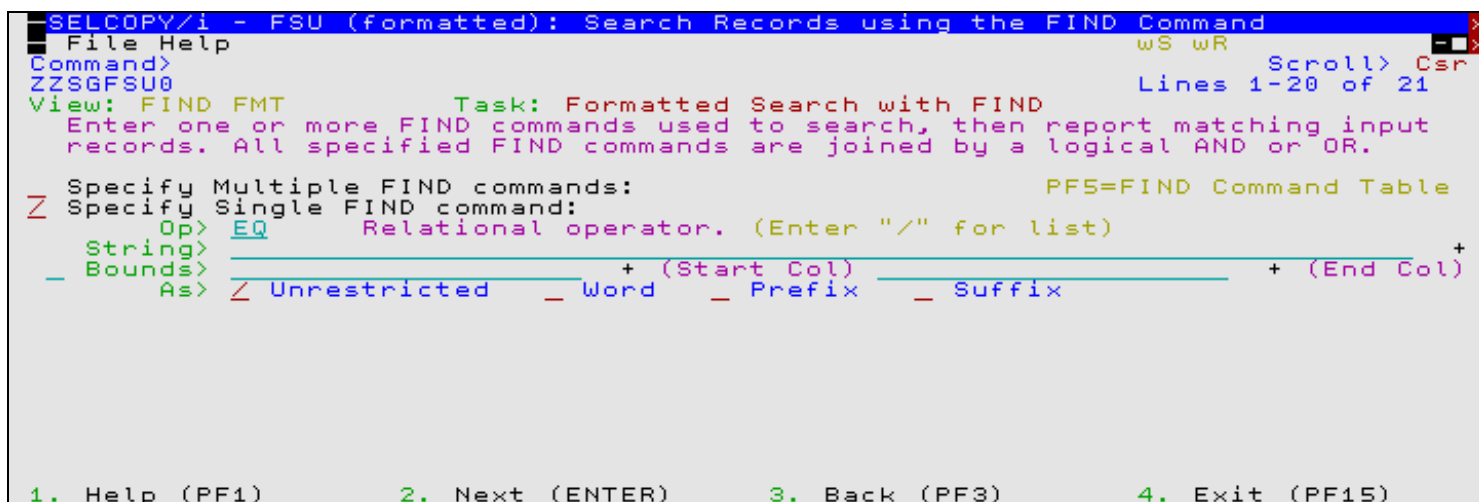
Select this option if only one FIND command is required. The fields that follow together constitute an edit **FIND** command used to search unformatted input file records.

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.



```

SELCOPY/i - FSU (formatted): Search Records using the FIND Command
File Help                               wS wR
Command>                               Scroll> Csr
ZZSGFSU0                               Lines 1-20 of 21
View: FIND FMT                          Task: Formatted Search with FIND
Enter one or more FIND commands used to search, then report matching input
records. All specified FIND commands are joined by a logical AND or OR.

Specify Multiple FIND commands:          PF5=FIND Command Table
Specify Single FIND command:
  Op> EQ                                Relational operator. (Enter "/" for list)
String>
Bounds>
As> / Unrestricted _ Word _ Prefix _ Suffix

1. Help (PF1)      2. Next (ENTER)      3. Back (PF3)      4. Exit (PF15)

```

Figure 44. 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, <PF5> or <Enter> will display the **Multiple Find Commands** panel. When a list of FIND commands has been entered and closed, <PF5>, 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 constitute 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.

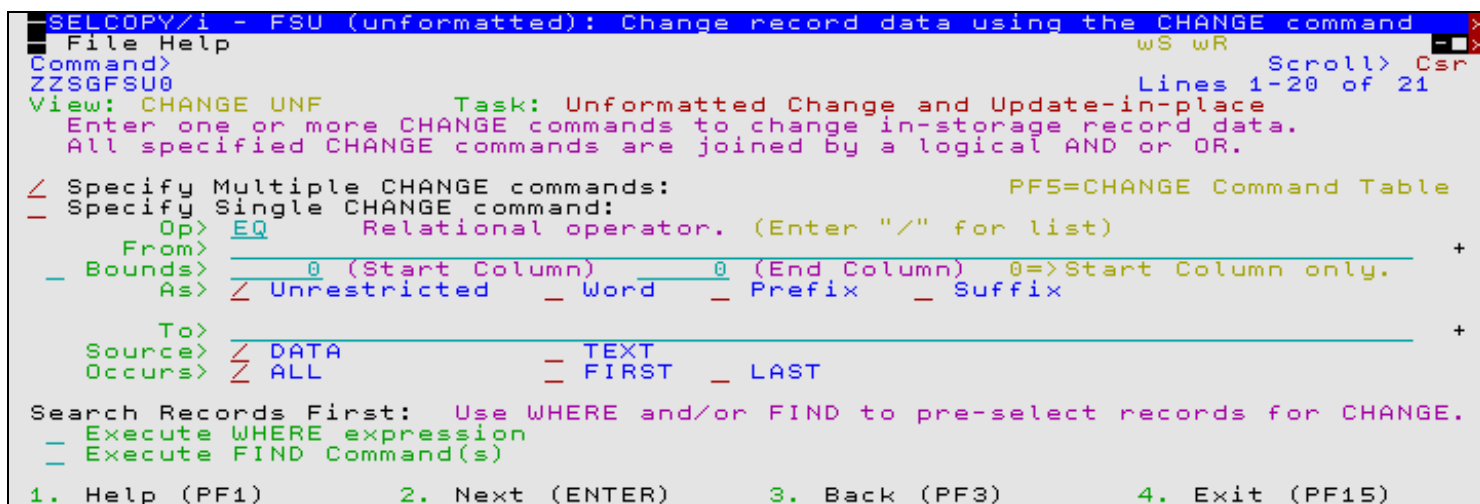


Figure 45. FSU (unformatted): Change record data using the CHANGE command.

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 be applied to the record data.

If this option is selected, <PF5> or <Enter> will display the **Multiple Change Commands** panel. When a list of CHANGE commands has been entered and closed, <PF5>, 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 constitute 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 appropriate.

If the length of the replace value is greater than that of the search value, consecutive blanks 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.

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 selected 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.

```

SELCOPY/i - FSU (formatted): Change record data using the CHANGE command
File Help                               wS wR  Scroll> Csr
Command>
ZZSGFSU0                                Lines 1-20 of 21
View: CHANGE FMT      Task: Formatted Change and Update-in-place
Enter one or more CHANGE commands to change in-storage record data.
All specified CHANGE commands are joined by a logical AND or OR.

/ Specify Multiple CHANGE commands:                PF5=CHANGE Command Table
- Specify Single CHANGE command:
  Op> EQ      Relational operator. (Enter "/" for list)
  From> _____ + (Start Col) _____ + (End Col) +
  - Bounds> _____ + (Start Col) _____ + (End Col)
  As> / Unrestricted - Word - Prefix - Suffix

  To> _____ +
  Source> / DATA - TEXT
  Occurs> / ALL - FIRST - LAST

Search Records First: Use WHERE and/or FIND to pre-select records for CHANGE.
- Execute WHERE expression
- Execute FIND Command(s)

1. Help (PF1)      2. Next (ENTER)      3. Back (PF3)      4. Exit (PF15)

```

Figure 46. 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 be applied to the record data.

If this option is selected, <PF5> or <Enter> will display the **Multiple Change Commands** panel. When a list of CHANGE commands has been entered and closed, <PF5>, 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 constitute 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 *field_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 appropriate.

If the length of the replace value is greater than that of the search value, consecutive blanks 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 selected 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.

```

SELCOPY/i - FSU: Copy Selected records to an Output File
File Help
Command>
ZZSGFSU0
View: OUTPUT-FILE UNF Task: Copy to a File
Specify the output file to which selected records from all input files
will be copied. The Allocate Non-VSAM dialog panel will open if a
data set name is specified for a file that does not already exist.

PDS/PDSE member, Sequential, VSAM or HFS path:
  Name> _____ + Member> _____
  Volume> _____ If dataset is uncataloged.
  Strip/Pad Char> _____ (e.g. X'FF') If copying fixed->var length records.

Output Options:
  _ Append output records to existing file data.

1. Help (PF1)      2. Next (ENTER)    3. Back (PF3)     4. Exit (PF15)

```

Figure 47. 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.

Strip/Pad Char>

Specifies a single byte pad or strip character implemented 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 matching field name belonging to record-type definitions of matching record-type name within the input and output structures can be remapped.

A record-type field definition in the input structure that corresponds to an equally named 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.

```

SELCOPY/i - FSU: Remap Fields in Selected records to an Output File
File Help                               wS wR                               Scroll> Csr
Command>                                Lines 1-20 of 21
ZZSGFSU0
View: OUTPUT-FILE FMT  Task: Formatted Field Remap to a File
Specify output file and structure used to remap records selected for copy.
The Allocate Non-VSAM dialog will open if the output file is new. Data is
copied only between fields and record type definitions of the same name.
Matching fields may be of different data format and record offset.

PDS/PDSE member, Sequential, VSAM or HFS path:
  Name> _____ + Member> _____
  Volume> _____ If dataset is uncataloged.

Output Structure/Copybook File for Record Remap:
  Dsn> _____ Member> _____
  Type> / SDO _ AData _ COBOL _ PL1 Copybook Source Type
  Strip/Pad Char> _____ (e.g. X'FF') If copying fixed->var length records.

Output Options:
  _ Append output records to existing file data.

1. Help (PF1)      2. Next (ENTER)      3. Back (PF3)      4. Exit (PF15)

```

Figure 48. 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.

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> Y/N

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 implemented 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.

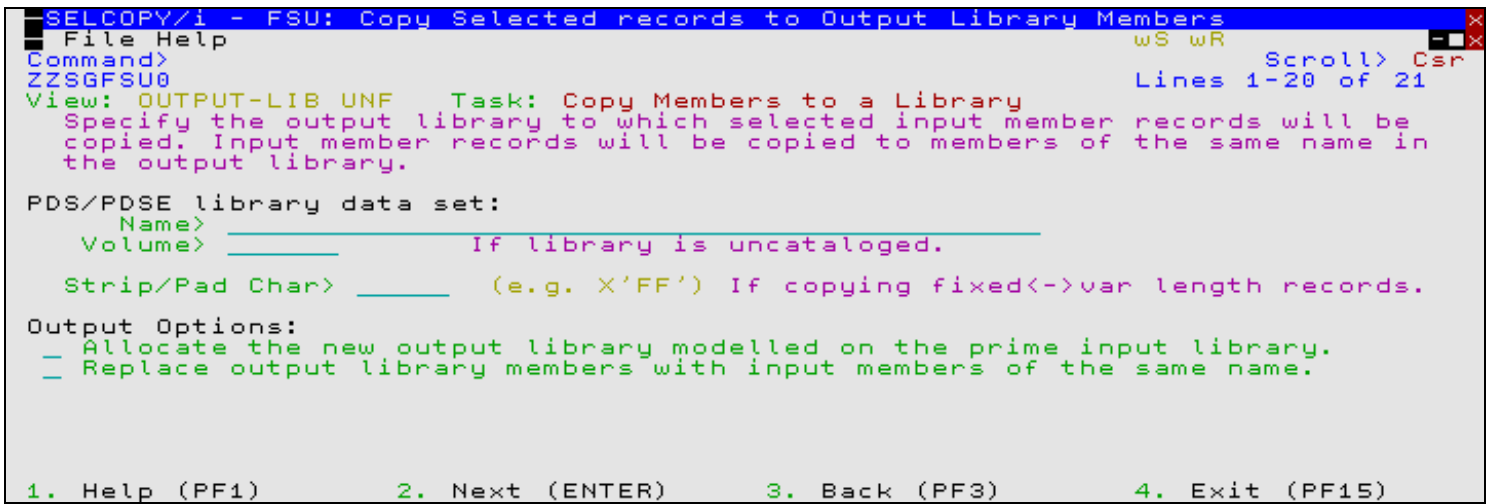
Applicable only when copying members from a PDS/PDSE library to the output file. This option causes a delimiter record to be written before the data belonging to each member copied.

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.



```

SELCOPY/i - FSU: Copy Selected records to Output Library Members
File Help                               wS wR
Command>                               Scroll> Csr
ZZSGFSU0                               Lines 1-20 of 21
View: OUTPUT-LIB UNF   Task: Copy Members to a Library
Specify the output library to which selected input member records will be
copied. Input member records will be copied to members of the same name in
the output library.

PDS/PDSE library data set:
  Name> _____
  Volume> _____ If library is uncataloged.
  Strip/Pad Char> _____ (e.g. X'FF') If copying fixed->var length records.

Output Options:
  _ Allocate the new output library modelled on the prime input library.
  _ Replace output library members with input members of the same name.

1. Help (PF1)       2. Next (ENTER)       3. Back (PF3)       4. Exit (PF15)

```

Figure 49. 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 implemented as follows:

- 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.
- 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.

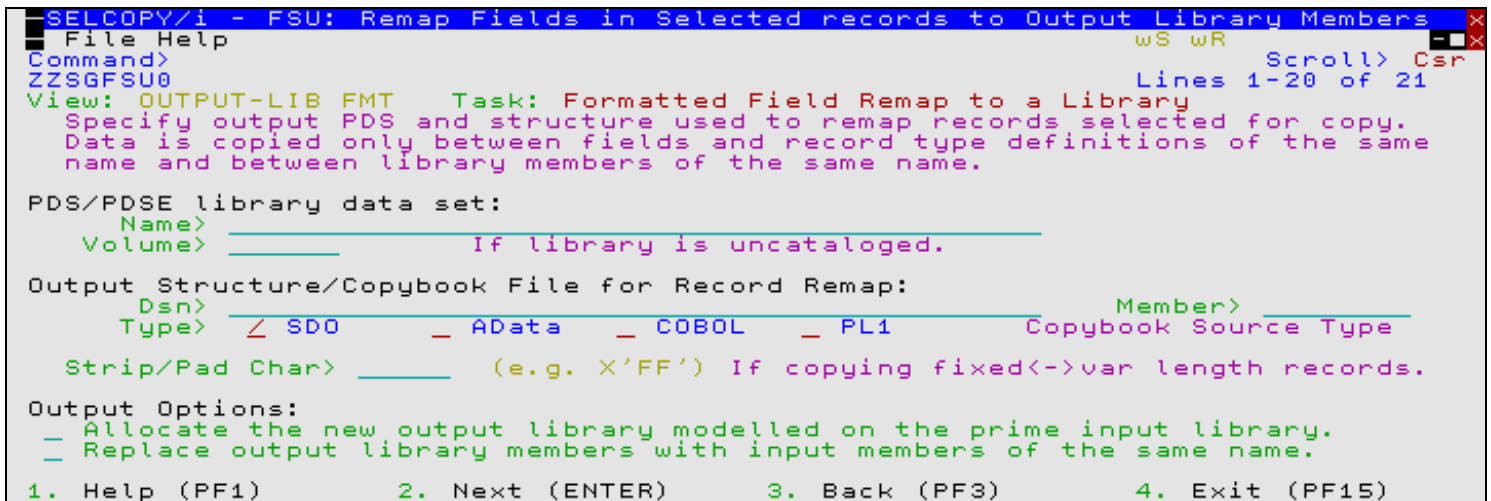
If an existing member in the output library has the same name as a member to be copied from an input library, it will be overwritten. In the case where members of the same name are copied from multiple input libraries, then the output library will contain the member copied last.

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 matching field name belonging to record-type definitions of matching record-type name within the input and output structures can be remapped.

A record-type field definition in the input structure that corresponds to an equally named 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.



```

SELCOPY/i - FSU: Remap Fields in Selected records to Output Library Members
File Help                               wS wR
Command>                                Scroll> Csr
ZZSGFSU0                                Lines 1-20 of 21
View: OUTPUT-LIB FMT   Task: Formatted Field Remap to a Library
Specify output PDS and structure used to remap records selected for copy.
Data is copied only between fields and record type definitions of the same
name and between library members of the same name.

PDS/PDSE library data set:
  Name> _____
  Volume> _____ If library is uncataloged.

Output Structure/Copybook File for Record Remap:
  Dsn> _____ Member> _____
  Type> / SDO   _ AData   _ COBOL   _ PL1   Copybook Source Type
  Strip/Pad Char> _____ (e.g. X'FF') If copying fixed->var length records.

Output Options:
  _ Allocate the new output library modelled on the prime input library.
  _ Replace output library members with input members of the same name.

1. Help (PF1)      2. Next (ENTER)      3. Back (PF3)      4. Exit (PF15)
  
```

Figure 50. 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> Y/N

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 implemented 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 libraries, 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.

```

SELCOPY/i - FSU: Report File
File Help JCL Command
Command>
ZZSGFSU0
View: REPORT Task: Unformatted Search with FIND
Generate an output report for the search, update, copy or remap operation.
Output is to a temporary file or a new or existing sequential data set or
library member. For non-temporary report output, the file DSN must be no
longer than 40 characters as a structure of the same DSN but with an
additional low level qualifier of "SDO" will automatically be generated.
This structure is used to display the formatted report lines.

/ Generate temporary report output.
- Write report output to library member or sequential data set:
  Name> _____ + Member> _____
  Volume> _____ If dataset is uncataloged.

- Suppress generation of report output.

Report Options:
- Suppress display of the output report following execution.

1. Help (PF1)      2. Execute (ENTER)  3. Back (PF3)     4. Exit (PF15)

```

Figure 51. 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 eligible 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.

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.

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 library members.

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 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 utility instead of the generic fileid mask.

On completing entries in this panel view, press <PF3> 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 accommodate 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, place the cursor on the required table entry and press <PF2>.

```

SELCOPY/i - Multiple Fileid Masks
File Edit Actions Options Utilities Window SwapList Help  wS wR
Command>
ZZSGFSU3
Enter Fileids and/or Fileid Masks.
Use wildcard chars "*", "**" and/or "%" in masks.
Enter Fileids/Fileid Masks.
Sel DSN/HFS Mask Member Mask Volume Mask
000000 *** <...+...1...+...2...+...3...+...4...> <...+...1.> <...+>
000001 S NBJ.**.JCL SS*
000002 S NBJ.**.SYSUDUMP XV*
000003 S NBJ.TEMP.D120112.T01%***% CVOL*
000004 S /u/cbl/nbj/temp.**
000005 S NBJ.XFILE.**
000006 *** End of Data ***

```

Figure 52. Multiple Fileid Masks.

Single Row (Zoomed) View

This panel displays the table column entries for an individual table row as panel input fields. Input fields may be expanded, by placing the cursor in the required field and pressing <PF2>, in order to enter values that exceed the width of the input field area.

Press <PF5> to display the **Select Files to Process** panel from this panel.
Press <PF3> to return to the table view.

```

SELCOPY/i - Enter Fileid Mask
File Edit Actions Options Utilities Window SwapList Help  wS wR
Command>
ZZSGFSU3
Specify a PDS/PDSE Library, Sequential, VSAM DSN mask or HFS path mask.
Select this File Mask for Processing> /
DSN/Path Mask> NBJ.**.JCL
Member Mask> SS*
Volume Mask>

```

Figure 53. Enter Fileid Mask (Zoomed View).

Input Values

Fields which together constitute a fileid mask that identifies one or more files to be processed by the utility.

se1> 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 entries 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 "," (comma) 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.

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 utility instead of the generic library member mask.

On completing entries in this panel view, press <PF3> 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**. Scroll left/right (using <PF10> and <PF11> respectively) to view 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 an 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 mask(s).

If the input member mask column area is not large enough to accommodate the required entry value, display the table row in zoomed format and expand the field entry as required. To display a table row in single row (zoomed) format, place the cursor on the required table entry and press <PF2>.

Press <PF6> to select or deselect all table entries.

```

SELCPY/i - Select Files to Process
File Edit Actions Options Utilities Window SwapList Help  wS wR
Command>
ZZSGFSU1
Select files to be included in the operation.          PF6=Select/Deselect ALL
File Mask> NBJ.CBLI.**
To (de)select members place cursor on each lib name and press ENTER.  35 Rows
Sel File Name      Member      Volume Org VSAM  GDG  RecFm
Mask              Mask
. <...+...1...+...2...+...3..> <...+...> <...+> <.> <..> . <...>
*** Top of Data ***
NBJ.CBLI.CBLE      *          PO      N      VB      000000
NBJ.CBLI.CBLE.FSU  *          PO      N      VB      000001
NBJ.CBLI.CBLX.FSU  *          PO      N      VB      000002
NBJ.CBLI.CMX       *          PS      N      VB      000003
NBJ.CBLI.DATSALES.SEQ.BIG  *          PS      N      VB      000004
NBJ.CBLI.DIST.CMX  *          PO      N      VB      000005
NBJ.CBLI.DIST.LST  *          PO      N      FBA     000006
NBJ.CBLI.FILTER    *          PO      N      FB       000007
NBJ.CBLI.INI       *          PS      N      VB      000008
NBJ.CBLI.INI.PROFIRST  *          PS      N      VB      000009
NBJ.CBLI.INS.CMX   *          PS      N      VB      000010
NBJ.CBLI.IPO       *          PO      N      VB      000011

```

Figure 54. 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. Input fields may be expanded, by placing the cursor in the required field and pressing <PF2>, in order to enter values that exceed the width of the input field area.

Press <PF5> to display the **Select Input Members** panel from this panel.

Press <PF3> to return to the table view.

```

SELCPY/i - Select Files to Process
File Edit Actions Options Utilities Window SwapList Help  wS wR
Command>
ZZSGFSU1
Select File for Processing> /
File Name: NBJ.CBLI.CBLE.FSU
Volume:
Member Mask: *
Organisation: PO      VSAM Type:      GDG Base: N
RECFM: VB      LRECL:      255      BLKSIZE: 32760

```

Figure 55. 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 Input 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.

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, press <PF3> 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**, **Alias**, **Created**, **LastMod**, **Cursize**, **Inisize**, **TTR**, **User** and **AliasOf**. Scroll left/right (using <PF10> and <PF11> respectively) to view columns that are not in view.

Only column **Sel** is enterable. All other column entries are included for information only.

Press <PF6> to select or deselect all table entries.


```

SELCOPY/i - Select Input Members
File Edit Actions Options Utilities Window SwapList Help  wS wR  Scroll> Csr
Command>
ZZSGFSU2
Select members to be included in the operation.          PF6=Select/Deselect ALL
Member Mask> *
Select Members to process.
Sel Member      Alias Created      LastMod      Cursize Inisize TTR      Us      66 Rows
. <...+...> . <...+...> <...+...1...+> <...+> <...+> <...+> <.
*** Top of Data ***
ABBEY      N      2006/09/14  2007/12/07  15:15      18      16      00001E  J  000000
ABBEYC     N      2007/02/16  2007/02/16  09:47      17      17      00001A  J  000002
ABBEYX     N      2006/09/14  2007/12/07  15:15      18      16      000029  J  000003
ACB        N      2008/09/29  2009/07/14  11:11      18      18      000026  J  000004
ADACOPYT   N      2004/04/26  2009/07/14  11:11      51      18      000010  J  000005
ADACOPYV   N      2003/08/21  2009/07/14  11:11      21      16      000009  J  000006
AIXCOPY    N      2009/08/12  2009/08/19  16:45      152     101     00002F  J  000007
AMAINLK    N      2010/02/01  2010/12/10  13:11      28      22      000033  N  000008
AMBLIST    N      2005/11/11  2009/07/14  11:11      8        8      000016  J  000009
AMEBCDIC   N      2008/09/29  2009/07/14  11:11      20      20      000027  J  000010
AMEX       N      2009/10/05  2009/10/05  16:31      7        7      000030  J  000011
AMSDEL     N      2010/09/23  2010/09/23  15:11      81      82      000032  J  000012
AMSUPPM    N      2008/01/09  2009/07/14  11:11      24      20      000023  J  000013

```

Figure 56. Select Input Members.

Single Row (Zoomed) View

This panel view displays the table column entries for an individual table row as panel input fields.

Press <PF3> to return to the table view.

```

SELCOPY/i - Select Input Members
File Edit Actions Options Utilities Window SwapList Help  wS wR  Scroll> Csr
Command>
ZZSGFSU2
Lines 1-20 of 24
PF3=Table View

Select Member for Processing> /

Member Name: AMBLIST
Alias: N
Date Created: 2005/11/11
Date Modified: 2009/07/14 11:11
Initial Size: 8
Current Size: 8
TTR: 000016
User Id: JGE
Alias Of:

```

Figure 57. 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.

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 and 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 58. Find/Condense Input Members.

Press <PF3> 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.

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, press <PF3> 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, place the cursor on the required table entry and press <PF2>.

```

SELCOPY/i - Major OR Sub-Expressions
File Edit Actions Options Utilities Window SwapList Help  wS wR
Command>
ZZSGEXP0
Add one or more alternate (logically ORed) expressions to the table below.
Press <ENTER> on each OR expression to input one or more mandatory
(logically ANDed) sub-expressions.

Structure:  CBL.SDE.COPYBOOK.COBOL          Member:  GENAMEMP
Record-Type: AM                               Type:    Cob

Enter Alternate (Major OR) Expressions.                                     3 Rows
  OR NOT AND
  Sub-Expressions
  <> . <...+...1...+>
000000 *** Top of Data ***
000001 OR - > 2 specified
000002 OR - > 1 specified
000003 OR - > 4 specified
000004 *** End of Data ***

```

Figure 59. Major OR Sub-Expressions.

Single Row (Zoomed) View

This panel displays the table column entries for an individual table row as panel input fields.

Press <PF5> to display the **AND Sub-Expressions** panel from this panel view.

Press <PF3> to return to the table view.

```

SELCOPY/i - Enter Major OR Sub-Expression
File Edit Actions Options Utilities Window SwapList Help  wS wR
Command>
ZZSGEXP0
Enter a logical OR expression which consists of one or more logically
ANDed sub-expressions.

Structure:  CBL.SDE.COPYBOOK.COBOL          Member:  GENAMEMP
Record-Type: AM                               Type:    Cob

Enter a logical OR expression which consists of one or more logically
ANDed sub-expressions.

NOT Logical Operator:
_ Negate Boolean condition set by this OR expression.

OR Sub-Expressions:
  Press <PF5> or click here to input AND sub-expressions.

```

Figure 60. 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), P11 (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 separated 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, press <PF3> 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, place the cursor on the required table entry and press <PF2>.

```

SELCPY/i - AND Sub-Expressions
File Edit Actions Options Utilities Window SwapList Help  wS wR  Scroll> Csr
Command>
ZZSGEXP1
Add one or more mandatory (logically ANDed) expressions to the table below.
Press <ENTER> on each AND expression to input one or more alternate
(logically ORed) sub-expressions.

Structure:      CBL.SDE.COPYBOOK.COBOL          Member:  GENAMEMP
Record-Type:   AM                               Type:    Cob

Enter AND Expressions.                                2 Rows
  AND NOT Minor OR
        Sub-Expressions
<.> .      <...+...1...+>
000000 *** Top of Data ***
000001 AND - > 3 specified
000002 AND - > 2 specified
000003 *** End of Data ***

```

Figure 61. AND Sub-Expressions.

Single Row (Zoomed) View

This panel displays the table column entries for an individual table row as panel input fields.

Press <PF5> to display the **Minor OR Sub-Expressions** panel from this panel view.

Press <PF3> to return to the table view.

```

SELCPY/i - Enter AND Sub-Expression
File Edit Actions Options Utilities Window SwapList Help  wS wR  Scroll> Csr
Command>
ZZSGEXP1
Lines 1-15 of 15
PF3=Table View

Structure:      CBL.SDE.COPYBOOK.COBOL          Member>  GENAMEMP
Record-Type:   AM                               Type:    Cob

Enter a logical AND expression which consists of one or more logically
ORed sub-expressions.

NOT Logical Operator:
  _ Negate Boolean condition set by this AND expression.

OR Sub-Expressions:
  Press <PF5> or click here to input OR sub-expressions.

```

Figure 62. 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 (SELCPY/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 separated 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, press <PF3> 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 single 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, place the cursor on the required table entry and press <PF2>.

```
SELCOPY/i - Minor OR Sub-Expressions
File Edit Actions Options Utilities Window SwapList Help wS wR
Command>
ZZSGEXP2
Add one or more alternate (logically ORed) expressions to the table below.

Enter "%" as the Field Name value to select from a list of formatted field
names and so identify the first term. <PF2> on a table row displays the row
in zoomed format which assists in second term field selection.

Structure: CBL.SDE.COPYBOOK.COBOL Member: GENAMEMP
Record-Type: AM Type: Cob

Enter Minor OR Expressions. 3 Rows
OR Field Name Ref# Fmt Pic ROp Value/Field
<> <...+...1...+...> <.> <> <...+...> <.> <...+...1...+...2...>
000000 *** Top of Data ***
000001 AMDATE 2 AN X(8) >= '2008'
000002 OR AMT 74 AN << #19 /* AMAELCUR /*
000003 OR AMIVC 49 AN X(12) = c'VCA@00025649'
000004 *** End of Data ***
```

Figure 63. Minor OR Sub-Expressions.

Single Row (Zoomed) View

This panel displays the table column entries for an individual table row as panel input fields.

Press <PF3> to return to the table view.

```

SELCOPY/i - Enter Minor OR Sub-Expression
File Edit Actions Options Utilities Window SwapList Help  wS wR  Scroll> Csr
Command>
ZZSGEXP2                                     Lines 1-20 of 22
                                                PF3=Table View
Structure:  CBL.SDE.COPYBOOK.COBOL          Member:  GENAMEMP
Record-Type: AM                             Type:    Cob

Enter "%" as the Field Name value to select from a list of formatted field
names and so identify the first term of the expression.

Term 1 (Field):
Field Name>
Ref#:      0      Level:  0      Data Format:      Pic:
-----

Operator:    (Enter blank for list of operators/definitions)
=___ Relational Operator.

Term 2:      (Select Character String/Numeric value or Field name)
/ String/Value>
-----
- Field Name> (Must correspond to the format of the source field.)
Ref#:      0      Level:  0      Data Format:      Pic:
-----
(Enter "%" to select from a list of formatted fields.)

```

Figure 64. 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-enterable fields as described for **Term 1** above.

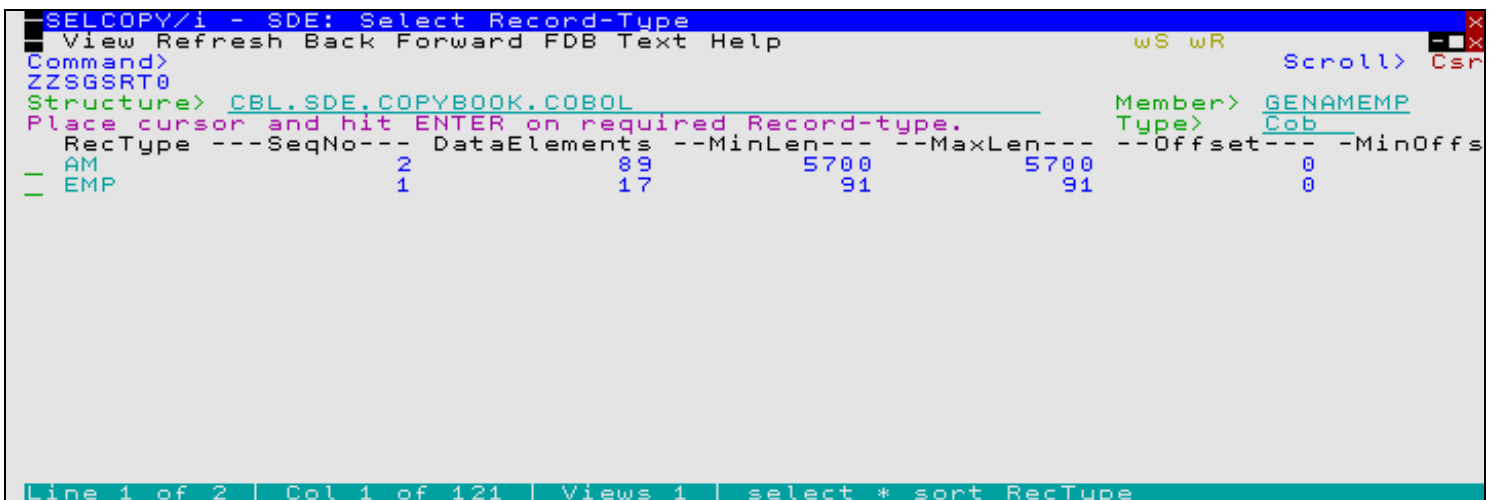
SDE: Select Record-Type

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.



```

SELPCOPY/i - SDE: Select Record-Type
View Refresh Back Forward FDB Text Help          wS wR          Scroll> Csr
Command>
ZZSGSRT0
Structure> CBL.SDE.COPYBOOK.COBOL                Member> GENAMEMP
Place cursor and hit ENTER on required Record-type. Type> Cob
RecType ---SeqNo--- DataElements --MinLen--- --MaxLen--- --Offset--- -MinOffs
-- AM                2             89          5700      5700         0
-- EMP               1             17          91         91          0

```

Line 1 of 2 | Col 1 of 121 | Views 1 | select * sort RecType

Figure 65. SDE: Select Record-Type.

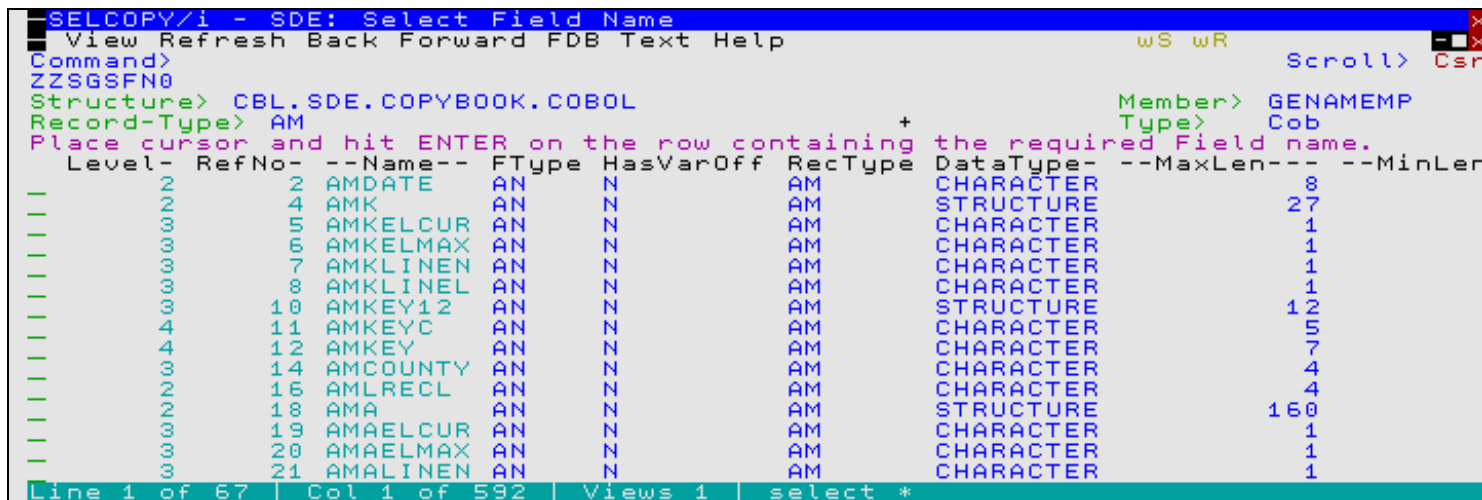
SDE: Select Field Name

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.



```

SELCOPY/i - SDE: Select Field Name
View Refresh Back Forward FDB Text Help          wS wR
Command>
ZZSGSFN0
Structure> CBL.SDE.COPYBOOK.COBOL                Member> GENAMEMP
Record-Type> AM                                  Type> Cob
Place cursor and hit ENTER on the row containing the required field name.
Level- RefNo- --Name-- FType HasVarOff RectType DataType- --MaxLen-- --MinLen
--
  2      2   AMLRECL  AN      N      AM      CHARACTER      4
  2      4   AMKELCUR AN      N      AM      CHARACTER      1
  3      5   AMKELMAX AN      N      AM      CHARACTER      1
  3      6   AMKELMAX AN      N      AM      CHARACTER      1
  3      7   AMKELMAX AN      N      AM      CHARACTER      1
  3      8   AMKELMAX AN      N      AM      CHARACTER      1
  3     10   AMKEY12  AN      N      AM      STRUCTURE      12
  4     11   AMKEYC   AN      N      AM      CHARACTER      5
  4     12   AMKEY    AN      N      AM      CHARACTER      7
  3     14   AMCOUNTY AN      N      AM      CHARACTER      4
  2     16   AMLRECL  AN      N      AM      CHARACTER      4
  2     18   AMA      AN      N      AM      STRUCTURE     160
  3     19   AMAELCUR  AN      N      AM      CHARACTER      1
  3     20   AMAELMAX  AN      N      AM      CHARACTER      1
  3     21   AMALINEN  AN      N      AM      CHARACTER      1
Line 1 of 67 | Col 1 of 592 | Views 1 | select *
  
```

Figure 66. SDE: Select Field Name.

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, press <PF3> 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, place the cursor on the required table entry and press <PF2>.

```

SELPCOPY/i - Multiple FIND Commands
File Edit Actions Options Utilities Window SwapList Help  wS wR  Scroll> Csr
Command>
ZZSGFSU4
Enter FIND Command Parameters. The specified logical operator indicates
whether all (AND) or any (OR) of the FIND commands must be true for a record
to satisfy the search criteria.          Logical Operator> AND
Enter FIND Operations.                  4 Rows
      Rop Find Value                    Scope      Left  Right
      <> <...+...1...+...2...+...3...> <...+> <...> <...>
000000 *** Top of Data ***
000001 EQ 'EXEC'          WORD          5      71
000002 EQ 'PGM='         PREFIX        16     71
000003 EQ 'PARM'         CHARS        16     71
000004 EQ 'MAP'          CHARS        16     71
000005 *** End of Data ***

```

Figure 67. SELPCOPY/i - Multiple FIND Commands.

Single Row (Zoomed) View

This panel displays the table column entries for an individual table row as panel input fields.

Press <PF3> to return to the table view.

```

SELPCOPY/i - Find Operation
File Edit Actions Options Utilities Window SwapList Help  wS wR  Scroll> Csr
Command>
ZZSGFSU4
Lines 1-20 of 24
PF3=Table View

Specify FIND command parameters:

Op> EQ Relational operator. (Enter "/" for list)
Value> 'EXEC'

Bounds: A single column or range of columns. (Default is ALL columns.)
Left Column> 5 0=>All Columns.
Right Column> 71 0=>Left Column only.

As> _ Unrestricted / Word _ Prefix _ Suffix

```

Figure 68. SELPCOPY/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, press <PF3> 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, place the cursor on the required table entry and press <PF2>.

```

SELPCOPY/i - Multiple FIND Commands
File Edit Actions Options Utilities Window SwapList Help  wS wR  Scroll> Csr
Command>
ZZSGFSU8
Enter FIND Command Parameters. The specified logical operator indicates
whether all (AND) or any (OR) of the FIND commands must be true for a record
to satisfy the search criteria.                               Logical Operator> OR
Enter FIND Operations.                                       3 Rows
      Rop Find Value                Scope  Left      Right
      <> <...+...1...+...2...> <...+> <...+...1...+> <...+...1...+>
000000 *** Top of Data ***
000001 EQ '2010/10/22'          CHARS  AMDATE
000002 EQ c'John Taylor'       CHARS  AMPM          AMPT
000003 EQ 'Yorkshire'         CHARS  AMCOUNTY
000004 *** End of Data ***

```

Figure 69. SELPCOPY/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.

Press <PF3> to return to the table view.

```

SELPCOPY/i - Find Operation
File Edit Actions Options Utilities Window SwapList Help  wS wR  Scroll> Csr
Command>
ZZSGFSU8
Lines 1-20 of 24
PF3=Table View

Specify FIND command parameters for a formatted record:
  Op> EQ Relational operator. (Enter "/" for list)
  Value> '2010/10/22'
  As> / Unrestricted _ Word _ Prefix _ Suffix
  Bounds: A single column or range of columns. (Enter % for column list.)
  Start Column> AMDATE + blank=>All Columns.
  End Column> + blank=>Start Column only.

```

Figure 70. SELPCOPY/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 *field_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, press <PF3> 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, place the cursor on the required table entry and press <PF2>.

```

SELCOPY/i - Multiple Change Commands
File Edit Actions Options Utilities Window SwapList Help  wS wR  Scroll> Csr
Command>
ZZSGFSU6
Enter Change Command Parameters. The logical operator indicates whether all
(AND) or only one (OR) of the Change operations are performed. OR=>the first
Change command for which the "From" value is found. Logical Operator> AND
Enter CHANGE Operations.
Rop FROM: TO: Scope Occur Src Left Right
Bnd Bnd Bnd
000000 <> <...+...1...+.> <...+...1...+.> <...+> <...> <...> <...> <...>
000001 *** Top of Data *** 'IEWL' CHARS FIRST DATA 16 71
000002 EQ 'HEWL' 'NOMAP' CHARS ALL DATA 16 71
000003 EQ 'MAP' c'AMODE=24' c'AMODE=31' CHARS ALL DATA 16 71
000004 *** End of Data ***

```

Figure 71. 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.

Press <PF3> to return to the table view.

```

SELCOPY/i - CHANGE Operation
File Edit Actions Options Utilities Window SwapList Help  wS wR  Scroll> Csr
Command>
ZZSGFSU6
Specify CHANGE command parameters:
Op> EQ Relational operator. (Enter "/" for list)
From> 'HEWL'
Bounds: A single column or range of columns. (Default is ALL columns.)
Left Column> 16 0=>All Columns.
Right Column> 71 0=>Left Column only.
As> / Unrestricted _ Word _ Prefix _ Suffix
To> 'IEWL'
Source> / DATA Do not attempt to preserve record data positions.
_ TEXT Preserve record data positions following changed data.
Occurs> _ ALL Change all occurrences within a record.
_ FIRST Change only the first occurrence within a record.
_ LAST Change only the last occurrence within a record.

```

Figure 72. 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 appropriate.

If the length of the replace value is greater than that of the search value, consecutive blanks 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, press <PF3> 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, place the cursor on the required table entry and press <PF2>.


```

SELCOPY/i - Multiple Change Commands
File Edit Actions Options Utilities Window SwapList Help  wS wR  Scroll> Csr
Command>
ZZSGFSU6
Enter Change Command Parameters. The logical operator indicates whether all
(AND) or only one (OR) of the Change operations are performed. OR=>the first
Change command for which the "From" value is found.      Logical Operator> AND
Enter CHANGE Operations.
      Rop FROM:                TO:                Scope  Occur  Src      Left  Right
      Bnd                      Bnd                      <...> <...> <...> <...> <...>
000000 <> <...+...1...+.> <...+...1...+.> <...+> <...> <...> <...> <...>
000000 **** Top of Data ****
000001 EQ 'HEWL' 'IEWL' CHARS FIRST DATA 16 71
000002 EQ 'MAP' 'NOMAP' CHARS ALL DATA 16 71
000003 EQ c'AMODE=24' c'AMODE=31' CHARS ALL DATA 16 71
000004 **** End of Data ****

```

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.

Press <PF3> to return to the table view.

```

SELCOPY/i - CHANGE Operation
File Edit Actions Options Utilities Window SwapList Help  wS wR  Scroll> Csr
Command>
ZZSGFSU6
Lines 1-20 of 24
PF3=Table View

Specify CHANGE command parameters:
Op> EQ Relational operator. (Enter "/" for list)
From> 'HEWL'
+
Bounds: A single column or range of columns. (Default is ALL columns.)
Left Column> 16 0=>All Columns.
Right Column> 71 0=>Left Column only.
As> / Unrestricted _ Word _ Prefix _ Suffix
To> 'IEWL'
+
Source> / DATA Do not attempt to preserve record data positions.
_ TEXT Preserve record data positions following changed data.
Occurs> _ ALL Change all occurrences within a record.
_ FIRST Change only the first occurrence within a record.
_ LAST Change only the last occurrence within a record.

```

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 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 *field_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 appropriate.

If the length of the replace value is greater than that of the search value, consecutive blanks 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 Output

Report Format

The report generated by the file Search/Update/Copy/Remap utility is a structured data file designed to be browsed (not 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 must be an already allocated, sequential data set. If not specified, the default DSN for the report and SDO data sets is "*prefix*.FSU.Dyyyyddd.Thhmmss" and "*prefix*.FSU.Dyyyyddd.Thhmmss.SDO" respectively. The high level qualifier, *prefix*, is the value assigned to **System.UserDSNPrefix** in the SELCOPY/i User INI file.

If the utility is run in the **background** as a batch job, then specification of a report output DSN is mandatory. 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 the 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 FSUEND 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 and the accompanying SDO as a physical sequential data set.

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 `FSOUT report_file_name.`
- Issue the prefix command `FO` against the report DSN in a data set list or VTOC list window.
- From the File Search/Update/Copy/Remap Panel, select menu item "Reports" and press the <Enter> key or, if configured, **double-click the left mouse button** on the required report file.

```

-CBLE
File Edit Actions Options Utilities Window SwapList Help      wS wR
-Edit NBJ2.FSU.D2008338.T133417:1 using NBJ2.FSU.D2008338.T133417.SDO -+X
Command>
Record type: Command      F(114)
Command
#3
AN 2:113
<---+---1---+---2---+---3---
000001 fsu where(#1 >> '/') f(('PGM='))

Record type: Summary      V(47,48)
RunType  RecordsTot  FilesTot
#3          #4          #5
AN 2:8      BN 10:4      BN 14:4
<---+---> <---+---> <---+---> <-
000002 UPDATE          5589          59

Record type: IOError      F(34)
zDsn      zMember      EnqErr  Open
#3          #5          #7
AN 2:11    AN 14:8      BN 23:1  BN 2
<---+---1> <---+---> <->
000003 NBJ.JCL.FSU BINDPKG          1

Record type: Hit          V(37,117)
zMember  zT      zRecord
#6        #13     #15
AN 14:8   AN 37:1 AN 38:80
<---+---> - <---+---1---+---2---+---3---+---4---+---5---
000004 BINDPLAN B //SDB2BIND EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=4M
==CHG> BINDPLAN A //SDB2BIND EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=0M
000006 CBLINST B //LOAD EXEC PGM=IEBCOPY,REGION=4M
==CHG> CBLINST A //LOAD EXEC PGM=IEBCOPY,REGION=0M
000008 CBLINS01 B //LOAD EXEC PGM=IEBCOPY,REGION=4M
==CHG> CBLINS01 A //LOAD EXEC PGM=IEBCOPY,REGION=0M
000010 CBLINS01 B //GETMEMS EXEC PGM=IEBCOPY,REGION=4M
==CHG> CBLINS01 A //GETMEMS EXEC PGM=IEBCOPY,REGION=0M
000012 CBLINS08 B //CBLICMX EXEC PGM=CBLAVARL,REGION=4M
==CHG> CBLINS08 A //CBLICMX EXEC PGM=CBLAVARL,REGION=0M
000014 COPYSEQ B //C1 EXEC PGM=IEBGENER,REGION=4M
--- Press F1 to edit file at cursor line ---
Se | Line=2 | Col=1 | Alt=0,0;0 | Size=33 | Recl=117 | Fmt=V | Files=1 | Views=

```

Figure 75. File Search/Update/Copy/Remap Output Output.

In the FSU output displayed in Figure 75., the user has hit the PF2 key 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 utility and so governs the format of the Hit records.

Operation	RunType
Unformatted/Formated Search	FIND
Unformatted/Formated Update	UPDATE or NOUPDATE
Unformatted/Formated 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 always 0 (zero).

ChgRecsErr

The total number of selected input records for which a CHANGE error has occurred.
For RunType "FIND", this value is always 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 always 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.
- 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.
- 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.
- 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.

<code>zDsn</code>	The DSN (or HFS path name) for which the I/O error occurred.
<code>zMember</code>	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.
<code>EnqErr</code>	1 if the error occurred when attempting to obtain an exclusive SPFEDIT ENQ for UPDATE on the file, otherwise 0.
<code>OpenErr</code>	1 if the error occurred when attempting to open the file, otherwise 0.
<code>ReadErr</code>	1 if the I/O error occurred when attempting to read a block of data from the file, otherwise 0.
<code>UpdErr</code>	1 if the I/O error occurred when attempting to re-write (update-in-place) a record to the file, otherwise 0.
<code>OutErr</code>	1 if the I/O error occurred when attempting to write a block of data to the OUTPUT file, otherwise 0.
<code>RecordsRead</code>	The number of records successfully read before the I/O error occurred.
<code>RecordsUpd</code>	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.

<code>zDsn</code>	The DSN of the PDS/PDSE library.
<code>zMember</code>	The PDS/PDSE member or member alias name.
<code>zAliasOf</code>	The PDS/PDSE member name for which this library entry is an alias.
<code>zHitRef</code>	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 occurred.

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 unexpected return code will be if a selection (run) time error (RC=44) has occurred. 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.

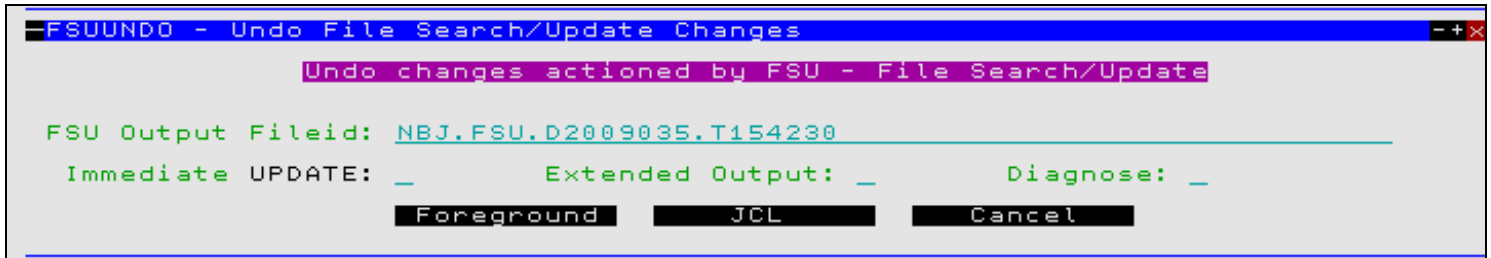


Figure 76. FSUUNDO - File Update Utility Undo Panel.

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 CBL 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.

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                   ## 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 ###
66CCEDCDE444CECC4DCD7CCCCDDE6DCCDD7FD44444444774C8A47774444444444444444444444
11753454200057530774E9523678B957965E040000000BB07530BBB000000000000000000000

Record Expected:
//GETMEMS     EXEC PGM=IEBCOPY,REGION=0M
66CCEDCDE444CECC4DCD7CCCCDDE6DCCDD7FD4444444444444444444444444444444444444444
11753454200057530774E9523678B957965E0400000000000000000000000000000000000000
.....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 76. FSUNDO TERSE Output with VERIFY.


```

//GETMEMS EXEC PGM=IEBCOPY,REGION=4M      ## Get ###
66CCEDCDE444CECC4DCD7CCCCDDE6DCCDD7FD44444444774C8A477744444444444444444444444
11753454200057530774E9523678B957965E440000000BB07530BBB00000000000000000000000

Record Expected:
//GETMEMS EXEC PGM=IEBCOPY,REGION=4M
66CCEDCDE444CECC4DCD7CCCCDDE6DCCDD7FD44444444444444444444444444444444444444444
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
66EDDDCC4444CECC4DCD7CCCCDDE6DCCDD7FD4444444444444444444444444444444444444444
11453614000057530774E9523678B957965E040000000000000000000000000000000000000000

Record New:
//UNLOAD EXEC PGM=IEBCOPY,REGION=0M
66EDDDCC4444CECC4DCD7CCCCDDE6DCCDD7FD4444444444444444444444444444444444444444
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 76. 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
- 2 Compare Libraries

COMPF - Compare Files
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 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, COMPFILE utilises functions and features provided by the structured data environment (SDE). Therefore, the COMPFILE utility is only available to users who have a licensed version of SELCOPY installed and operational on their system.

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.
- GDG datasets.
- VSAM (KSDS, ESDS, RRDS, VRDS).
- HFS Files.
- (DB2 Tables planned but not yet supported).

Output Report

The report generated by the compare files 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 compare.

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 have been inserted into the NEW file.

Deleted

Records that no longer exist in the NEW file and so 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 discriminate 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 *"SELCOPY/i 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 "*Unsyncronised 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 that 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 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 synchronisation 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 synchronisation 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 synchronised record pair so sorted key synchronisation processing occurs.

Compare Files Panels

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

```

SELCOPY/i - Compare Files: Basic Options
File Help
Command>
ZZSGCF00
Files: PDS(E) member, Sequential, VSAM dataset or HFS path (PF5=CMX PF6=JCL)
New File> _____ + Member> _____
Volume> _____ If dataset is uncataloged.

Old File> _____ + Member> _____
Volume> _____ If dataset is uncataloged.

- Use Extended options e.g. Formatted compare, Keyed synchronisation etc.
  Sync > Read-Ahead Synchronisation type (Read-Ahead or 1-to-1)
  Limit > 0 Halt after this number of differences. (0=no limit)
  Include Matches Show matched records in the output report.

Compare Position/Length:
- Pos > 0 Start comparison at this position within the record.
- Length > 0 Number of bytes to compare.
- Strip > Ignore trailing '?' or X'??' differences.

Record Selection:
- Start > _____ + / Record - Key - RBA
- For > 0 # records
  
```

Figure 77. 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 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.

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 and 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 specified 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.

Sync>

Select either **Read-Ahead** or **1-to-1**.

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.

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).

Include Matches

Corresponding records from both NEW and OLD files that match are to be included as matched records in the output report file.

This field corresponds to the COMPFILE parameter INCMATCHED.

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).

Compare Type - Formatted/Unformatted ?

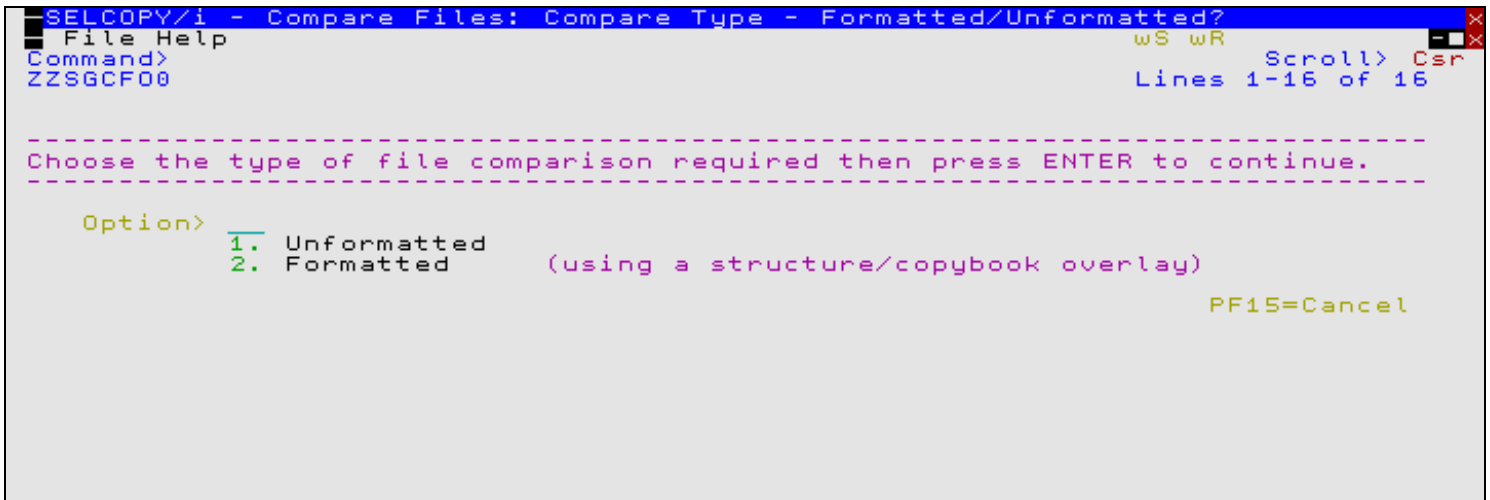


Figure 78. SELCOPY/i - Compare Files: Compare Type - Formatted/Unformatted?

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 an 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

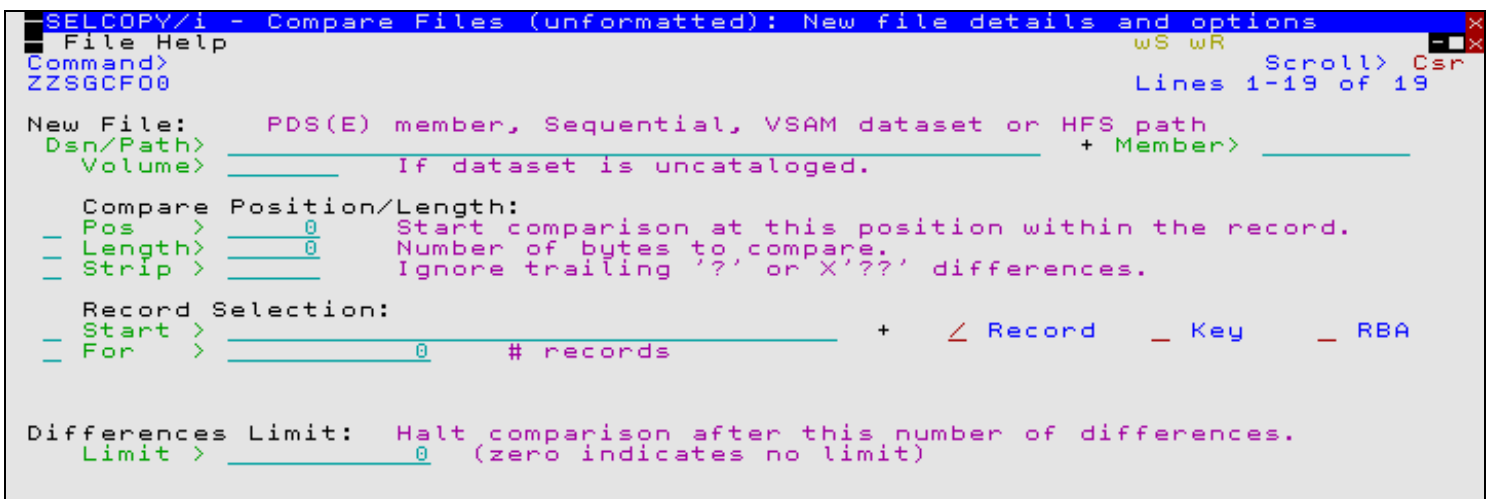


Figure 79. SELCOPY/i - Compare Files (unformatted): New file details and options.

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 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

```

SELCOPY/i - Compare Files (unformatted): Old file details and options
File Help
Command>
ZZSGCF00
Old File:
Dsn/Path>
Volume>
Compare Position/Length:
Pos>
Length>
Strip>
Record Selection:
Start>
For>
Record
Key
RBA
  
```

Figure 80. 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 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

```

SELCOPY/i - Compare Files (unformatted): Re-synchronisation options
File Help                                     wS wR
Command>                                     Scroll> Csr
ZZSGCF00                                     Lines 1-20 of 21

Synchronisation:
 / Read-Ahead ...
   a maximum of: 100 rec(s). Re-sync on: 1 matching rec(s).
   - Use blank lines to re-sync.
- 1-to-1
- Keyed (Sorted)
- Keyed (Unsorted)
Report:
- Include Matched
- Exclude Changed
- Exclude Inserted
- Exclude Deleted

Upper/Lower Case:
- Perform case-insensitive compare

Report File: (default is 'userid.SELCOPYI.COMPFILE.REPORT')
- Dsn> _____ Member> _____
- Volume> _____ If dataset is uncataloged.

Note: The report must be viewed using a SELCOPY/i structure-definition
object, which is dynamically created by adding '.SDO' to the above dsn.

```

Figure 81. 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 resynchronise 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 COMPFIL 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 COMPFIL 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 COMPFIL 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 COMPFIL 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 COMPFIL 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 COMPFIL 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.

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 Deleted

Select this option to exclude deleted records from the output report file.

This option corresponds to COMPFILE parameter EXDELETED.

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).

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.

The report is a structured data file designed to be browsed (not printed) using an SDE 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 eligible 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.

Compare Files (unformatted): Specify Key fields

```

SELCOPY/i - Compare Files (unformatted): Specify Key fields
File Help
Command> ZZSGCF00
-----
Choose the action required by entering a number in the input field then
pressing ENTER, or alternatively point-and-shoot at the option description.
-----
Option> 1          Blank to continue.
          1. Specify Key fields (Position/Length)
  
```

Figure 82. 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

```

SELCOPY/i - SDE CompFile - KEY Columns clause
File Help
Command> ZZSGCFKP
Select Key Length/Position(s):
Insert a table row corresponding to each required key segment.
SDE CompFile - KEY Columns clause.
Key Length Key Pos New Key Pos Old
<...+...1> <...+...1> <...+...1>
000001      12      1      101
000002       5     51     201
000003 *** End of Data ***
  
```

Figure 83. 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

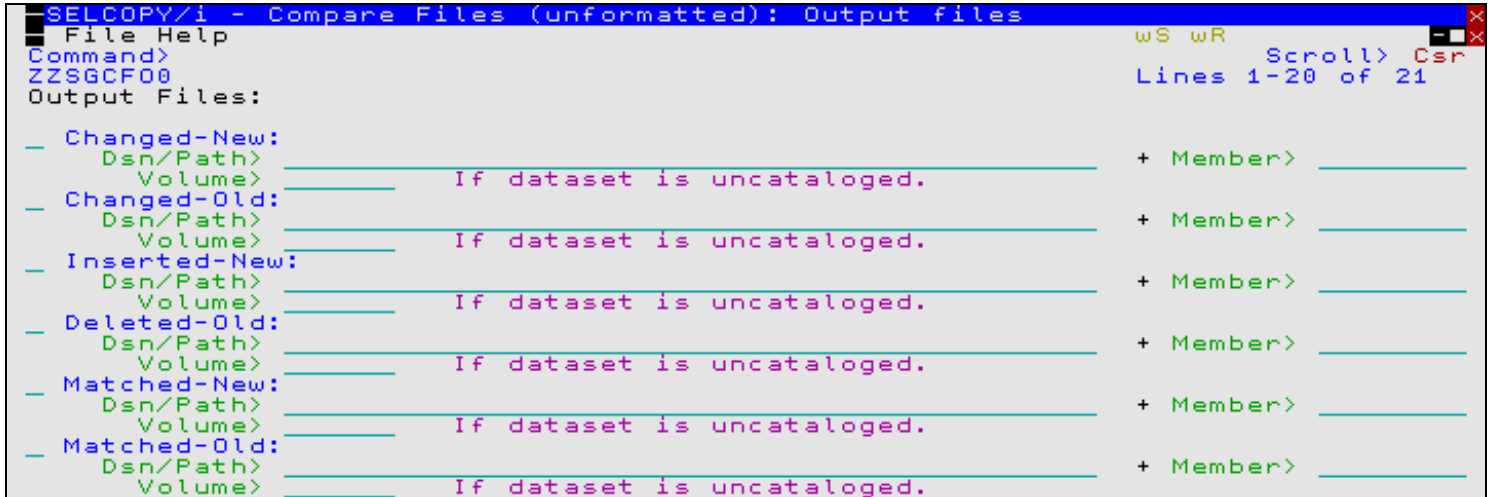


Figure 84. SELCOPY/i - Compare Files (unformatted): Output Files.

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 eligible 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

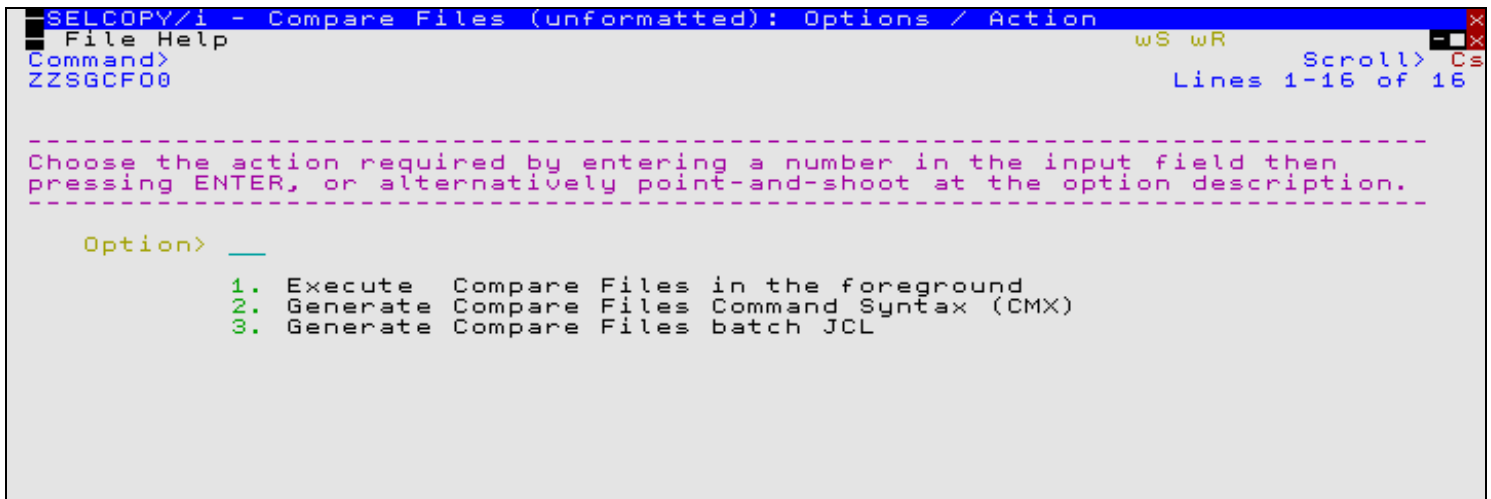


Figure 85. SELCOPY/i - Compare Files (unformatted): Options / Action.

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

```

SELCOPY/i - Compare Files (formatted): New file details and options
File Help
Command>
ZZSGCF00
New File: PDS(E) member, Sequential, VSAM dataset or HFS path
Dsn/Path> _____ + Member> _____
Volume> _____ If dataset is uncataloged.
Structure/Copybook overlay
Dsn> _____ Member> _____
Volume> _____ If dataset is uncataloged.
Type: / SDO _ AData _ Cobol _ PL1
Record Selection:
Start > _____ + / Record _ Key _ RBA
For > _____ # records
Differences Limit: Halt comparison after this number of differences.
Limit > _____ (zero indicates no limit)
  
```

Figure 86. SELCOPY/i - 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 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> Y/N

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

```

SELCOPY/i - Compare Files (formatted): Old file details and options
File Help                                     wS wR
Command>                                     Scroll> CS
ZZSGCF00                                     Lines 1-16 of 16

Old File:      PDS(E) member, Sequential, VSAM dataset or HFS path
Dsn/Path> _____ + Member> _____
Volume> _____ If dataset is uncataloged.

_ Structure/Copybook overlay (Select only if different from New file)
Dsn> _____ Member> _____
Volume> _____ If dataset is uncataloged.
Type:  / SDO    _ AData  _ Cobol  _ PL1

Record Selection:
_ Start > _____ + / Record  _ Key    _ RBA
_ For   > _____ # records

```

Figure 87. SELCOPY/i - 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 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.

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. Record type definitions of the same name must exist in both the NEW and OLD structures. Furthermore, only fields of the same name within these matching record types will be compared. i.e. Only records assigned a record type that is defined in both structures, are eligible for compare and only record data occupying fields of equal field name are compared.

Fields of the same name 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 ',').

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.

Recompile> Y/N

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

```

SELCOPY/i - Compare Files (formatted): Re-synchronisation options
File Help                               wS wR
Command>                                Scroll> CS
ZZSGCF00                                Lines 1-20 of 20

Synchronisation:
 / Read-Ahead a max of: 100 rec(s). Re-sync on: 1 matching rec(s).
- 1-to-1
- Keyed (Sorted)
- Keyed (Unsorted)

Report:
- Include Matched
- Exclude Changed
- Exclude Changed Field Names
- Exclude Inserted
- Exclude Deleted

Upper/Lower Case:
_ Perform case-insensitive compare

Report File: (default is 'userid.SELCOPYI.COMPFILE.REPORT')
- Dsn> _____ Member> _____
  Volume> _____ If dataset is uncataloged.

Note: The report must be viewed using a SELCOPY/i structure-definition
object, which is dynamically created by adding '.SDO' to the above dsn.

```

Figure 88. SELCOPY/i - 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 do the following:

- If Read-Ahead or 1-to-1 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 resynchronise 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.

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 Inserted

Select this option to exclude inserted records from the output report file.

This option corresponds to COMPFILE parameter EXINSERTED.

Exclude Deleted

Select this option to exclude deleted records from the output report file.

This option corresponds to COMPFILE parameter EXDELETED.

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.

The report is a structured data file designed to be browsed (not printed) using an SDE 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 eligible 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.

Compare Files (formatted): Specify Key fields

```

SELCOPY/i - Compare Files (formatted): Specify Key fields
File Help
Command> ZZSGCF00
-----
Choose the action required by entering a number in the input field then
pressing ENTER, or alternatively point-and-shoot at the option description.
-----
Option> 1 Blank to continue.
1. Select Key Columns by fixed Position/Length
2. Select Key Columns by Name
-----
Lines 1-16 of 16

```

Figure 89. SELCOPY/i - 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

```

SELCOPY/i - SDE CompFile - KEY Columns clause
File Help
Command>
ZZSGCFKP
Select Key Length/Position(s):
Insert a table row corresponding to each required key segment.
SDE CompFile - KEY Columns clause.
Key Length Key Pos New Key Pos Old
<.....1> <.....1> <.....1>
000001      12      1     101
000002       5     51     201
000003 *** End of Data ***
  
```

Figure 90. 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 (formatted): 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 (formatted): Specify Key fields** panel view.

SELCOPY/i Compare Files - KEY Columns (Record Types List)


```

SELCOPY/i - SDE CompFile - KEY Columns clause
File Help
Command>
ZZSGCFKC
Select Key Columns from Structure/Copybook:
Using: JGE.CBLI.SDO(SALES)
Type: / SDO AData Cobol PL1
SDE CompFile - KEY Columns clause.
Sel Record Type Fields Selected Use When
+
000000 *** Top of Data ***
000001 S REC-CUST > 0 specified
000002 REC-CARD > 0 specified
000003 S REC-ORDER > 0 specified
000004 REC-PAYMENT > 0 specified
000005 REC-NOTE > 0 specified
000006 *** End of Data ***

```

Figure 91. SELCOPY/i - 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 displayed 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 SELCOPY/i **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.

SELCOPY/i Compare Files - Select from Field Names List

```

SELCOPY/i - SDE CompFile - Field names list
File Help
Command>
ZZSGCFOF
Record-Type: REC-NOTE
SDE CompFile - Field names list.
Lev Field Field Pic Max Len Min Len Struct Parent
Name Type Type Offset Offset
+ +
000001 --- 2 CUST-ID FB 9(5) 2 row(s) excluded ---
000003 2 CUST-ID FB 9(5) 4 4 8 8
000004 --- 1 NOTE AN X(50) 1 row(s) excluded ---
000005 2 NOTE AN X(50) 50 50 16 16
000006 *** End of Data ***

```

Figure 92. SELCOPY/i - 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 SELCOPY/i **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

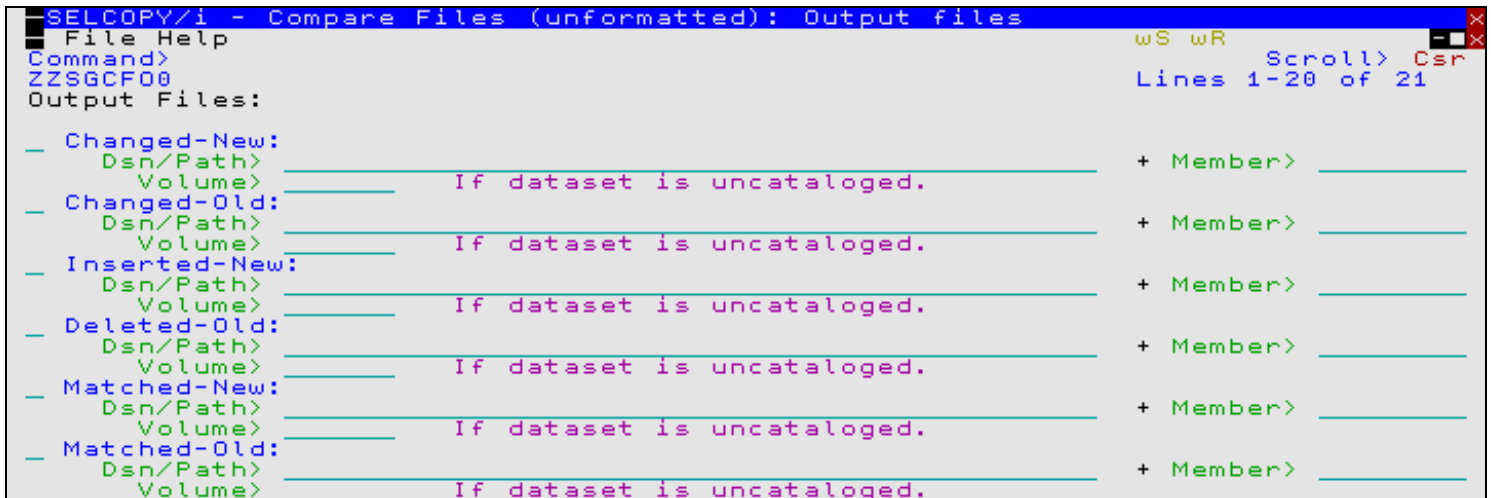


Figure 93. SELCOPY/i - Compare Files (formatted): Output Files.

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 eligible 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

```
SELCOPY/i - Compare Files (formatted): Options / Action
File Help
Command>
ZZSGCF00
WS WR
Scroll> CS
Lines 1-16 of 16

-----
Choose the action required by entering a number in the input field then
pressing ENTER, or alternatively point-and-shoot at the option description.
-----

Option> ___

1. Select Record-types for comparison
2. Select Column Names for comparison
3. Execute Compare Files in the foreground
4. Generate Compare Files Command Syntax (CMX)
5. Generate Compare Files batch JCL
```

Figure 94. SELCOPY/i - 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 SELCOPY/i 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 **SDEMAIN** 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 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.

SELCOPY/i Compare Files - Select Record Types to Compare

```

SELCOPY/i - SELCOPY/i Compare Files: Choose record-types to compare
File Help                                     wS wR   Scroll> Csr
Command>
ZZSGCFOV
Select record-types to compare from Structure/Copybook:      PF1=Help
Name> JGE.CBLI.SDO(SALES) +
SDE CompFile - VIEW clause.                               5 Rows
Record Type Use When
+
000000 *** Top of Data ***
000001 REC-CUST
000002 ----- 1 row(s) excluded -----
000003 REC-ORDER
000004 ----- 1 row(s) excluded -----
000005 REC-NOTE
000006 *** End of Data ***

```

Figure 95. SELCOPY/i - 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 SELCOPY/i **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.

SELCOPY/i Compare Files - Select Field Names to Compare

```
SELCOPY/i - CompFile: Select Field Names to Compare
File Help
Command>
ZSNGCFOS
Select record-types from Structure/Copybook:
Using: JGE.CBLI.SDO(SALES)
Type: / SDO AData Cobol PL1
SDE CompFile - SELECT clause.
Sel Record Type Fields Selected Use When
*** Top of Data ***
000001 S REC-CUST > 0 specified
000002 S REC-CARD > 0 specified
000003 S REC-ORDER > 0 specified
000004 S REC-PAYMENT > 0 specified
000005 S REC-NOTE > 0 specified
000006 *** End of Data ***
5 Rows
```

Figure 96. SELCOPY/i - Compare Files - Select Field Names to Compare.

The **Compare Files - Select Field Names to Compare** panel (ZSNGCFOS) 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 (ZSNGCFOF) 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 displayed. 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** (ZSNGCFOV), 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*.

SELCOPY/i Compare Files - Select Field Names List

Lev	Field Name	Field Type	Field Pic	Max Len	Min Len	Struct Offset	Parent Offset
000001				2	row(s)	excluded	
000003	2 CUST-ID	FB	9(5)	4	4	8	8
000004				1	row(s)	excluded	
000005	2 NOTE	AN	X(50)	50	50	16	16
000006	*** End of Data ***						

Figure 97. SELCOPY/i - 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 SELCOPY/i **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 a structured data file designed to be browsed (not printed) using a structure definition object (SDO) within a SELCOPY/i 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 eligible 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, 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 compare files 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.

```

SELCOPY/i - Browse NBJ2.SELCOPYI.COMPFILE.REPORT using NBJ2.SELCOPYI.COMPFILE
File Edit Actions Options Utilities Window SwapList Help  WS WR  Scroll> Csr
Command>
Record type: Compare  Variable(13,268) Offset=0 Data elements=9
zId zRecord
<> <---+---1---+---2---+---3---+---4---+---5---+---6---+---7---+
AddSELECT:
if xSELECT <> '/' then return /* Option not active */
/* ***.select .sel */
'ec i  SELECT
zSELECT = strip(zSELECT)
do forever
  l=length(zSELECT);          if l < 50 then leave
  if p = 0 then p=lastpos(' ',left(zSELECT,50));
  if p = 0 then leave
  p=lastpos(' ',left(zSELECT,50)); if p = 0 then leave
  interpret 'parse var zSELECT  SELECT' p 'zSELECT'
  SELECT = strip( SELECT)
  zSELECT = strip(zSELECT)
  'ec i          'left( SELECT,53) '\ '
  'ec i          'left(xSELECT,53) '\ '
end
'ec i          'left(zSELECT,53) '\ '
return

AddWHERE:
if xWHERE <> '/' then return /* Option not active */
/* ***.where .wh */
'ec i  WHERE (
zWHERE = strip(zWHERE)
if pos('|',zWHERE) > 0 then IncludesOR=1
if pos('linend.2,zWHERE) > 0 then IncludesLE=1
do forever
  l=length(zWHERE);          if l < 50 then leave
  p=lastpos(' ',left(zWHERE,50)); if p = 0 then leave
  interpret 'parse var zWHERE  WHERE' p 'zWHERE'
  WHERE = strip( WHERE)
  'ec i          'left(zWHERE,53) '\ '
end
'ec i          'left(zWHERE,53) '\ '
return

---- 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 | V

```

Figure 98. SELCOPY/i - Compare Files - Output Report - 80 column screen.

To view these suppressed fields press <PF2> 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.) Alternatively, issue the command **SElECT *** with the cursor positioned on any record of record type Compare to reveal the suppressed fields. If necessary, the command **PREfix ON n** will provide a prefix area of length *n* (default is 8).

```

SELCOPY/i - Browse NBJ2.SELCOPYI.COMPFILE.REPORT:2 using NBJ2.SELCOPYI.COMPFI
File Edit Actions Options Utilities Window SwapList Help  WS WR  Scroll> Csr
Command>
Record type: Compare  Variable(13,268) Offset=0 Data elements=9
Record> 00000380  Flags: f          Length: 69

Field      Data
3 zId
3 zNewRecNo 0000000344
3 zOldRecNo 0000000342
3 zLrecl    56
3 zRecord   l=length(zSELECT);          if l < 50 then leave

```

Figure 99. SELCOPY/i - Compare Files - Output Report - Zoomed view.

A compare files output report is viewed on a 3270 terminal of width greater than 80 characters will include a prefix area with no report field suppression.

```

SELCOPIY/i - Browse NBJ2.SELCOPIY1.COMPFILE.REPORT using NBJ2.SELCOPIY1.COMPFILE.REPORT.SDO 32752 V SEQ
File Edit Actions Options Utilities Window SwapList Help WS WR
Command>
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 = @ then p=lastpos(' ',left(zSELECT,50));
00000383 I 0000000347 24 if p = @ then leave
00000384 D 0000000343 57 p=lastpos(' ',left(zSELECT,50)); if p = @ 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 = @ 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;@ | Size=693 | Recl=32752 | Fmt=V | Files=1 | Views=1 | 2012/04/13 10:19:35

```

Figure 100. SELCOPIY/i - Compare Files - Output Report - wide-screen.

The report file is comprised of several different 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 preceded by a report record with zID field "CN" for the corresponding NEW file record in the record pair. Changed-Old records are colour-coded YELLOW .
H	Applicable to Basic and Extended Unformatted Compare only, a highlight-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 . Highlight-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>
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)
00000005 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

```

Figure 101. 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 <i>Compare-record_type</i> 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 <i>SELECT field FROM record_type</i>), 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 separate Field report record is written for each field specifically selected for caompare in each record type. 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.

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.

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 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.
<PF6>	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.

The compare process utilises the SELCOPY program with input control statements being read from member ZZSCOMPL of the distributed sample library SZSSAM1. 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 **COMPFIL** command to compare the two library members individually. When the SYSPRINT output is displayed in a SELCOPY/i text edit view, the generated COMPFIL commands may be executed using the ACTION facility (i.e simply position the cursor on the required COMPFIL 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.

```

Compare Library Members
File Run Command JCL Help
Command>
ZZSCOMPL
Lines 1-14 of 14
Scroll> Csr

Libraries:
New DSN > CBL.CBLI190.ASM
Old DSN > CBL.CBLI190.ASM.COPY

Select Member(s):
Pattern 1> CNV* (Single Character Wildcard = %)
Pattern 2> EDT* (Multiple Character Wildcard = *)
Pattern 3>
Pattern 4>

Options:
Strip > NO Ignore trailing ' ' differences.
  
```

Figure 102. Compare 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 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 **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

New DSN>
This input field (LIBRARY1) is mandatory and identifies the NEW PDS or PDSE library DSN to be compared.

Old DSN>
This input field (LIBRARY2) is mandatory and identifies the OLD PDS or PDSE library DSN to be compared.

Select Members (s) :
Pattern 1>
Pattern 2>
Pattern 3>
Pattern 4>

These input fields (MEMBER1, MEMBER2, MEMBER3 and MEMBER4) are optional and allow the user to provide up to 4 alternative member name masks for selecting members to be compared from the **NEW library**. If a NEW library member name matches any of these masks, then it will be selected for compare.

Member names in the OLD library that match this mask but do not exist in the NEW library will **not** be included for compare.

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.

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 contains "YES".

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 (CBL) text editor.

Each line of the output report corresponds to a single member compare.

```

NBJ2.SELCOPY.D2012005.T1236038.SYSPRINT      133 F SEQ      Size=10      Alt=0,0;0
Command>
-----Status----- | ---Compare Command--- (Use PF4 to compare individual member)
Old mbr + recs       | <CompFile NBJ.CBLI310.ASM.NEW(ZZSGCFKP) NBJ.CBLI310.ASM(ZZSGC
New mbr missing      | <CompFile NBJ.CBLI310.ASM.NEW(ZZSGCFOS) NBJ.CBLI310.ASM(ZZSGC
Data mismatch        | <CompFile NBJ.CBLI310.ASM.NEW(ZZSGFLTJ) NBJ.CBLI310.ASM(ZZSGF
New mbr + recs       | <CompFile NBJ.CBLI310.ASM.NEW(ZZSGFLTW) NBJ.CBLI310.ASM(ZZSGF
Data mismatch        | <CompFile NBJ.CBLI310.ASM.NEW(ZZSGFLT0) NBJ.CBLI310.ASM(ZZSGF
Old mbr missing      | <CompFile NBJ.CBLI310.ASM.NEW(ZZSNCF0F) NBJ.CBLI310.ASM(ZZSNC
Data mismatch        | <CompFile NBJ.CBLI310.ASM.NEW(ZZS2CSDS) NBJ.CBLI310.ASM(ZZS2C
Data mismatch        | <CompFile NBJ.CBLI310.ASM.NEW(ZZS2CTAL) NBJ.CBLI310.ASM(ZZS2C
Data mismatch        | <CompFile NBJ.CBLI310.ASM.NEW(ZZS2CTAN) NBJ.CBLI310.ASM(ZZS2C
Data mismatch        | <CompFile NBJ.CBLI310.ASM.NEW(ZZS2CTAU) NBJ.CBLI310.ASM(ZZS2C
* * * End of File * * *

```

Figure 103. 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 invokes 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)

SELCOPY/i can invoke the SELCOPY program allowing interactive execution and debug within SELCOPY/i windows.

- [Invoking SELCOPY Debug](#)
- [Load Library Search Chain](#)
- [SELCOPY Debug Windows](#)
- [SELCOPY Debug Commands](#)
- [SELCOPY Debug Function Keys](#)

Invoking SELCOPY Debug

The SELCOPY Debug application window is a special instance of the CBLLe Text Edit application and always opens as an MDI frame window within the SELCOPY/i main window display area (i.e not as a child window of the invoking CBLLe Text Edit main window.) See [SELCOPY Debug Main Window](#) for information on the SELCOPY Debug window environment.

SELCOPY Debug is started via the following:

- Select option 1. 'SELCOPY/debug' from the [Utilities Menu](#)
- Select 'SELCOPY Debug/Dev' from the File menu in the [CBLLe main window menu bar](#).
- Enter the command [SELCOPY](#) on the command line of any window.

See also the CBLLe macro, [JCLCMX](#), which is a pre-processor tool used to convert MVS JCL decks containing SELCOPY steps into a format suitable for input to SELCOPY Debug. Simply edit the JCL deck using CBLLe, then key in JCLCMX from the edit command line to generate command files containing ALLOC commands that correspond to the DD statements; CALL commands that correspond to EXEC statements; and SELCOPY commands that correspond to EXEC PGM=SELCOPY statements.

A control file name containing the SELCOPY source statements must be provided as SYSIN/SYSIPT input to the SELCOPY Debug window either as a parameter on the SELCOPY command or via the "Run SELCOPY" dialog window. The control file may be specified as a complete fileid (DSN) or as a previously allocated filename (DD/FILEDEF/DLBL) which may be a concatenation of data sets. **e.g.**

```
selcopy -ctl cbl.ssc.ctl(ssdemo01)
```

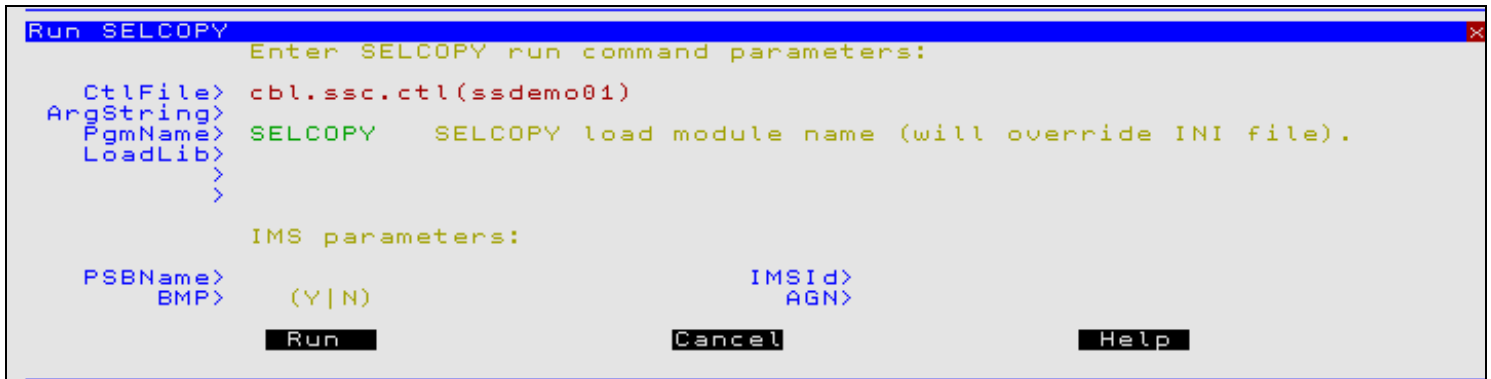


Figure 104. Run SELCOPY Dialog Window.

If the specified control file is empty or does not yet exist, then SELCOPY Debug opens an empty SYSIN/SYSIPT CBLLe text edit 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 begin debugging the new SELCOPY statements. If the SELCOPY control fileid does not already exist, then for MVS systems, the [Allocate NonVSAM](#) dialog will be opened before the save is actioned.

When SELCOPY Debug starts, the Execute SELCOPY main window is opened together with a CBLLe edit view for the control statement SYSIN/SYSIPT input. If the SYSIN/SYSIPT file name was specified as an explicit fileid (DSN) which is not locked (ENQ'd) by another process and the user has sufficient authority, the file is edited Read/Write. Where the SYSIN/SYSIPT input file name was specified as a previously allocated filename (DD/FILEDEF/DLBL), then the data is edited Read/Only.

The SELCOPY program is then executed and stopped following control statement analysis but before actioning the first executable statement. The user may then begin debugging and stepping through the statements.

Note that interactive execution of SELCOPY using statement stepping and/or break points is not supported if the SELCOPY control 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.

Any required SELCOPY input or output files must be allocated before execution of SELCOPY Debug. This excludes SYSIN and SYSPRINT which are handled by the SELCOPY/i window management software. See the CBL `ALLOCATE` command and, where the SELCOPY statements form part of an MVS batch stream, use the CBL `JCLCMX` REXX macro to configure the environment prior to starting SELCOPY Debug.

SELCOPY
WS WR

-SYSPRINT: NBJ2.SELCOPY.SSDDEM001.SYSPRINT 133 V 3
-Work Area

Command>
Scroll> Csr

```

|...+...1...+...2...+...3...+...4...+
000391 | 127 127 20 EQU AMTEND 5190
000392 | 128 128 20
000393 | 129 129 20 EQU AMV 5201
000394 | 130 130 20 EQU AMVELCUR 5201
000395 | 131 131 20 EQU AMVELMAX 5202
000396 | 132 132 20 EQU AMVLINEN 5203
000397 | 133 133 20 EQU AMVLINEL 5204
000398 | 134 134 20 EQU AMVL 30
000399 | 135 135 20 EQU AMVFIRST 5211 *
000400 | 136 136 20 EQU AMVLAST 5211 *
000401 | 137 137 20 EQU AMVEND 5480
                    
```

-Pos @ARR (WorkArea POS 11101)
Scroll> Csr

Command>
Scroll> Csr

```

1 40404040 40404040 40404040 40404040
                    
```

-Pos PDSIN (WorkArea POS 20801)
Scroll> Csr

Command>
Scroll> Csr

```

1 C2C9E3F0 F2404040 00020C00 01070035 BIT02 . . .
17 0101171F 0101171F 16240005 00050000 . . .
33 D1C7C540 40404040 40404040 40404040 JGE
                    
```

-Pos TOT
Scroll> Csr

Command>
Scroll> Csr

```

1 00000003
                    
```

-Pos MAT
Scroll> Csr

Command>
Scroll> Csr

```

1 00000002
                    
```

-Pos UNM
Scroll> Csr

Command>
Scroll> Csr

```

1 00000001
                    
```

-Equates
View Refresh Back Forward FDB Text Help

Command>
View Refresh Back Forward FDB

EQUName	-PosValue-	PtrName	Address-
IN1	1841	@	000C5170
LSTR	11101	@ARR	000C759C
LSTRL	12	LRecl	000C4A4B

-SYSIN: CBL.SSC.CTL(SSDEM001) 218 V PDSE Size=100 Alt=0,0;6
Scroll> Csr

Command>
Scroll> Csr

```

|...+...1...+...2...+...3...+...4...+...5...+...6...+...
000037 @arr = marr
000038 log 'About to enter loop.'
000039 ==memloop== * Loop to build array of member names.
000040 if @arr > marr+marrl-1 !t plog 'Array exceeded' !t eoj
000041
000042
000043
000044
000045 rd pds1 dsn=in1 dir into @arr * Directory records only
000046 pos @arr+8, @arr+lrecl-1 = * Blank rest of record fo
000047
000048 if eof p
000049
000050 *memloope*
000051 log 'Exit
000052
000053 print fro
000054
000055 pos tot l
000056 pos mat l
000057 pos unm l
000058
000059 ==pdsloop==
000060 rd p
000061 if eof p
000062
000063
000064
000065
000066
000067
000068
000069 if data pds2
                    
```

-WTOL0G: NBJ2.SELCOPY.SSDDEM001.WTOL0G
Command>

Command>
Command>

```

|...+...1...+...2...+...3
000000 * * * Top of File * * *
000001 Program about to start -----
000002 About to enter loop.
000003 Exit loop.
000004 BBCARD01
000005 * * * End of File * * *
                    
```

-TRACE: NBJ2.SELCOPY.SSDDEM001.TRACE
Command>

Command>
Command>

```

|...+...1...+...2...+...3
000140 if dir pds2
000141 if dir pds2
000142 if dir pds2
000143 then add 1
000144 goto pdsloop
000145 then add 1
000146 then space
000147 16. rd pds2 dsn=in
000148 * * * End of File * * *
                    
```

Edit Keys
Command> if data pds2...

Watch Pos "pds2"
Print array for debug.

```

----- "pds2" <edit>
Track "pds2"
----- "pds2" <edit>
                    
```

Track List
* Initialise to hex zeroe

Break <toggle>
* Initialise to hex zeroe

```

* Does the same thing.
                    
```

Goto EOJ
* Dir + Data records only

Window Layout
* Log totals.

```

* Close input library.
* End of job.
                    
```

F1=StepOver F2=StepInto F4=Point-and-Shoot Options F13=Go F14=BreakPoi

Te | Line=37 | Col=1 | Alt=0,0;6 | Size=100 | Recl=218

Figure 105. SELCOPY Debug in 58x120 3270 Session.

Under MVS, interactive execution of IMS SELCOPY programs in DLI and BMP region types is supported. The following SELCOPY Debug attributes relating to IMS SELCOPY programs may be specified as a parameter on the SELCOPY command or via the "Run SELCOPY" dialog window.

- The PSB name.
- The IMS region (either DLI or BMP). Default is DLI.
- The Subsystem name of the IMS Region to which the connection should be made. **(Optional)**
- IMS Application Group name. **(Optional)**

For IMS DLI regions, IEFORDER, DFSVSAMP must either be allocated or included in the SELCOPY/i System or User INI file for dynamic allocation by SELCOPY Debug at execution time.

Load Library Search Chain

The location of the SELCOPY program executed and any routines called by the SELCOPY operation, CALL, 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, COBOL routine or any MVS program module developed using Language Environment.

For MVS 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 "Run SELCOPY" dialog window or via the -LIB parameter on the SELCOPY line 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 separated by ',' (commas), ';' (semi-colons) or ' ' (blanks).

Note: If the DSNs are separated by blanks, quotes must also be used to delimit the list of DSNs, not the individual DSNs.

The following SELCOPY line command and "Run SELCOPY" dialog examples illustrate use of LibPath.

```
selcopy -ctl INCTL -lib "SYS3.TEST.LOADLIB; SYS3.NBJ.TEST.EXE"
```



Figure 106. Run SELCOPY Dialog Window with Load Library Path.

This feature is exploited by the **JCLCMX** CBLc REXX macro which translates JCL statements from a batch job into the equivalent SELCOPY/i commands, thus setting up an environment suitable for interactive execution of a SELCOPY batch job.

Where STEPLIB JCL statements are found in EXEC PGM=SELCOPY steps, the library names are included on a -LIB parameter in the generated, equivalent SELCOPY command.

SELCOPY Loop Break-in

The nature of SELCOPY 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 control deck.

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 eligible 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 within the SELCOPY job stream may be executed before a virtual break point is encountered and processing is paused.

This break-in threshold is controlled via the SELCOPY/i INI option **SELCOPY.LoopBreakIn** which has a default value of 1,000,000.

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 SELCOPY/i 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 CBLLe 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 CBLLe frame window (including SDE Edit). The SELCOPY Debug frame window is actually a CBLLe frame window with additional features and characteristics specifically relating to SELCOPY execution. These features are discussed in this section whereas details on CBLLe frame window features may be found in the [CBLLe Text Edit](#) documentation.

The SELCOPY Debug Main window must always contain the Control Cards, Output Listing and TRACE Windows. Closing any of these windows will quit the SELCOPY main window and so end the Debug session.

When a session is started, these 3 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 repositioned as in Figure 106.

```

SELCOPY
File View Go StepOver StepInto ReRun Window Help      wS wR
--SYSIN: CBL.SSC.CTL(SQ11525) 218 V PDSE Size=57 Alt=0,0;0
Command>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
* Now populate the table.
db2 sql="insert into sq11525 values ('Zero row ',0, 0, 0, 0,0, 0, 0, 0)"
db2 sql="insert into sq11525 values ('First row ',1,-1,-0.1,-5,1,-1,-0.01,-0.0
db2 sql="insert into sq11525 values ('Second row ',2, 2, 0.2, 6,2,-2, 0.20, 0.2
db2 sql="insert into sq11525 values ('Third row ',3,-3,-0.3,-7,3,-3,-0.30,-0.3
db2 sql="commit"

* Now print the table.
==readloop==
read indb      table='SQ11525' CHAR PFX SEP

if eof indb
  then goto drop
print type=mc
goto readloop

* Now drop the table.
==drop==
*db2 sql="drop table sq11525"
db2 sql="commit"

--SYSPRINT: NBJ3.SELCOPY.SQ11525.SYSPRINT
Command>
|...+....1....+....2....+....3....+....4..
 9.          then goto drop

10.        print type=mc
11.        goto readloop

--TRACE: NBJ3.SELCOPY.SQ11525.TRACE -+
Command>
|...+....1....+....2....+....3....+....4.
*** Top of File ***
** SELCOPY interactive execution started
*** End of File ***

Work Area
Command>
 1 40404040 40404040
 9 40404040 40404040
17 40404040 40404040
25 40404040 40404040
33 40404040 40404040
41 40404040 40404040
49 40404040 40404040
57 40404040 40404040
65 40404040 40404040
73 40404040 40404040
81 40404040 40404040
89 40404040 40404040
97 40404040 40404040

Te | Line=37 | Col=1 | Alt=0,0;0 | Size=57 | Recl=218

```

Figure 107. 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 CLI 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. LOCATE, TAG, ALL,

CHANGE, SET ZONE, etc.)

In addition to any CBL 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.

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 losing the SYSPRINT/SYSLST output from a previous run.

The SYSPRINT window is an edit view which supports execution of CBL 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.

```

SELCOPY - SYSPRINT: NBJ3.SELCOPY.SSDEM001.SYSPRINT 133 V SEQ Size=240
File View Go StepOver StepInto ReRun Window Help ws wr
Command> Scroll> Csr
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...
18.   if dir   pds2
      then    add 1 to totl at tot   type=b * +1 to total field.
19.   then if  pos marr, @arr+marre-1 = 8 at pdsin step=marre * Scan ar
19.   then add 1 to matl at mat   type=b * +1 to match field.
20.   then space 2 * Space 2 lines.
21.   then print from pdsin len 8 * Print matching member na
22.   else flag eom * Do not read data records
23.   then log from pdsin len 8 * Log mismatching member n
24.   then add 1 to unml at unml type=b * +1 to mismatch field.

25.   goto pdsloop * Get next record.
      *pdsloope*

==log_rtn==
-----
* .....1.....2.....3.....4.....5.....
26.   pos lstr = 'Total Members: xxx, Matching members: xxx, Mis-matching
27.   cvbc totl at tot   to lstr+15 fmt zz9
28.   cvbc matl at mat   to lstr+39 fmt zz9
29.   cvbc unml at unml to lstr+66 fmt zz9
30.   plog fr lstr len lstr
      *log_rtn*
31.   =ret=

INPUT  SEL  SEL
RECNO  TOT  ID.
-----
1074   1   9 ABND01  1  ABT01   2  ADA01   3  ADA02   4  ADA03   5  ADA04
      ADA09  ADA10  ADA11  ADDLIT  AMEQU  AMEX
      ARIT04  ARIT05  ARIT06  ARIT07  AT01   AT02

```

Figure 108. 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 CBL 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.)

```

SQLLOG: NBJ.SELCOPY.SQ11525.SQLLOG 133 V SEQ Size=77 Alt=0,0;78
Command> |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8...
000001 *** CBL Dynamic SQL Interface Version 2.02 AT ***
000002
000003 CBL$010I 16:58:52 CBL Dynamic SQL Interface is started. Date: 2009-01-14
000004
000005 CBL$000I 16:58:53 (Sel 1) Connected to DB2 Version 9.1.0
000006 Subsystem:DBGG Plan:CBPLAN0
000007 User:NBJ Current SQLID:NBJ
000008
000009 CBL$007I 16:58:55 (Sel 1) EXECUTE CREATE SQL Code=0
000010
000011 create table sq11525 ( char char(10) not null with default, dec1 dec(1,0)
000012 ) not null with default, dec2 dec(1,0) not null with default,
000013 dec3 dec(1,1) not null with default, int1 int not null with
000014 default, dec4 dec(2,0) not null with default, dec5 dec(2,0) not
000015 null with default, dec6 dec(2,2) not null with default, dec7 dec(
000016 3,3) not null with default, dec8 dec(4,3) not null with default,
000017 int2 int not null with default, dec9 dec(5,3) not null with
000018 default, decA dec(4,0) not null with default, decB dec(3,0) not
000019 null with default )
000020 DB2 CPU= 000000.264342 seconds.
000021
000022 CBL$007I 16:58:55 (Sel 2) EXECUTE COMMIT SQL Code=0
000023 DB2 CPU= 000000.009960 seconds.
000024
000025 CBL$007I 16:58:55 (Sel 3) EXECUTE INSERT SQL Code=0
000026
000027 insert into sq11525 values ('Zero row ',0, 0, 0, 0,0, 0, 0, 0, 0, 0, 0,
000028 0,0)
000029 Rows Inserted=1 DB2 CPU= 000000.069964 seconds.
000030
000031 CBL$007I 16:58:56 (Sel 4) EXECUTE INSERT SQL Code=0
000032
000033 insert into sq11525 values ('First row ',1,-1,-0.1,-5,1,-1,-0.01,-0.001,
000034 -5.001,-8,-0.001,4,4)
    
```

Figure 109. 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 ICGF 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 CBL 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.)

```

WTOLOG: NBJ.SELCOPY.S$DEMO01.WTOLOG 133 V SEQ Size=5 Alt=0,0;8
Command> |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8...
000000 *** Top of File ***
000001 Program about to start ----->
000002 About to enter loop.
000003 Exit loop.
000004 SELCOPY --- 'GOTO CANCEL' CONDITION MET.
000005 SELCOPY REL 2.02 SELECT TIME ERROR 513 JOB=NBJ 17.13 WED 14 JAN 2009
000006 *** End of File ***
    
```

Figure 110. 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, **SHOWPOPMENU**.

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.

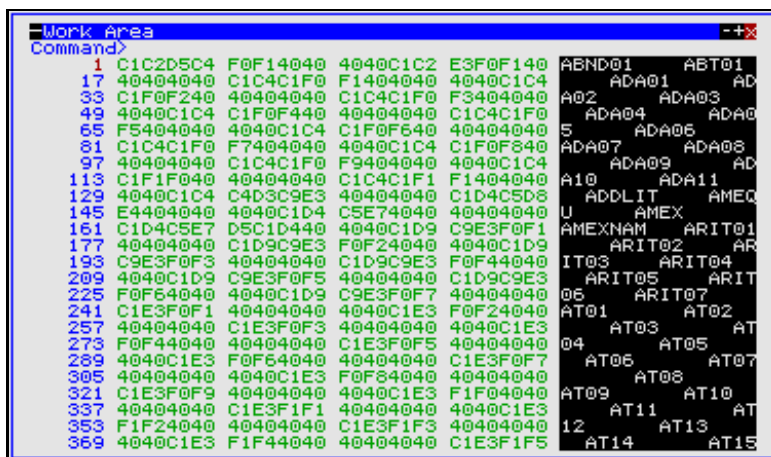


Figure 111. SELCOPY Work Area Window.

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, **SHOWPOPMENU**.

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 112. POS Window (inside work area)



Figure 113. 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 114. 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 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.

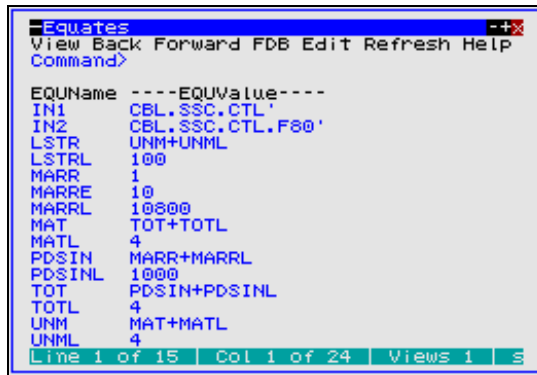


Figure 115. SELCOPY EQUates Window.

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.

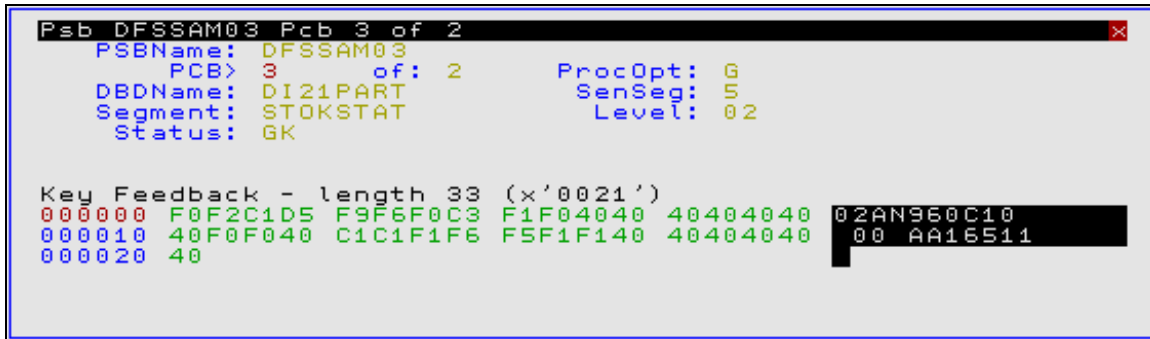


Figure 116. SELCOPY PCB Window.

TRACE Window

The TRACE window is opened automatically when SELCOPY Debug is started. It may also be 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.

```

TRACE: NBJ.SELCOPY.SSDEMO01.TRACE 133 V SEQ Size=22 Alt=0,0:23
Command>
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7
000006 ** SELCOPY interactive execution started 2009/01/14 17:43:47 ***
000007 1. log 'Program about to start'
000008 2. @arr = marr
000009 3. log 'About to enter loop.'
000010 if @arr > marr+marrl-1
000011 rd pds1 dsn=ini dir . into @arr * Directory reco
000012 pos @arr+8, @arr+lrecl-1 = . . * Blank rest of r
000013 if eof pds1
000014 else @arr = @arr+marr * Next input posi
000015 then goto memloop
000016 if @arr > marr+marrl-1
000017 rd pds1 dsn=ini dir . into @arr * Directory reco
000018 pos @arr+8, @arr+lrecl-1 = . . * Blank rest of r
000019 if eof pds1
000020 else @arr = @arr+marr * Next input posi
000021 print from marr,@arr+marr-1 * Print array for
000022 pos tot len=totl = 'x'00' fill '00' * Initialise to h
000023 * * * End of File * * *
    
```

Figure 117. SELCOPY TRACE Window.

Point-and-Shoot Popup Menu

All CBL type SELCOPY Debug windows, including the SYSIN/SYSIPT and SYSPRINT/SYSLST windows, support the point-and-shoot options popup menu. The popup menu is opened using the SDBPOPUP edit macro defined on 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.

```

Text-Edit Keys
Action> equ      omsg      lmsg+lmsgL...
Watch Pos "omsg"
----- "omsg" <edit>
Track    "omsg" <edit>
----- "omsg" <edit>

Track List

Break <toggle> (BR)

Restart (RR)
Run to next break-point (GO)
Immediate End-of-Job (EOJ)

Window Layout
    
```

Figure 118. SELCOPY Point-and-Shoot Popup Menu Window.

The menu enables the user to quickly and easily perform the following, commonly used tasks:

Text-Edit Keys/Debug Keys

Certain default PFKey assignments for CBL text edit differ to defaults set up for the SELCOPY Debug environment. This entry enables the user to toggle between the CBL (Edit) default keys and SELCOPY Debug (Debug) keys as follow:

PFKey	Edit	Debug
PF1	SOS LINEADD	STEPOVER
PF2	DUPLICATE	STEPINTO
PF13	SOS LINEDEL	GO
PF14	SPLTJOIN	BR ON CURSOR PERM

Action> *command_string*

Execute *command_string* as determined by text at the focus line and column. See the CBL **ACTION** facility for further information on how *command_string* is determined.

Watch Pos "*expression*"

Execute the SELCOPY Debug primary command **WINDOW POS *expr*** to open a **POS storage display window** starting at the position defined by expression. Show POS entries exist for POS expressions determined by the token on which focus was placed when the menu was opened. Possible expressions are:

1. The focus token that forms part of a larger positional expression.
2. If applicable, the full positional expression containing the focus token.

Watch Pos "*expression*" <edit>

Same as the **Show Pos** entry but instead of executing the **WINDOW POS *expr*** command directly, it is placed at the edit command prompt ready for edit and submission by the user.

Track "expression"

Invoke the **SDBTRACK** edit macro which issues the SELCOPY Debug primary command **TRACK expr** to start or stop tracking a position in storage referenced by the POS expression. Track entries exist for POS expressions determined as for "Show Pos".

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.

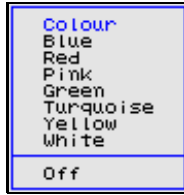


Figure 119. SELCOPY TRACK Colour Popup Menu Window.

Track "expression" <edit>

Same as the **Track** entry but instead the generated "SDBTRACK expression" macro invocation is placed at the edit command prompt ready for edit and submission by the user.

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.

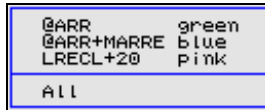


Figure 120. SELCOPY TRACK List Popup Window.

Break <toggle> (BR)

Toggle a break point on and off for the SELCOPY operation at the focus position.

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.

Run to next break-point (GO)

Allow the program to run until either a break point or End-of-Job is encountered.

Immediate End-of-Job (EOJ)

Force immediate End-of-Job.

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.

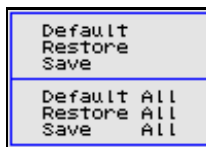


Figure 121. SELCOPY Window Layout Popup Menu.

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	PF Key
BREAKPOINT	Set/unset temporary break points.	PF14
EOJ	Force SELCOPY End-of-Job.	-
GO	Continue processing.	PF13
RERUN	Re-run from the beginning.	-

STEPINTO	Step (Trace) Into sub-routines.	PF2
STEPOVER	Step (Trace) Over sub-routines.	PF1
TRack	Track a SELCOPY POS expression.	-
Window	Open a specified SELCOPY Debug window.	-

See **SELCOPY Debug Function Keys** for complete list of default PFKeys.

BREAKPOINT

Syntax:

```
>>-- BReakpoint -----><
```

Description:

Use the BREAKPOINT command to set or unset a permanent BReak point at the focus statement in the **Control Card window**. By default, BREAKPOINT is assigned to PF06.

If a break point is set at a particular control statement, then processing will be paused on the next attempt to execute that statement.

Any number of concurrent break points may be active during job execution.

Parameters:

BREAKPOINT has no parameters.

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.

Parameters:

EOJ has no parameters.

GO

Syntax:

```
>>-- GO -----><
```

Description:

Use the GO command to continue processing of the control statements. By default, function key **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.

Parameters:

GO has no parameters.

RERUN

Syntax:

```
>>--+ RErun --+-----><
      |         |
      +- RR -----+
```

Description:

Use the RERUN command to Re-Run the job from the beginning. No further statements will be executed from the existing job run.

Storage is cleared and values and break points, set during control statement analysis, are re-initialised. All existing breakpoints are cleared.

Parameters:

RERUN has no parameters.

STEPINTO

Syntax:

```
>>--+ STEPIntO--+-----><
      |         |
      +- SI -----+
```

Description:

Use the STEPINTO command to step through the SELCOPY control statements logically one at a time. By default, STEPINTO is assigned to PF02.

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:

STEPINTO has no parameters.

STEPOVER

Syntax:

```
>>--+ STEPOver--+-----><
      |         |
      +- SO -----+
```

Description:

Use the STEPOVER command to step through the SELCOPY control statements logically one at a time. By default, STEPOVER is assigned to PF01.

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:

STEPOVER has no parameters.

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.

WINDOW

Syntax:

```

>>-- Window --+--- @ -----+-----><
               |             |
               +--- AT-----+
               |             |
               +--- Ctl  -----+
               |             |
               +--- EQUates -----+
               |             |
               +--- List  -----+
               |             |
               +--- PCB  -----+
               |             |
               +--- POS expr -----+
               |             |
               +--- SQL  -----+
               |             |
               +--- Workarea -----+
               |             |
               +--- WTO  -----+
               |             |
               +--- TRace -----+

```

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 Function Keys

You can assign 3270 Program Function Keys (PFKeys) to line commands as applicable to the window class. The **KEYS** line command may be used to display and globally assign Function key settings.

SELCOPY Debug PFKeys have default functions assigned as determined by the window's class (i.e. Edit, Storage or List.) In addition to these, edit views and storage windows have the following PFKey definitions:

PF1	StepOver	Execute the next SELCOPY operation (step over a sub-routine.)
-----	----------	---

PF2	StepInto	Execute the next SELCOPY operation (step into a sub-routine.)
F16	Macro SdbPopup	Invoke the SDBPOPUP edit macro to display a functions menu (edit views only.)
PF5	ShowPopupMenu	Open the storage display options popup menu (storage display windows only.)
PF13	Go	Continue SELCOPY processing.
PF14	BreakPoint	Toggle a break point on and off at the focus operation (edit views only.)

For edit views only, the default action of the PF1/PF2/PF13/PF14 keys may be governed via the SdbPopup macro which allows the user to select either "Edit" or "Debug" PFKey configuration.

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 execution
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 Datasets/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

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 **CBL 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 **CBL 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 characteristics defined by the list window class. For example, the Execute CBLVCAT window supports **Prefix Commands** and filtering, to display new views of the data.

```

SELCOPY/i - Execute CBLVCAT
View Refresh Back Forward FDB Raw Text Help
Command>
VCAT Command> LISTVCAT KEY=CBL
VCAT Program> CBLV
-----sysPrint-----
1CBLVCAT REL 3.10 AT CBL - Bridgend UK (Internal Only)
OS JOB=NBJ2 10.25 FRI 13 APR
-----
LISTVCAT KEY=CBL
-----
ICF CAT CBLMCT (3390) TYPE NRECS PCNT ---- ALLOC TRACKS ---- FRSP LMAX KL,RKP CISIZE BUFSP EXCPS
----- TOTAL PRIME SEC CI CA ---- /BLK/IMB ---- /IXL ----
CBL.ACS.TRAN.LST NONVSAM VOL1=CBLM09 3390 201
CBL.ADCD.CBLI.CMX NONVSAM VOL1=CBLM03 3390 200
CBL.ADCD.TEST NONVSAM VOL1=CBLM06 3390 200
CBL.AIRPORTS.BIN NONVSAM VOL1=CBLM07 3390 200
CBL.AIRPORTS.CSV NONVSAM VOL1=CBLM08 3390 200
CBL.AM.G1465.TXT NONVSAM VOL1=CBLM05 3390 201
CBL.AM.G1621.TXT NONVSAM VOL1=CBLM07 3390 201
CBL.AM.G1645.TXT NONVSAM VOL1=CBLM10 3390 201
CBL.AM.LOAD NONVSAM VOL1=CBLM04 3390 199
CBL.AM.LOAD.SQ10152 NONVSAM VOL1=CBLM04 3390 199
CBL.AMALL.DA NONVSAM VOL1=CBLM02 3390 200
CBL.AMALL.EBCDIC.DA NONVSAM VOL1=CBLM07 3390 200
CBL.AMALL.EBCDIC.DA.KSDS KSDS(R) 4096 **97.9** C=31 C=27 C=4 5700 7,25 18432 38912 20.2K 200
VOL1=CBLM08
IX 32 65.4 1 1 1 505 512 IXL=2 2542
VOL1=CBLM08
CBL.AMALL.G1465.DA NONVSAM VOL1=CBLM08 3390 200
CBL.AMCUST.G1465.DA NONVSAM VOL1=CBLM07 3390 200
CBL.AMCUST.G1516.DA NONVSAM VOL1=CBLM02 3390 200
CBL.AMCUST.G1586.DA NONVSAM VOL1=CBLM07 3390 201
CBL.AMCUST.G1621.DA NONVSAM VOL1=CBLM10 3390 201
CBL.AMCUST.G1645.DA NONVSAM VOL1=CBLM10 3390 201
CBL.AMCUST.G1647.DA NONVSAM VOL1=CBLM11 3390 201
CBL.AMEX.CTL NONVSAM VOL1=CBLM03 3390 201
CBL.AMEX.EXE.XMIT.BIN NONVSAM VOL1=CBLM09 3390 201
CBL.AMEX.JCL NONVSAM VOL1=CBLM06 3390 201
CBL.AMSUPP.DA NONVSAM VOL1=CBLM08 3390 200
CBL.AMSUPP.DA.COPY NONVSAM VOL1=CBLM08 3390 200
CBL.AMSUPP.DA.RRDS RRDS 496 ** ALL** 62 1 1*5 5700 6144 12288 275 200
VOL1=CBLM07
CBL.AON#US.COPYBOOK.COBOL NONVSAM VOL1=CBLM08 3390 201
CBL.AON#US.COPYBOOK.COBOL.XMIT.BIN NONVSAM VOL1=CBLM02 3390 201
CBL.AON#US.DATA.ADFNSL01 NONVSAM VOL1=CBLM06 3390 201
CBL.AON#US.DATA.ADFNSL01.F1000 NONVSAM VOL1=CBLM09 3390 201
Line 1 of 6114 Col 1 of 135 Views 1 select *
    
```

Figure 122. 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.

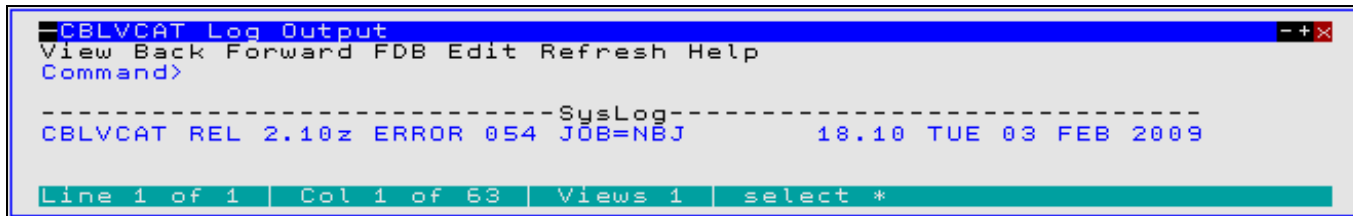


Figure 123. CBLVCAT Log Output window.

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 CBLLe 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 CBLLe 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 CBLLe 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 PF2 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.

```

SELCOPY/i - CBLVCAT Raw: listcat key=nbj type=c 2011/12/09 1
View Refresh Back Forward FDB Text Help wS wR Scroll> Csr
Command>
VCAT Command> listcat key=nbj type=c
-----DSN----- --TYPE-- --NRECS-- --PCNT--
NBJ.CBL.EMP.D2010190.KSDS KSDS (R) 1 0.1
IX 1 2.1
NBJ.CBLIDEMO.KSDS KSDS 403 56.0
IX 3 25.0
NBJ.CBLIDEMO.V0000.KSDS KSDS 500 24.9
IX 13 39.4
NBJ.CBLINST.CBL11091.CSI KSDS 48953 11.8
IX 6 0.5
NBJ.CBLINST.CBL11091.SILOG KSDS 2 0.1
IX 1 2.1
NBJ.DATASET.BASIC01.KSDS KSDS (R) 42 24.6
IX 1 2.1
NBJ.DATASET.COPY.KSDS KSDS (R) 42 24.6
IX 1 2.1
NBJ.DATASET.KSDS KSDS (R) 42 29.2
IX 1 2.1
NBJ.DDIR KSDS (R) 44 0.7
IX 3 25.0
NBJ.EXT.STR4.KSDS KSDS 40 3.5
IX 1 3.1
NBJ.FSU.D2010004.T145611 ESDS (R) 34 8.4
NBJ.FSU.D2010238.T170652 ESDS (R) 4502 **98.1**
NBJ.FSU.D2010238.T180454 ESDS (R) 4502 **98.6**
NBJ.FSU.D2010298.T151027 ESDS (R) 6 8.4
NBJ.FSU.D2010298.T151124 ESDS (R) 6 8.4
NBJ.FSU.D2010298.T115456 ESDS (R) 80 25.0
NBJ.FSU.TEST.ESDS ESDS (R) 16 4.2
NBJ.FSU.TEST.KSDS KSDS 403 83.0
IX 10 20.5
NBJ.FSU.TEST.KSDS2 KSDS (R) 500 **99.3**
IX 25 75.8
NBJ.KSDS KSDS (R) 1 0.2
IX 1 2.1
NBJ.SELCOPY.DEMO.KSDS KSDS (R) 0 ( 696)
IX
NBJ.ZZSDB2B.ESDS ESDS (R) 153 25.0
NBJ.ZZSDB2B.KSDS KSDS (R) 0 ( 690)
Line 1 of 45 | Col 1 of 589 | Views 1 | select * sort DSN
    
```

Figure 124. CBLVCAT LISTVCAT Raw Data Window.

```

SELCPY/i - CBLVCAT Raw: listvtoc vol=cblm08 2011/12/
View Refresh Back Forward FDB Text Help wS wR
Command>
VCAT Command> listvtoc vol=cblm08
----- DSN-----
CBL.AIRPORTS.CSV 050/08 050/08
CBL.AMALL.EBCDIC.DA.KSDS.DATA 4593/00 4623/14
CBL.AMALL.EBCDIC.DA.KSDS.INDEX 023/00 023/00
CBL.AMALL.G1465.DA 4381/00 4464/14
CBL.AMSUPP.DA 082/00 082/14
" 081/00 081/14
" 079/00 080/14
CBL.AMSUPP.DA.COPY 018/00 018/14
" 019/00 019/14
" 016/00 017/14
CBL.AON#US.COPYBOOK.COBOL 381/00 381/14
" 382/00 382/14
" 380/00 380/14
" 233/00 233/14
" 215/00 215/14
CBL.APAR.OA06896 039/00 039/14
CBL.APAR.OA06896.FTP 011/05 011/05
CBL.APAR.OA06896.PTF.PACKED 034/00 038/14
CBL.APAR.OA13742.PTF.PACKED 383/00 550/14
CBL.APAR.OA13742.TXT 012/13 012/13
" 012/08 012/12
CBL.APAR.UA37111 4514/08 4514/13
CBL.APAR.UA37111.FTP 069/07 069/07
CBL.APAR.UA37111.LST 4985/04 4986/08
CBL.APAR.UA37111.TXT 4514/07 4514/07
" 4514/02 4514/06
CBL.APAR.UK27934.PTF.PACKED 9722/00 9889/14
CBL.BA.CBLI.SYSTEM.INI 013/00 013/00
CBL.BA.EXE 5065/04 5065/08
" 051/06 051/10
" 5092/00 5092/04
" 5081/09 5081/13
" 5065/09 5065/13
" 5093/00 5093/04
" 5092/10 5092/14
" 5092/05 5092/09
" 050/10 051/01
Line 1 of 813 Col 1 of 199 Views 1 select * sort DSN

```

Figure 125. 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, using the entry name as the Audit DSN field 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 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 CBL 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 PF2 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 CBI VCAT severity message level
VOLUME	Char	Catalog Volume
GMAX	Char	GDG Maximum Level
GVER	Char	GDG Version number
GEN	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 **CBL 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 characteristics as List windows. For example filtering is supported to display new views of the data.


```

CBL - IDCAMS Command: LISTCAT ALL ENTRY(CBL.CBLI.MBRLIST.KSDS.CMP)
View Back Forward FDB Edit Refresh Help          wS wR
Command>
AMSCCommand> LISTCAT ALL ENTRY(CBL.CBLI.MBRLIST.KSDS.CMP)
>
Asa -----Line-----
1 IDCAMS SYSTEM SERVICES
0 MARGINS(1 32760)
0 IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
0
LISTCAT ALL ENTRY(CBL.CBLI.MBRLIST.KSDS.CMP)
0 CLUSTER ----- CBL.CBLI.MBRLIST.KSDS.CMP
  IN-CAT --- USERCAT.CBLCAT
  HISTORY
  DATASET-OWNER----- (NULL)          CREATION-----2008.255
  RELEASE-----2          EXPIRATION-----0000.000
  SMSDATA
  STORAGECLASS ----CBLDFLT          MANAGEMENTCLASS--CBLDFLT
  DATACLASS -----CBLXACMP          LBACKUP ---0000.000.0000
  BWO STATUS-----00000000          BWO TIMESTAMP---000000 00:00:00.0
  BWO----- (NULL)
  RLSDATA
  LOG ----- (NULL)          RECOVERY REQUIRED --(NO)          FRLOG -
  VSAM QUIESCED ----- (NO)          RLS IN USE ----- (NO)
0 LOGSTREAMID----- (NULL)
  RECOVERY_TIMESTAMP LOCAL-----X'0000000000000000'
  RECOVERY_TIMESTAMP GMT-----X'0000000000000000'
  PROTECTION-PSWD----- (NULL)          RACF----- (NO)
  ASSOCIATIONS
  DATA----- CBL.CBLI.MBRLIST.KSDS.CMP.DATA
  INDEX----- CBL.CBLI.MBRLIST.KSDS.CMP.INDEX
0 DATA ----- CBL.CBLI.MBRLIST.KSDS.CMP.DATA
  IN-CAT --- USERCAT.CBLCAT
  HISTORY
  DATASET-OWNER----- (NULL)          CREATION-----2008.255
  RELEASE-----2          EXPIRATION-----0000.000
  ACCOUNT-INFO----- (NULL)
  PROTECTION-PSWD----- (NULL)          RACF----- (NO)
  ASSOCIATIONS
  CLUSTER--CBL.CBLI.MBRLIST.KSDS.CMP
  ATTRIBUTES
  KEYLEN-----15          AVGLRECL-----256          BUFSPAC
Line 1 of 118 | Col 1 of 127 | Views 1 | select *
    
```

Figure 126. IDCAMS Command window.

Panel Fields

AMSCCommand>
Specify valid IDCAMS command syntax.

Prefix Commands

No prefix line commands are supported for IDCAMS Command windows.

Columns Displayed

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 **CBL**e 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 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.

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.

```

POWER Command Output
View Back Forward FDB Edit Refresh Help
Command>
POWER Command> D LST
JobName- Number Sfx Q Sys Pr Disp Cl Cards -Pg- Cc --Form-- ---To--- --From
BASEREST      7 0 L 3 H A 150 7 1 SYSA SYSA
CATVTAM     151 0 L 3 H A 24 4 1 SYSA SYSA
CBLCATL     786 0 L 3 H A 3 1 1 SYSA SYSA
CBLCATL     787 0 L 3 H A 88 5 1 SYSA SYSA
CBLCATL     788 0 L 3 H A 3 1 1 SYSA SYSA
CBLDEFS     785 0 L 3 H A 11 3 1 SYSA SYSA
CBLIVTAM     819 0 L 3 H A 19 2 1 SYSA SYSA
CBLIVTAM     820 0 L 3 H A 13 2 1 SYSA SYSA
CBLLINK     789 0 L 3 H A 715 27 1 SYSA SYSA
CBLLOAD     810 0 L 3 H A 2381 38 1 SYSA SYSA
CBLNAMEA     809 0 L 3 H A 719 21 1 SYSA SYSA
CBLVVJ07     815 0 L 3 H A 104 4 1 SYSA SYSA
CEEWARC    1177 0 L 3 D A 11 2 1 SYSA SYSA
CICSICCF     920 0 L 3 H A 317 10 1 SYSA SYSA
CICSICCF    1019 0 L 3 H A 308 10 1 SYSA SYSA
CICSICCF    1038 0 L 3 H A 349 11 1 SYSA SYSA
CICSICCF    1057 0 L 3 H A 2178 38 1 SYSA SYSA
CICSICCF    1112 0 L 3 H A 250 9 1 SYSA SYSA
CICSICCF    1118 0 L 3 H A 250 9 1 SYSA SYSA
Line 1 of 142 | Col 1 of 338 | Views 1 | select * sort JobName,Number,Sfx

```

Figure 127. 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 CBL e 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: <ul style="list-style-type: none"> • System.TrustedUser=Yes in the SELCOPY/i INI file. • System.VSESMLogon=Yes in the SELCOPY/i 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)
PXFMRLN	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	UIn	.. SHARED SYSID 4 ACC. CNT.
PXFMMAC5	UIn	.. SHARED SYSID 5 ACC. CNT.
PXFMMAC6	UIn	.. SHARED SYSID 6 ACC. CNT.
PXFMMAC7	UIn	.. SHARED SYSID 7 ACC. CNT.
PXFMMAC8	UIn	.. SHARED SYSID 8 ACC. CNT.
PXFMMAC9	UIn	.. 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 **CBL 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.

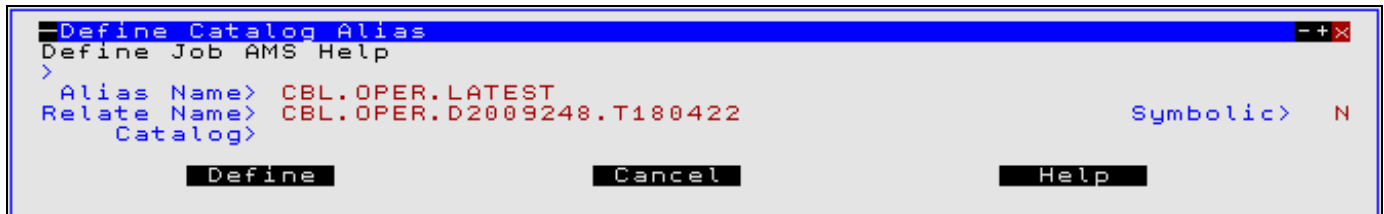


Figure 128. Define Catalog ALIAS Dialog window.

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 CBL command **SUBMIT**.) (Background)

AMS

Opens a CBL 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**.

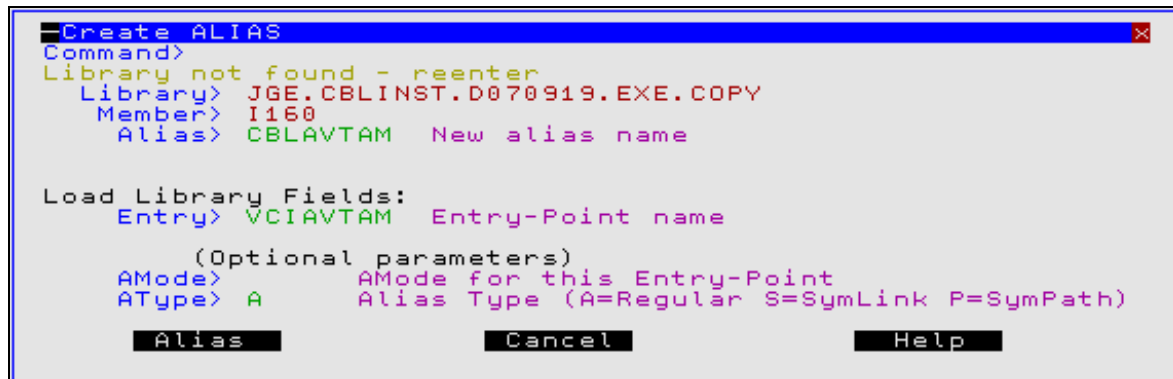


Figure 129. Create Library ALIAS Dialog window.

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 aliases 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 "JOB CARD" 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.

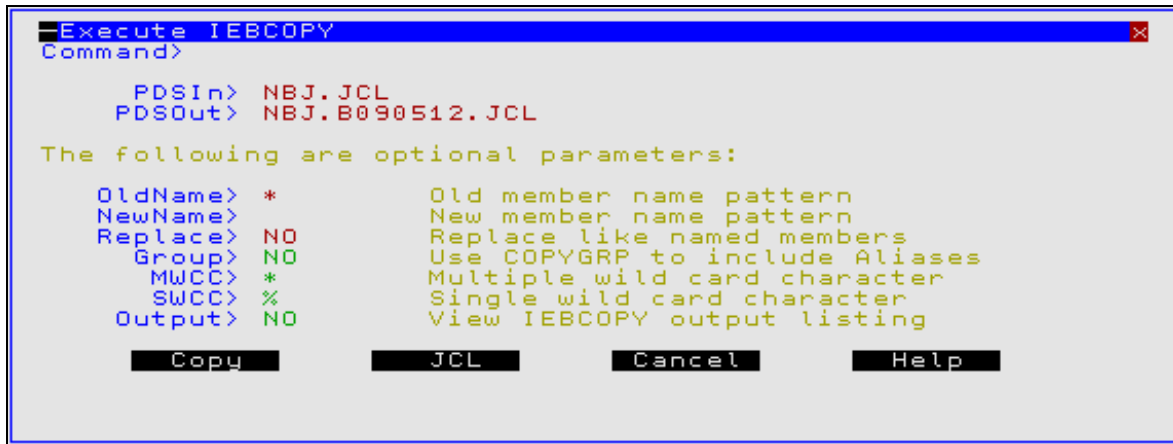


Figure 130. IEBCOPY Dialog window.

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".

DB2 Dynamic SQL

The DB2 Dynamic SQL window is a **list window**, opened on execution of the **SQL** command.

DB2 Dynamic SQL window may be used to submit SQL commands to a DB2 subsystem and display the resultant messages and/or table views.

Note that SQL statements may also be executed via the **Execute SQL Statements** panel as part of the SELCOPY/i DB2 interface.

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 **CBLNAME** load module.

```

CBL - Dynamic SQL: DB9G
View Back Forward FDB Edit Refresh Log Help          wS wR
Command>
DB2 Subsystem> DB9G                                Plan> CBLPLAN0
Select Limit> 200                                  AutoCommit> Y
SQL Statement> SELECT * FROM SYSIBM.SYSCOLUMNS WHERE COLTYPE='CHAR' ORDER BY
> NAME
>
-----NAME-----      -----TBNAME-----      TBCREATOR  COLNO-  COLTYPE-  LENG
-----
ACCESSPATH              SYSPACKSTMT              SYSIBM     14  CHAR
ACCESSPATH              SYSSTMT                  SYSIBM     11  CHAR
ACCESSPATH              SYSPACKSTMT              SYSIBMV    14  CHAR
ACCESSPATH              SYSSTMT                  SYSIBMV    11  CHAR
ACCESSTYPE              PLAN_TABLE               SYSIBM     10  CHAR
ACCESSTYPE              PLAN_TABLE               DB20SC     10  CHAR
ACCESSTYPE              PLAN_TABLE               DSN8910    10  CHAR
ACCESSTYPE              PLAN_TABLE               CLARKG     10  CHAR
ACCOUNTING              DB2_THREAD_STATUS       SYSIBM     19  CHAR
ACCTNO                  SUPPLIER                 Q          1  CHAR
ACQUIRE                SYSPLAN                  SYSIBM     12  CHAR
ACQUIRE                SYSIBMV                  SYSIBMV    12  CHAR
ACTION                  TOPTVAL                  DSN8910    2  CHAR
ACTION                  VOPTVAL                  DSN8910    2  CHAR
ACTION_IND              ADBCHK                   IBMUSER    8  CHAR
ACTIVE                  SYSROUTINES              SYSIBM     90  CHAR
ACTIVE                  DB2_THREAD_STATUS       SYSIBM     5  CHAR
ACTIVE                  SYSROUTINES              SYSIBMV    90  CHAR
ACTKWD                  EACT                     DSN8910    2  CHAR
ACTKWD                  ACT                       DSN8910    2  CHAR
ACTKWD                  VACT                     DSN8910    2  CHAR
ADDED_PRED              DSN_PREDICAT_TABLE      SYSIBM     20  CHAR
ADDED_PRED              DSN_PREDICAT_TABLE      DB20SC     20  CHAR
ADMRDEPT                DEPT                     DSN8910    4  CHAR
ADMRDEPT                VDEPT                    DSN8910    4  CHAR
ADMRDEPT                VHDEPT                   DSN8910    4  CHAR
ADMRDEPT                VDEPMG1                  DSN8910    7  CHAR
ADMRDEPT                EDEPT                    DSN8910    4  CHAR
ADMRDEPT                NEWDEPT                  DSN8910    4  CHAR
ADMRDEPT                DEPT                     FMNDB2     4  CHAR
ADMRDEPT                VDEPT                    FMNDB2     4  CHAR
ADMRDEPT                VHDEPT                   FMNDB2     4  CHAR
Line 1 of 200 | Col 1 of 468 | Views 1 | select *

```

Figure 131. DB2 Dynamic SQL window.

The contents of the display area may be edited in a new file using the CBL 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.


```

CBL - SQL log
View Back Forward FDB Edit Refresh Help
Command>

1          *** CBL Dynamic SQL Interface Version V11.1.0 ***
CBL0010I 17:03:38 CBL Dynamic SQL Interface is started. Date: 2009-02-03
CBL0000I 17:03:38 (Sel 0) Connected to DB2 Version 9.1.0
                  Subsystem:DB9G          Plan:CBLPLAN0
                  User:NBJ             Current SQLID:NBJ

CBL0004I 17:03:38 (Sel 1) OPEN  SELECT cursor SQL00001
          SELECT *
          FROM SYSIBM.SYSCOLUMNS
          WHERE COLTYPE='CHAR'
          ORDER BY NAME

CBL0006I 17:03:42 (Sel 1) CLOSE SELECT cursor SQL00001
          Note: End of SELECT not reached.
          Prepare          DB2 CPU= 000000.104791 seconds.
          OPEN             DB2 CPU= 000002.490252 seconds.
          Rows Fetched=200  Total DB2 CPU= 000000.283559 seconds.
                           DB2 CPU= 000002.878602 seconds.

CBL0007I 17:03:42 (Sel 2) EXECUTE COMMIT SQL Code=0
                           DB2 CPU= 000000.008171 seconds.

No list view yet, enter parameters

```

Figure 132. 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.

Name	Type	Key	Offset	Length	Title	DispLen	Digits	Scale
ALTEREDTS	Char	No	7151	26	ALTEREDTS	26	0	
CCSID	Int	No	7177	4	CCSID	11	0	
COLCARD	Int	No	405	4	COLCARD	11	0	
COLCARDF	Float	No	6840	8	COLCARDF	24	0	
COLNO	Int	No	390	2	COLNO	6	0	
COLSTATUS	Char	No	6848	1	COLSTATUS	1	0	
COLTYPE	Char	No	392	8	COLTYPE	8	0	
CREATEDTS	Char	No	7121	26	CREATEDTS	26	0	
DATATYPEID	Int	No	6853	4	DATATYPEID	11	0	
DEFAULT	Char	No	5179	1	DEFAULT	1	0	
DEFAULTVALUE	VChar	No	5302	1536	DEFAULTVALUE	1	0	
FLDPROC	Char	No	5183	1	FLDPROC	1	0	
FOREIGNKEY	Char	No	5182	1	FOREIGNKEY	1	0	
HIDDEN	Char	No	7181	1	HIDDEN	1	0	
HIGH2KEY	VChar	No	409	2000	HIGH2KEY	8	0	
IBMREQD	Char	No	4414	1	IBMREQD	1	0	
KEYSEQ	Int	No	5180	2	KEYSEQ	6	0	
LABEL	VChar	No	5184	90	LABEL	0	0	
LENGTH	Int	No	400	2	LENGTH	6	0	
LENGTH2	Int	No	6849	4	LENGTH2	11	0	
LOW2KEY	VChar	No	2411	2000	LOW2KEY	8	0	
NAME	VChar	No	0	128	NAME	18	0	
NULLS	Char	No	404	1	NULLS	1	0	
PARTKEY_COLSEQ	Int	No	7148	2	PARTKEY_COLSEQ	6	0	
PARTKEY_ORDERING	Char	No	7150	1	PARTKEY_ORDERING	1	0	
RELCREATED	Char	No	7182	1	RELCREATED	1	0	
REMARKS	VChar	No	4415	762	REMARKS	26	0	
SCALE	Int	No	402	2	SCALE	6	0	
SOURCETYPEID	Int	No	6857	4	SOURCETYPEID	11	0	
STATS_FORMAT	Char	No	7147	1	STATS_FORMAT	1	0	
STATSTIME	Char	No	5276	26	STATSTIME	26	0	
TBCREATOR	VChar	No	260	128	TBCREATOR	8	0	
TBNAME	VChar	No	130	128	TBNAME	26	0	
TYPENAME	VChar	No	6991	128	TYPENAME	8	0	
TYPESHEMA	VChar	No	6861	128	TYPESHEMA	8	0	
UPDATES	Char	No	4413	1	UPDATES	1	0	

No list view yet, enter parameters

Figure 133. DB2 SQL FDB Window.

Panel Fields

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.

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 migration 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 **CBL 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.

```

FAV - Favourite Datasets/Commands
File Help
Command>
ZZSFAV00
Command/Func :
DSN/Variable #: 11 Project: CBL
Member Name : GENDAT01 Group : PL1
Max Entries : 99 Scroll-H: 1 Columns: 2 Type : SRC

1: HOME 2:
3: CBL.JCL 4: NBJ.JCL
5: CBL.SSC.CTL 6: NBJ.SSC.CTL
7: CBL.SSC.LST 8: NBJ.SSC.LST
9: CBL.PL1.COPYBOOK 10: CBL.PL1.SRC
11: CBL.PL1.JCL 12: CBL.PL1.OUT
13: CBL.PL1.LST 14:
15: 16:
17: LVOL CBLM* 18: /u/smpe/smpnts
19: LJQ NBJ 20: /u/cbl/nbj
21: LQ SYSDSN 22:
23: 24:
25: 26:
  
```

Figure 134. 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 field 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 SELCOPY/i User INI file variables, list item values can be redisplayed across SELCOPY/i 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 manageable.

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.

```

Operating System
LPA LinkList APFList Tasks Storage Programs Help
Command>

      Operating system:  z/OS
      Operating system release:  1.9.0
      VM guest machine id:
      CPU serial number:  0A000048
      CPU model:  1247
      Number of online CPUs:  2
      System name:  ADCCD
      Nucleus name:  IEANUC01
      Primary JES name:  JES2
      Master catalog:  CATALOG.Z19.MASTER
      DFP release:  3.3.2
      VTAM release:  6.1.9
      RACF release:  7.74.0
      RACF dataset name:  SYS1.RACFDS
      VM guest:  N
      Real storage (megabytes):  1536
      Expanded storage (megabytes):  0
      Maximum address spaces:  355
      PRISM indicator:  N
      GTF active:  N
      RACF active:  Y
      IPL volume:  Z9RES1
      IPL unit:  0A80
      Last IPL date:  09/02/03 15:13
      RACF user id:  NBJ
    
```

Figure 135. 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.

```

LPA Modules
View Back Forward FDB Edit Refresh Help
Command>

Address-  -Chain--  --RBP---  Dyn  --Name--  --EPA---  --MjP---  -Use-  At0  SSP  At1  At2
80CB4000  00000000  00000000  N    IFG0239I  00E07000  00C0BC5C0  0  18  0  B5  12
00CB4028  00000000  00000000  N    FLMS7C   850BD0F0  000000000  0  18  0  B1  22
00CB4050  00CB4000  00000000  N    IVTSMCES 84229798  00CBB648  0  18  0  B5  12
00CB4078  00000000  00000000  N    IGWAMCS1 85421D58  000000000  0  18  0  B1  22
00CB40A0  00CB4190  00000000  N    IKJT441R 83FDDBD8  00CC2010  0  18  0  B5  12
00CB40C8  00CB40A0  00000000  N    IXGINVR  8315FD80  000000000  0  18  0  B1  22
00CB40F0  00000000  00000000  N    IGWAMCS2 858F9480  000000000  0  18  0  B1  22
00CB4118  00CB41E0  00000000  N    CEL4CTBL 84C52000  000000000  0  18  0  B1  22
00CB4140  00000000  00000000  N    IGWAMCS4 83FAF680  000000000  0  18  0  B1  22
Line 1 of 2115 | Col 1 of 125 | Views 1 | select *
    
```

Figure 136. 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)

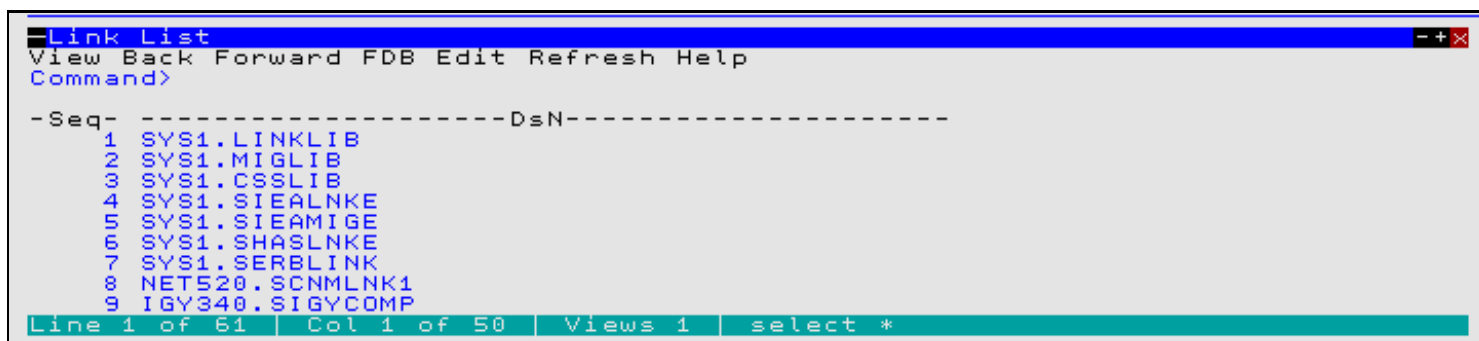
The Link List window may be opened via the following:

- 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.



```

Link List
View Back Forward FDB Edit Refresh Help
Command>

-Seq- -----DsN-----
 1 SYS1.LINKLIB
 2 SYS1.MIGLIB
 3 SYS1.CSSLIB
 4 SYS1.SIEALNKE
 5 SYS1.SIEAMIGE
 6 SYS1.SHASLNKE
 7 SYS1.SERBLINK
 8 NET520.SCNMLNK1
 9 IGY340.SIGYCOMP

Line 1 of 61 | Col 1 of 50 | Views 1 | select *

```

Figure 137. 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.

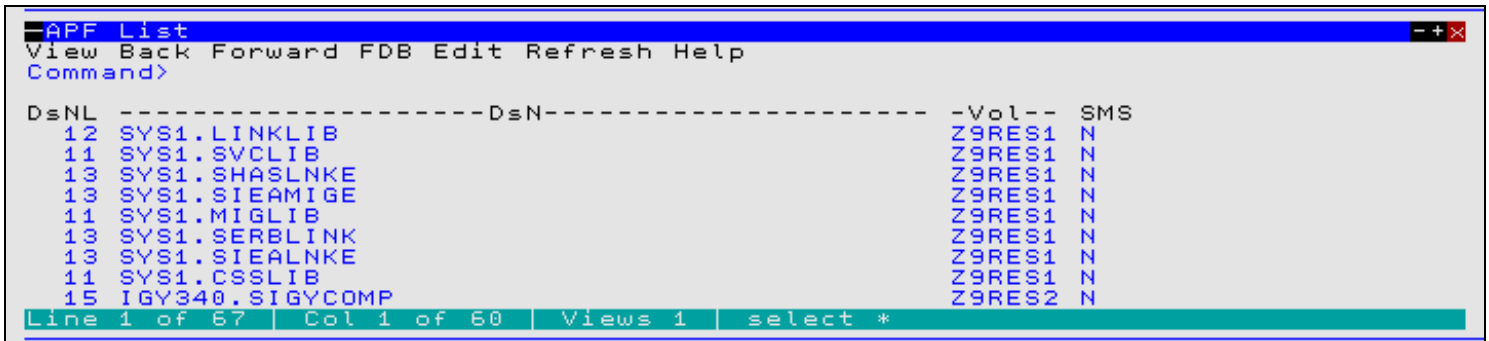


Figure 138. APF List window.

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.

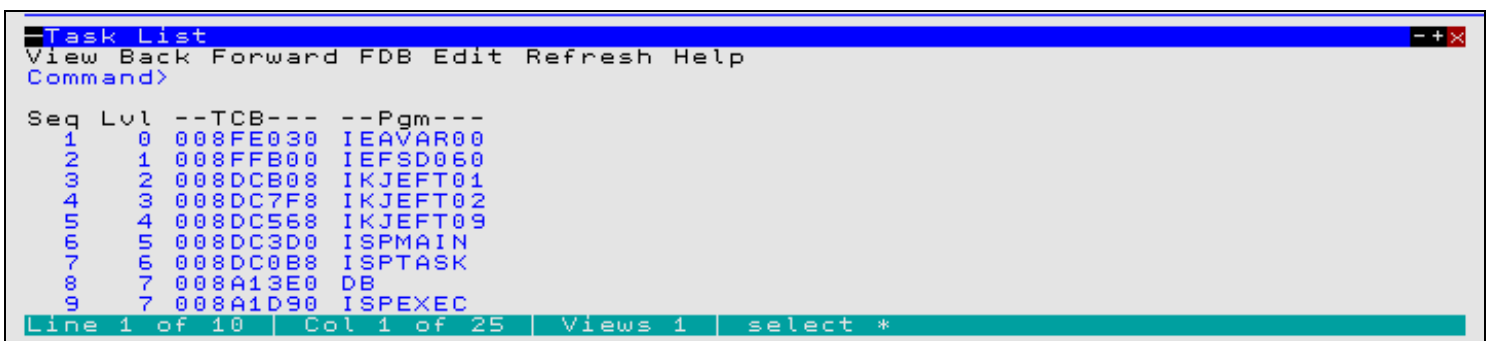


Figure 139. Task List window.

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.

```

Allocated storage
View Back Forward FDB Edit Refresh Help
Command>

Type --TCB--- Pool Key Flags Status Address- -Length-
PVT 008A13E0 0 8 08 Alloc 00064000 00001000
PVT 008A13E0 0 8 08 Free 00064000 00000CD8
PVT 008A13E0 0 8 08 Free 00064D70 00000098
PVT 008A13E0 0 8 08 Alloc 00065000 00002000
PVT 008A13E0 0 8 08 Free 00066620 000001C0
PVT 008A13E0 0 8 08 Alloc 00067000 00001000
PVT 008A13E0 0 8 08 Alloc 00068000 00009000
PVT 008A13E0 0 8 08 Alloc 0007C000 00001000
PVT 008A13E0 0 8 08 Alloc 0007D000 00009000
Line 1 of 196 | Col 1 of 53 | Views 1 | select *

```

Figure 140. 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.

```

Loaded Programs
View Back Forward FDB Edit Refresh Help
Command>

--Name--  --EPA---  Address-  ---Size---  SizeHex-  -Use-  AMode31  Auth  Rent  CDLoad
BPXWREXX  1E110F68  1E10C338  122056  0001DCC8  2 Y     N     Y     N
BPXWREXX  1E110F68  1E10C338  122056  0001DCC8  2 Y     N     Y     N
CBLDLL    1E3C4000  1E3C4000  929792  000E3000  1 Y     N     Y     N
CBLI      1E36B000  1E36B000  167936  00029000  1 Y     N     Y     N
CBLXREXX  1E12A9F0  1E12A9F0  512     00000200  1 Y     N     Y     N
DB        1E217000  1E217000  1032192 000FC000  1 Y     N     N     N
DBXXREXX  1E12ABF0  1E12ABF0  512     00000200  1 Y     N     Y     N
EAGRTPRC  044112B8  00000000  0       00000000  0 Y     N     Y     N
EAGRTXIN  0442D858  00000000  0       00000000  0 Y     N     Y     N
Line 1 of 70 | Col 1 of 108 | Views 1 | select *

```

Figure 141. 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.

SELCOPY/i Storage Statistics Window

The Storage Statistics window may be opened via the following:

- Select 'SELCOPY/i storage stats' from the Utilities/System menu in the **CBL** main window menu bar.
- Enter command **SYSSTOR** on the command line of any window.

The Storage Statistics window displays storage being used by SELCOPY/i at that moment in time. The values in each field will vary as windows are opened and closed.

The storage allocated by SELCOPY/i 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 SELCOPY/i, the area within the storage block occupied by that element, is freed.

If possible, SELCOPY/i 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.

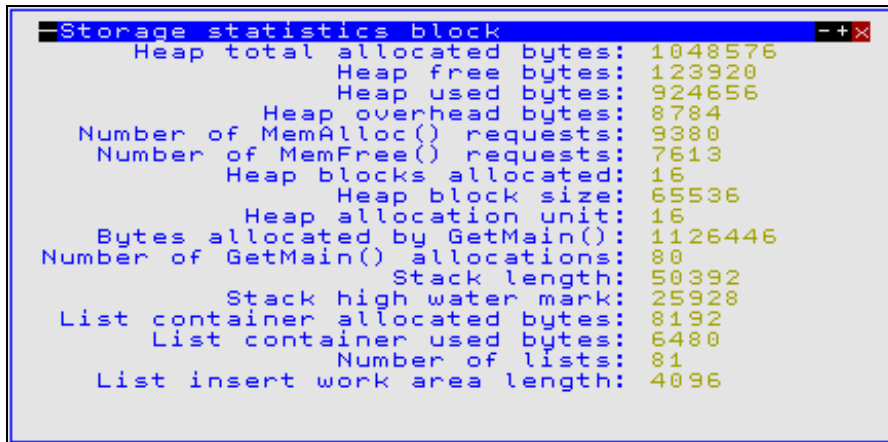
A function acquires dynamic storage for local variables from the stack when it is called and frees this storage when it returns control to its caller.

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 SELCOPY/i are in use.

The **Stack high water mark** represents the maximum amount of stack storage that has been in use in the current SELCOPY/i session.

Lists

The List insert work area and List container are areas of storage associated with SELCOPY/i listing facilities.



```

Storage statistics block
Heap total allocated bytes: 1048576
Heap free bytes: 123920
Heap used bytes: 924656
Heap overhead bytes: 8784
Number of MemAlloc() requests: 9380
Number of MemFree() requests: 7613
Heap blocks allocated: 16
Heap block size: 65536
Heap allocation unit: 16
Bytes allocated by GetMain(): 1126446
Number of GetMain() allocations: 80
Stack length: 50392
Stack high water mark: 25928
List container allocated bytes: 8192
List container used bytes: 6480
Number of lists: 81
List insert work area length: 4096
  
```

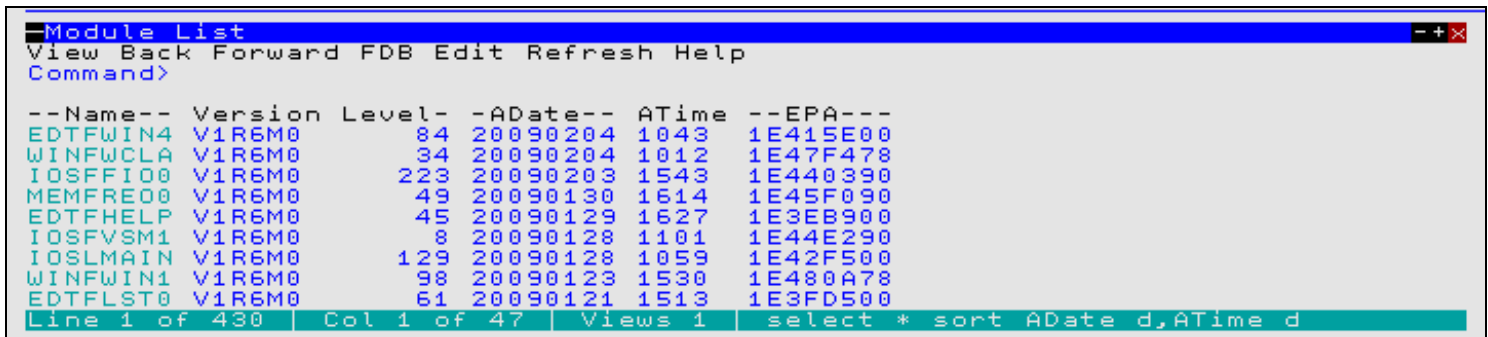
Figure 142. Storage Statistics window.

SELCOPY/i Module List Window

The SELCOPY/i Module List may be opened to display information on all modules that comprise SELCOPY/i, via the following:

- Select 'SELCOPY/i module list' from the Utilities/System menu in the **CBL** 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**.



```

Module List
View Back Forward FDB Edit Refresh Help
Command>

--Name--  Version  Level-  -ADate--  ATime  --EPA--
EDTFWIN4  V1R6M0    84  20090204  1043  1E415E00
WINFWCLA  V1R6M0    34  20090204  1012  1E47F478
IOSFFI00  V1R6M0   223  20090203  1543  1E440390
MEMFRE00  V1R6M0    49  20090130  1614  1E45F090
EDTFHELP  V1R6M0    45  20090129  1627  1E3EB900
IOSFVSM1  V1R6M0     8  20090128  1101  1E44E290
IOSLMAIN  V1R6M0   129  20090128  1059  1E42F500
WINFWIN1  V1R6M0    98  20090123  1530  1E480A78
EDTFLST0  V1R6M0    61  20090121  1513  1E3FD500
Line 1 of 430 | Col 1 of 47 | Views 1 | select * sort ADate d,ATime d
  
```

Figure 143. Module List window.

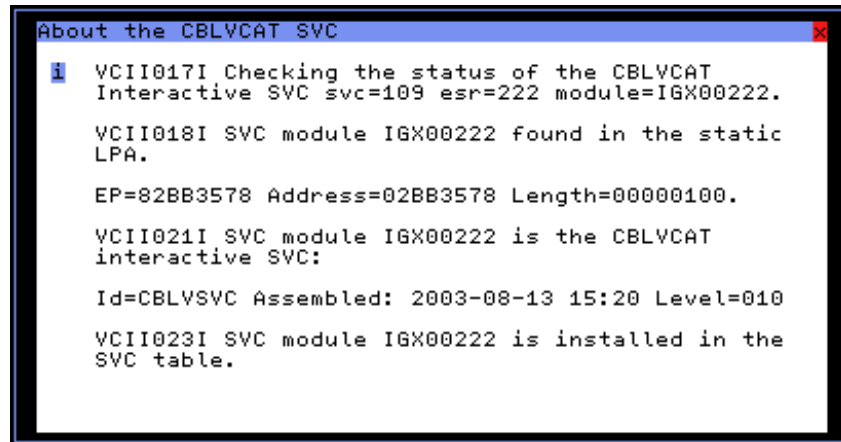
CBLVCAT SVC window (=8.8.7)

The CBLVCAT SVC window may be opened via the following:

- Select option 7. 'SVC Status' from the **System Information Menu**.
- Select 'SELCOPY/i SVC' from the Utilities/System menu in the **CBLLe 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.



```

About the CBLVCAT SVC
i VCII017I Checking the status of the CBLVCAT
Interactive SVC svc=109 esr=222 module=IGX00222.

VCII018I SVC module IGX00222 found in the static
LPA.

EP=82BB3578 Address=02BB3578 Length=00000100.

VCII021I SVC module IGX00222 is the CBLVCAT
interactive SVC:

Id=CBLVSVC Assembled: 2003-08-13 15:20 Level=010

VCII023I SVC module IGX00222 is installed in the
SVC table.
  
```

Figure 144. CBLVCAT SVC window.

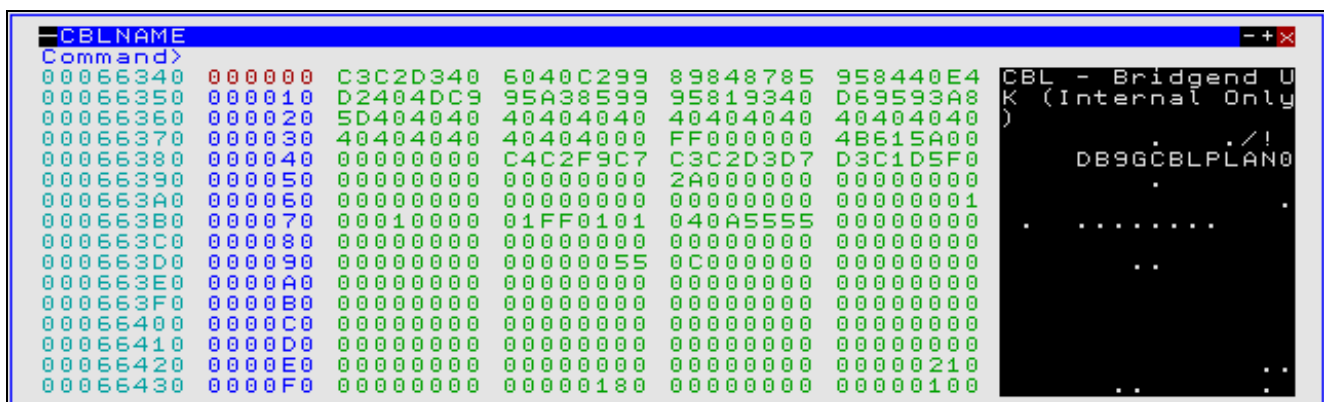
CBLNAME Window

The CBLVCAT SVC window may be opened via the following:

- Select 'CBLNAME' from the Utilities/System menu in the **CBLLe 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 SELCOPY/i.

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.



```

CBLNAME
Command>
00066340 000000 C3C2D340 6040C299 89848785 958440E4
00066350 000010 D2404DC9 95A38599 95819340 D69593A8
00066360 000020 5D404040 40404040 40404040 40404040
00066370 000030 40404040 40404000 FF000000 4B615A00
00066380 000040 00000000 C4C2F9C7 C3C2D3D7 D3C1D5F0
00066390 000050 00000000 00000000 2A000000 00000000
000663A0 000060 00000000 00000000 00000000 00000001
000663B0 000070 00010000 01FF0101 040A5555 00000000
000663C0 000080 00000000 00000000 00000000 00000000
000663D0 000090 00000000 00000055 0C000000 00000000
000663E0 0000A0 00000000 00000000 00000000 00000000
000663F0 0000B0 00000000 00000000 00000000 00000000
00066400 0000C0 00000000 00000000 00000000 00000000
00066410 0000D0 00000000 00000000 00000000 00000000
00066420 0000E0 00000000 00000000 00000000 00000210
00066430 0000F0 00000000 00000180 00000000 00000100
  
```

Figure 145. CBLNAME 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 **CBLLe 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.

```

SELFCOPY/i - File Search: CBL.JCL(S*) 2011/12/09 1
View Refresh Back Forward FDB Text Help wS wR
Command>
Dataset> CBL.JCL(S*)
Search string> PGM
-----Record-----
-Member- RecNo HitNo -----
SDATAXX      8      1 //STEP01 EXEC PGM=SELFCOPY,REGION=4096K
SDEBATCH     8      1 //STEP01 EXEC PGM=SDEAMAIN,REGION=0M
SDEBAT02     8      1 //STEP01 EXEC PGM=SDEAMAIN,REGION=0M
SDEBAT03     9      1 //STEP01 EXEC PGM=SDEAMAIN,REGION=0M
SDEBAT04     9      1 //STEP01 EXEC PGM=SDEAMAIN,REGION=0M
SDEBAT05     9      1 //STEP01 EXEC PGM=SDEAMAIN,REGION=0M
SDEBAT06     9      1 //STEP01 EXEC PGM=SDEAMAIN,REGION=0M
SDEBSQL      9      1 //STEP01 EXEC PGM=SDEAMAIN,REGION=0M
SDEBU        12     1 //SDEBU EXEC PGM=ADDRSSU,REGION=4096K
SDECOMPF     9      1 //*BXAVTAM EXEC PGM=DBXAVTAM,PARM='DBXAVTAM'
SDECOMPF    10     2 //STEP01 EXEC PGM=SDEAMAIN,REGION=0M
SDEFCOPY     9      1 //STEP01 EXEC PGM=SDEAMAIN,REGION=0M
SDELD        7      1 //STEP01 EXEC PGM=SDEAMAIN,REGION=0M
SDESTRUC     7      1 //STEP01 EXEC PGM=SDEAMAIN,REGION=0M
SEARCH       5      1 //SEARCH EXEC PGM=ISRSUPC,
SELCA MSD    13     1 //STEP0001 EXEC PGM=IDCAM,REGION=4M
SELCA MSD    49     2 //STEP0001 EXEC PGM=SELFCOPY
SELCBMP      29     1 //G EXEC PGM=DFSRR00,REGION=&RGN,
SELCCOMP     44     2 // DD DSN=IMS&REL..PGMLIB,DISP=SHR
SELCCOMP     6      1 //SELCCOMP EXEC PGM=SELFCOPY
SELCDM1      28     1 //SELC EXEC PGM=SELFCOPY
SELCEXP      1      1 // EXEC PGM=SELFCOPY
SELCF COP    6      1 //STEP0001 EXEC PGM=SDEAMAIN,REGION=0M
SELCDG       8      1 //SELC EXEC PGM=SELFCOPY
SELCHFS      6      1 //SELC EXEC PGM=SELFCOPY
SELCHFSW     7      1 //SELC EXEC PGM=SELFCOPY
SELCHNG4     8      1 //FILEMGR EXEC PGM=SELFCOPY
SELCIMS      40     1 //G EXEC PGM=DFSRR00,REGION=&RGN,
SELCIMS      51     2 // DD DSN=IMS&REL..PGMLIB,DISP=SHR
SELCLCTL     3      1 //STEP01 EXEC PGM=SELFCOPY
SELCLIVE     7      1 //SELCLNK EXEC PGM=IEWL,REGION=2M,
SELCLIVE    23     2 //SELCLNKQ EXEC PGM=IEWL,REGION=2M,
SELCLIVE    38     3 //SELCLNKL EXEC PGM=IEWL,REGION=2M,
SELCLIVE    52     4 //SELCLC EXEC PGM=IEWL,REGION=2M,
SELCLIVE    69     5 //SELCLVP EXEC PGM=SELFCOPY
SELCLKED    392    1 .TXT .↑ . .POS,KEYLOC FOR UNB ISAM.CANCELLED
Line 1 of 241 | Col 1 of 110 | Views 1 | select * sort Member,RecNo

```

Figure 146. 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 characters.

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 CBLLe 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 CBLLe 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 CBLLe CLI SUBMIT command using the entry name as input. (A CBLLe frame window must be active for this operation to succeed.)
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 CBLLe 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 PF2 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 CBLLe 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 CBLLe CLI SUBMIT command using the entry name as input. (A CBLLe frame window must be active for this operation to succeed.)
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 CBLLe 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 PF2 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 existence 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 SELCOPY/i (CBL) 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 provides 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 provides 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.


```

Search for Library Members
File Run Command Help
Command>
                                                                    Scroll> Csr
                                                                    Lines 1-24 of 24

Member(s) to locate:
Pattern 1> SELCOPY                                     (Single Character Wildcard = %)
Pattern 2> CBLV                                       (Multiple Character Wildcard = *)
Pattern 3> CBLI
Pattern 4>

Libraries to search: (Dataset or DD names with optional wildcards)
Pattern 1> CBL.LINK*
Pattern 2> NBJ.**.LOAD
Pattern 3>
Pattern 4>

Special concatenations to search: (optional)
Z APF Authorised load-libraries
Z Link Listed load-libraries
- SELCOPY/i edit macros

Filter: (optional)
Where> ALIAS='N'
(e.g. LastMod > '2010/10/05' & User = 'IBMUSER' )

Options:
Z Quiet (Suppress prompts to continue when very many, or zero hits)

```

Figure 147. 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 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>
Pattern 2>
Pattern 3>
Pattern 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>
Pattern 2>
Pattern 3>
Pattern 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 identify 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 SELCOPY/i 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](#).)

SELCOPY/i edit macros

This option field (MACCHK) indicates that all libraries in the user's current CBL text editor macro path are to be searched.

This list of libraries may be displayed using the CBL 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 existence 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 SELCOPY/i (CBL) 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.

```

- NBJ2.LLX.D2012006.T112436.TXT      255 V SEQ      Size=22      Alt=0,0;0      -+x
Command> | .....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7..
000001  ** NBJ2.LLX.D2012006.T112436.TXT *** L=001 --- 2012/01/06 11:24:36 (NBJ2
000002  <only 'member(\s)'; hide
000003  <only 'member(\s)'; hide; x all '0 member(\s)' | Show summary lines only
000004
000005  ** 'NBJ.JCL(SQ*)' ***
000006  <e 'NBJ.JCL(SQ11749)' | has 1 member(s)
000007  | 2008/03/07 15:17 8 NBJ
000008  ** 'NBJ.CTL(SQ*)' ***
000009  <e 'NBJ.CTL(SQ11884)' | has 1 member(s)
000010
000011  ** 'NBJ.SELCOPY.DEMO.CTL(SQ*)' ***
000012  <e 'NBJ.SELCOPY.DEMO.CTL(SQ11480)' | has 3 member(s)
000013  <e 'NBJ.SELCOPY.DEMO.CTL(SQ11480B)' | 2005/02/03 11:10 50 NBJ
000014  <e 'NBJ.SELCOPY.DEMO.CTL(SQ11480C)' | 2005/02/03 12:27 78 NBJ
000015  | 2005/02/11 13:16 46 NBJ
000016  ** 'NBJ.SSC.CTL(SQ*)' ***
000017  has 0 member(s)
000018  ** 'NBJ.SSC.CTL.F80(SQ*)' ***
000019  <e 'NBJ.SSC.CTL.F80(SQ11756A)' | has 2 member(s)
000020  <e 'NBJ.SSC.CTL.F80(SQ11756B)' | 2008/03/19 11:15 33 NBJ
000021  | 2008/03/19 11:30 124 NBJ
000022  <L1X NBJ.JCL(SQ*) NBJ.**.CTL(SQ*) SUBSET \WHERE LASTMOD < '2010' \
000023  * * * End of File * * *
Output from: L1X NBJ.JCL(SQ*) NBJ.**.CTL(SQ*) SUBSET \WHERE LASTMOD < '2010' \

```

Figure 148. 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 least 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 **CBL**e 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.

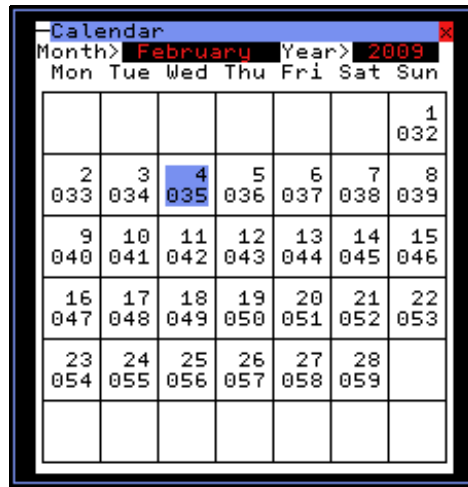


Figure 149. Calendar window.

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 **CBL 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.

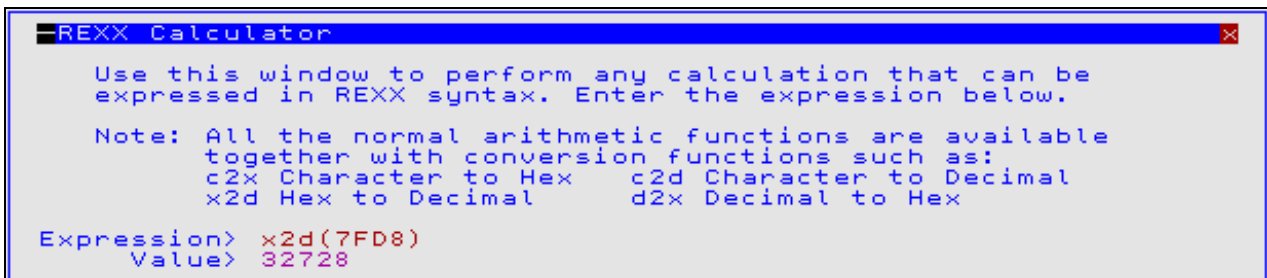


Figure 150. REXX Calculator window.

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 Define 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 **CBL main window menu** bar.
 - By any SELCOPY/i utility that requires allocation of an output data set.
 - Enter the CBL command **ALLOCATE** with no parameters on the command line of any window.
 - Perform CBL or SDE Edit using a new data set name, add some data and SAVE.
 - Perform CBL 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.

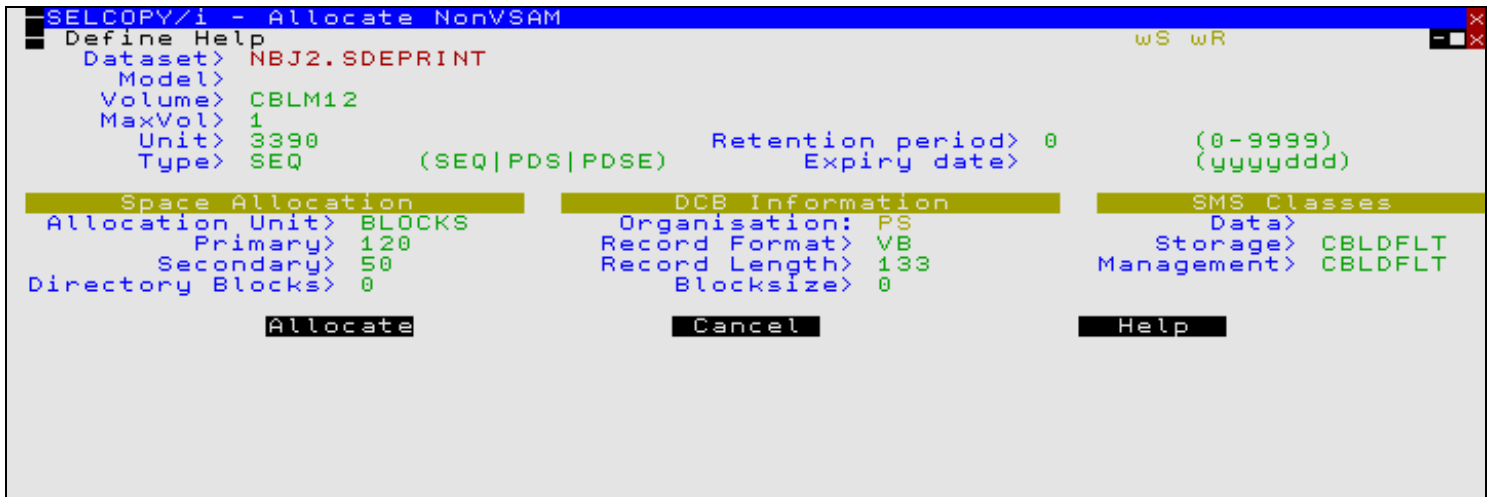


Figure 151. Allocate nonVSAM window.

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 CBL 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 CBL 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 **CBL 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 CBL or SDE Edit using a new data set name, execute SET DSORG KSDS/ESDS/RRDS/LDS, as required, add some data and SAVE.
- Perform CBL 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.

```

Define VSAM KSDS
Define Job AMS Help
>
Cluster Name> CBL.SELCOPY.DEMO1.KSDS          Type> KSDS
Model>                                           Reuse> Y
Catalog>                                         Span> N
                                                Speed> Y
                                                Writecheck> Y
                                                Erase> Y
Avg rec len> 70                                Key length> 6
Max rec len> 100                              Key offset> 0
Buffer size>                                  Free space/CI> 0
Expiry>                                        Free space/CA> 0
                                                SMS Data Class>
Region share option> 1                        SMS Stor Class> CBLDFLT
System share option> 3                       SMS Mgmt Class> CBLDFLT

Data name> CBL.SELCOPY.DEMO1.KSDS.DATA
Volumes> CBLM04                                Ordered> N
Space type> TRACKS                             Primary> 1      Secondary> 1
CI Size> 4096

Index name> CBL.SELCOPY.DEMO1.KSDS.INDEX
Volumes> CBLM04                                Ordered> N
Space type> TRACKS                             Primary> 1      Secondary> 1
CI Size> 512                                  Imbed> N      Replicate> N
  
```

Figure 152. 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 CBL command **SUBMIT**.)

AMS

Opens a CBL 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 **CBL** main window menu bar.
- Enter line command **AMSDIALOG** with option GDG on the command line of any window. Alternatively, use the synonym **AMSG**.

```

SELCPY/i - Define GDG Base
File Help JCL Command
Command>
ZZSGAMSG
wS wR  Scroll> Csr
Lines 1-20 of 21

To select an existing GDG on which to model the new GDG, enter a GDG name mask
which includes wild card characters "*", "**" and/or "%".

GDG Name> _____ (required)
Model> _____

Owner> _____ GDG owner id. Default is current TSO user id.
Limit> 10 (1-255) Max number of associated GDS entries. (required)
Catalog> _____ Catalog or Alias.

Action on exceeding limit:
/ Delete (roll off) only the oldest GDS entry belonging to the GDG.
- Delete all GDS entries belonging to the GDG. (EMPTY)

Action on delete (roll off) of GDS entries:
/ Uncatalog the GDS but do not scratch it from volumes it occupies.
- Uncatalog the GDS and scratch it from volumes it occupies. (SCRATCH)

```

Figure 153. 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>

Identify 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.

Data Set Information

The Data Set Information panels (DSINFONV) are **interactive panel windows**.

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

A file'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. 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.

Dataset Information - Non-VSAM

```

SELCOPY/i - Dataset Information - Non-VSAM (Page 1 of 3)
File Report
Command>
DSINFONV
Dataset: NBJ.INST.CBL13295.SZZSHELP.HTML
Catalog: USERCAT.CBLCAT

General Data:
Volume: CBLM12
Device Type: 3390
Organisation: PDS
Record Format: VB
Record Length: 256
Block Size: 27998
First Extent Tracks: 419
Secondary Tracks: 168

Current Allocation:
Number of Tracks: 419
Number of Extents: 1
Number of Volumes: 1
Max Dir Blocks: 700

Current Utilisation:
Number of Members: 1857
Used Dir Blocks: 310
Used Tracks: 320
Used Extents: 1

Dataset Date Information:
Created: 2013/10/22
Expires:
Last Referenced: 2013/11/06

SMS Classes:
Storage: CBLDFLT
Data:
Management: CBLDFLT

```

Figure 154. 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 displayed 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 foreground 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.
- 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.
- Catalog:**
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 system 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.

Stripe 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.

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.

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 system 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.

```

SELCOPY/i - Dataset Information - VSAM (Page 1 of 3)
File Report IDCAMS RecordCount          wS wR
Command>                                Scroll> Cs
DSINFOV                                  Lines 1-20 of 21
Entry:  CBL.BBDEMO02.KSDS
Data:   CBL.BBDEMO02.KSDS.DATA           Data Volume: CBLM03
Index:  CBL.BBDEMO02.KSDS.INDEX         Index Volume:  CBLM03
Catalog: USERCAT.CBLCAT

Data Component Information:              DEFINE Options:
Device Type: 3390                        Load Option:  SPEED
Organisation: KSDS                      Share Options: 1 - 3
KSDS Key Length: 6                      Buffer Space: 40960
KSDS Key Offset: 0                      Write Check: NO
Average Record Length: 64               Erase on Delete: NO
Maximum Record Length: 100              Imbedded Index: NO
                                           Replicated Index: NO
Allocated Space: Unit Primary Secondary Reuse Option: NO
Data: Tracts 1 1                       Spanned Records: NO
Index: Tracts 1 1                      Key Ranges Present: NO
Dataset Date Information:               AIX-Unique keys:
Created: 2008/06/11                    AIX-Upgrade:
Expires:
Last Modified: 2008/12/04 15:45:44 GMT

```

Figure 155. 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 displayed 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 foreground 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.

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.

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 dataset was defined.

Expires:

Date on which the dataset expires.

Last Modified:

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 to 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 DFSMSStvs processing. A VSAM RLS or DFSMSStvs 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 DFSMSStvs 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 DFSMSStvs 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 DFSMSStvs 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 DFSMSStvs processing. If the data set is opened for input in non-RLS mode, a VSAM RLS or DFSMSStvs 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 DFSMSStvs processing. If the data set is opened for input in non-RLS mode, a VSAM RLS or DFSMSStvs 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.

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.

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.

Page 4

This display lists the dataset's related entries i.e. each path and/or alternate index

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.

Create Structure (SDO) Menu (=9)

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.

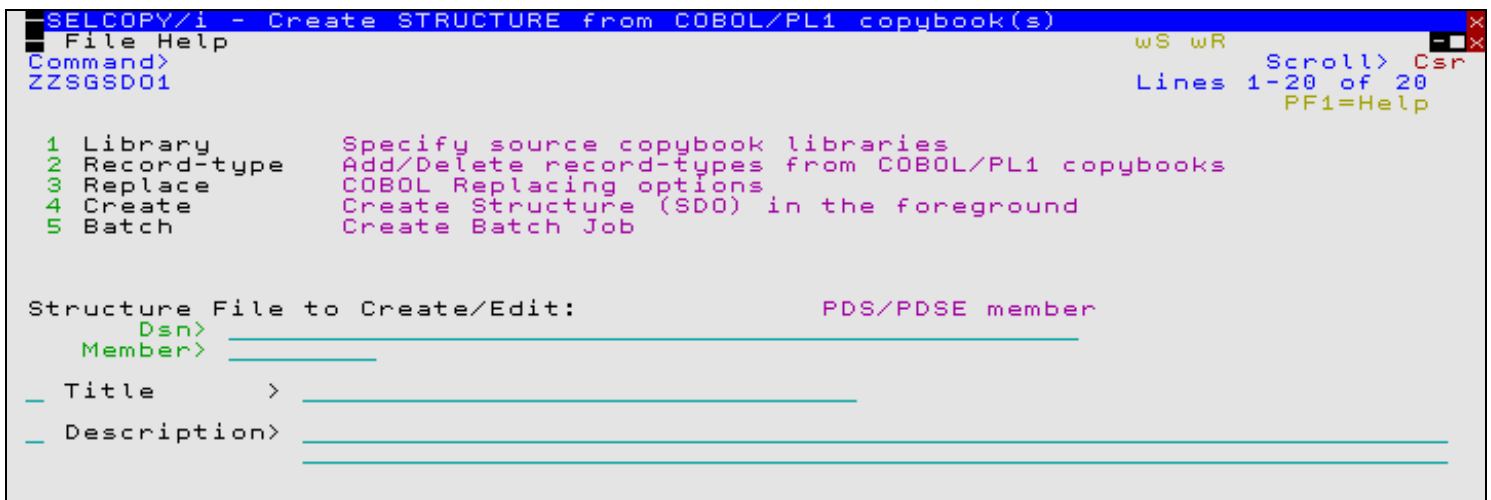
Options

- | | |
|------------|---|
| 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. |

Create Structure from Copybook(s)

The Create Structure from COBOL or PL1 copybooks panels assist with generation of a SELCOPY/i structure definition file using one or more 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)



```
SELCOPY/i - Create STRUCTURE from COBOL/PL1 copybook(s)
File Help                               wS wR
Command>                                Scroll> Csr
ZZSGSD01                                Lines 1-20 of 20
                                           PF1=Help

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

Structure File to Create/Edit:           PDS/PDSE member
  Dsn> _____
  Member> _____

_ Title      > _____
_ Description> _____
```

Figure 156. SELCOPY/i - 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**
2. **Record-type**
3. **Replace**
4. **Create**
5. **Batch**

Create

Create the SDO structure file in the SELCOPY/i 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 SELCOPY/i 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

```

SELCOPY/i - Create STRUCTURE from COBOL/PL1 copybook(s)
File Help
Command>
ZZSGSDOL
Create Structure - Copybook Library List.
Copybook Library Dataset name
<...+...>
000001 CBL.CBLI.COPYBOOK.COBOL
000002 NBJ.COBOB.CPYBK
000003 *** End of Data ***
PF1=Help
2 Rows

```

Figure 157. SELCOPY/i - 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 SELCOPY/i **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

```

SELCOPY/i - Create STRUCTURE from COBOL/PL1 copybook(s)
File Help
Command>
ZZSGSDOR
Add a table row then press PF2 to specify its record identification-criteria
Create Structure - Define Record-Types.
Copybook Type Record-Type Name (01-Lev)
Library Record
Member Offset
<...+...> <.> <...+...> <...+...> <...+...>
000001 K61XBAS PRI K-MST-AMIN 0
000002 KX01SEG SEC KX-AUTH 0
000003 KX02SEG SEC KX-LOCATION 0
000004 K60XBAS DEF K-MST-USER 0
000005 KNA1SEG SEC KN-MLT-ENG 0
000006 KNA2SEG SEC KN-INTER 0
000007 *** End of Data ***
Record Offset

```

Figure 158. SELCOPY/i - 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 SELCOPY/i **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 a 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 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**).

```

SELCOPY/i - Create STRUCTURE: Define record-type
File Help                                     wS wR
Command>                                     Scroll> Csr
ZZSGSDOR                                     Lines 1-16 of 16

Member  >  K61XBAS          Copybook Member Name
Name    >  K-MST-AMIN       + Record-Type Name
                                     Normally defined by 01-Level Name

Type    >  PRI              Default, Primary or Secondary
Language>  COBOL            Compiler Language
Offset  >  0                Offset within record at which to start mapping
/ Id    >  CTYPE >> C'AM1'  + Use PF2 to expand
                                     Record identification criteria

Press PF3 to return to the record-types list table.

```

Figure 159. SELCOPY/i - Create Structure - Define Record-Types Zoomed View.

Panel Input Fields

Member>

Name of a 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 a 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 member as a **COBOL** copybook, **PL1** include file or **ADATA** (COBOL or PL1) output file.

Note that it is possible to combine COBOL, PL1 and ADATA source members in the same SDO generation.

This field corresponds to CREATE STRUCTURE parameter SOURCE 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

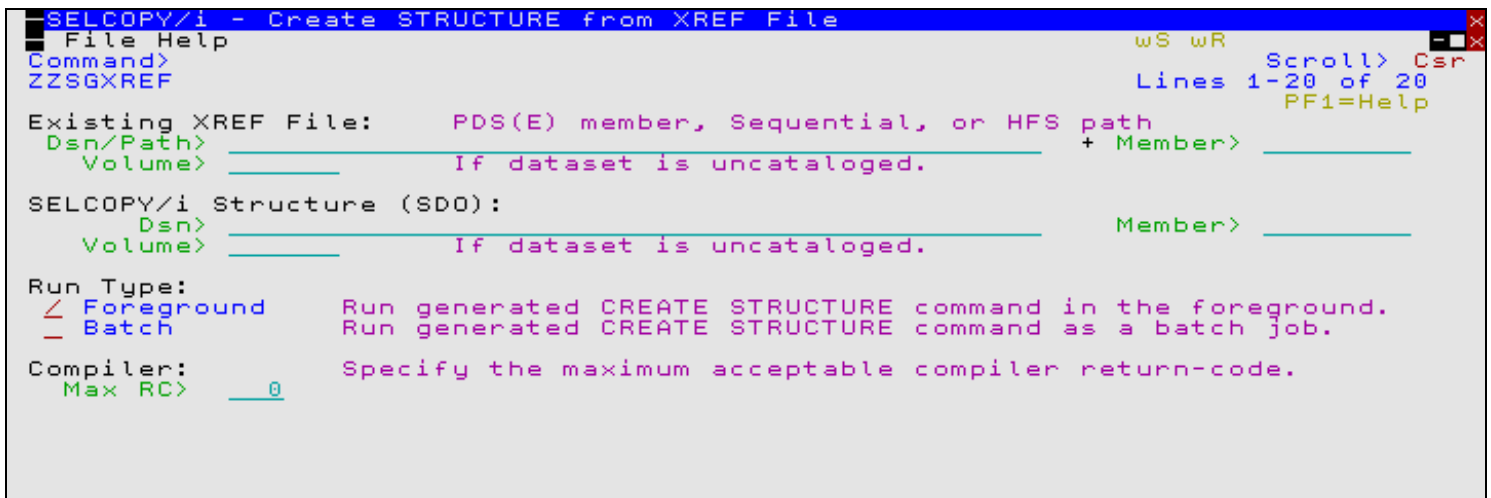


Figure 160. SELCOPY/i - Create Structure from XREF File.

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 SELCOPY/i 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.)

SELCOPY/i Structure (SDO):

Fields that identify the SELCOPY/i 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 SELCOPY/i **Foreground** or will generate a JCL **Batch** job as a temporary file displayed in a CBL text edit view.

The batch job runs SDEAMAIN to execute CREATE STRUCTURE syntax and may be submitted to batch using the SUBMIT command.

Compiler:

SELCOPY/i 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 SELCOPY/i 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.

```

SELCOPY/i - Display record layouts defined by an SDO/copybook
File Help
Command>
ZZSGLAYO
SELCOPY/i Structure (SDO) or Copybook dataset:
Dsn> CBL.CBLI.SDO
Volume> _____ If dataset is uncataloged.
Type: / SDO _ AData _ Cobol _ PL1
Options:
Number Width> _5
_ Expand Array Fields
Issue "LAYOUT" from any SDE browse/edit view to list the current structure.
Optionally add the "EXP" parameter to expand array fields.
WS WR
Scroll> Csr
Lines 1-20 of 20
PF1=Help

```

Figure 161. SELCOPY/i - Display Record Layout Panel.

Panel Input Fields

SELCOPY/i 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.


```

SELCOPY/i - Layout from CBL.CBLI.SDO(ASMADATA)
View Refresh Back Forward FDB Text Help
Command>
-----Name-----Picture-----RefNo Start -End- Leng
1 Source Group 1 1 232 2
2 ADataHdr Group 2 1 12
3 Language INTEGER(1) 3 1 1
3 Group 4 2 3
4 RecTypeX HEXADECIMAL(2) 5 2 3
4 RecTypeN INTEGER(2) 6 2 3
3 ALevel INTEGER(1) 7 4 4
3 ADAFlag HEXADECIMAL(1) 8 5 5
3 FLevel HEXADECIMAL(1) 9 6 6
3 HEXADECIMAL(4) 10 7 10
3 DataLen INTEGER(2) 11 11 12
2 SRCESD INTEGER(4) 12 13 16
2 SRCStmt INTEGER(4) 13 17 20
2 SRCInRec INTEGER(4) 14 21 24
2 SRCParRec INTEGER(4) 15 25 28
2 SRCInNum INTEGER(4) 16 29 32
2 SRCParNum INTEGER(4) 17 33 36
2 SRCLoctr HEXADECIMAL(4) 18 37 40
Line 1 of 722 | Col 1 of 81 | Views 1 | select *
    
```

Figure 162. SELCOPY/i - 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

Create File Filter (=10)

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 SELCOPY/i 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 **FILTER** with no parameters from the command line of any window.

```
SELCOPY/i - Create File Filter
File Help
Command>
ZZSGFLT0
ws wr
Scroll> Csr
Lines 1-20 of 21
PF1=Help

Filter File: PDS(E) member, Sequential, or HFS path + Member>
Dsn/Path>
Volume>
If dataset is uncataloged.

Filter Limit: Specify the maximum number of records to be selected.
Stopaft> 0 (zero indicates no limit)

Selection Criteria: Specify 'I' to set INCLUDE selection criteria.
Type> I Specify 'X' to set EXCLUDE selection criteria.

Structure File: Required for option 2.
Dsn>
Volume> Member>
Type: / SDO - AData - Cobol - PL1

Action>
1. Text-Edit existing filter file. (PF4)
2. Specify Unformatted Selection Criteria from scratch. (PF5)
3. Specify Formatted Selection Criteria from scratch. (PF6)
4. Create FILTER object. (PF13)
```

Figure 163. SELCOPY/i - 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 CBL 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 CBL e 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 expressions 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 CBL e 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

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.

```

SELCOPY/i - FILTER (unformatted) - Selection Criteria
File Help
Command>
ZZSGFLTR
FILTER (unformatted) - Selection Criteria.
AND/OR Position Length ROp Value
<.> <...+> <...+> <.> <...+...1...+...2...+...3...+...4...+>
000001 AND 23 3 = c' CBL'
000002 AND 35 2 = x' FFFF'
000003 OR 23 3 = c' NBJ'
000004 *** End of Data ***
  
```

Figure 164. SELCOPY/i - Unformatted Selection Criteria.

Standard SELCOPY/i **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 entries 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.

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*.

```

SELCOPY/i - Filter Definition: Field Selection Criteria
File Help
Command>
ZZSGFLTR
Logical Operator > AND Logically AND/OR with previous.
Will be ignored if first condition.
Field Details
Position > 35
Length > 2
Relational Operator > = Enter blank for a list of operators/meanings.
Value > x' FFFF'
  
```

Figure 165. SELCOPY/i - 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 (ZZSGFLT1) 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.

```

SELCOPY/i - Create Filter: Generate INCLUDE Subclause(s)
File Help                                     wS wR
Command>                                     Scroll> Csr
ZZSGFLT1                                     PF1=Help   PF5=Show Selected   PF6=Show All

Structure: NBj.CBLI.SDO(COBSALES)
Type:      / SDO      AData      Cobol      PL1
FILTER (formatted) - INCLUDE record-types (with optional criteria).      5 Rows
Sel Record Type Identification Criteria
+
000001 S REC-CUST
000002 = REC-CARD
000003 S REC-ORDER
000004 = REC-PAYMENT
000005 = REC-NOTE
000006 *** End of Data ***

```

Figure 166. SELCOPY/i - Formatted - INCLUDE/EXCLUDE record-types.

Standard SELCOPY/i **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.

```

SELCOPY/i - FILTER (formatted) - Selection Criteria
File Help                                     wS wR
Command>                                     Scroll> Csr
ZZSGFLTW                                     PF5=Show Selected   PF6=Show All
FILTER (formatted) - Selection Criteria.      13 Rows
AND Lev Name      Fmt Pic      ROp Value
/OR
<.> <.> <...+...> <> <...+...> <.> <...+...1...+...2...+...3...>
000001 ----- 2 CUST-ID      FB 9(5)      <= 05000
000002 ----- 3 row(s) excluded -----
000005 OR       2 COUNTRY      AN X(2)      = 'UK'
000006 ----- 8 row(s) excluded -----
000014 *** End of Data ***

```

Figure 167. SELCOPY/i - Formatted Selection Criteria.

Standard SELCOPY/i **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), relational operator (ROp) and the value against which the field will be tested. Null or invalid entries 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 single record view of the row using the ZOOM command (assigned to <PF2> by default.) As well 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 corresponds to the filter WHERE *expression*.

```

SELCOPY/i - Filter Definition: Field Selection Criteria
File Help                                     wS wR
Command>                                     Scroll> Csr
ZZSGFLTW                                     Lines 1-13 of 13

Logical Operator   >  AND           Logically AND/OR with previous.
                                   Will be Ignored if first condition.

Field Details
Name              :  CUST-ID
Level             :      2
Data Type         :  FB
Picture           :  9(5)

Relational Operator >  <= <u>____</u>  Enter blank for a list of operators/meanings.
Value             >  05000
  
```

Figure 168. SELCOPY/i - 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>

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:

Informational output fields that describe the formatted field.

Name:

Field name.

Level:

Level of nesting below the first level (GROUP or STRUCTURE) field.

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.

Print Data File (=11)

Print Data File - Input

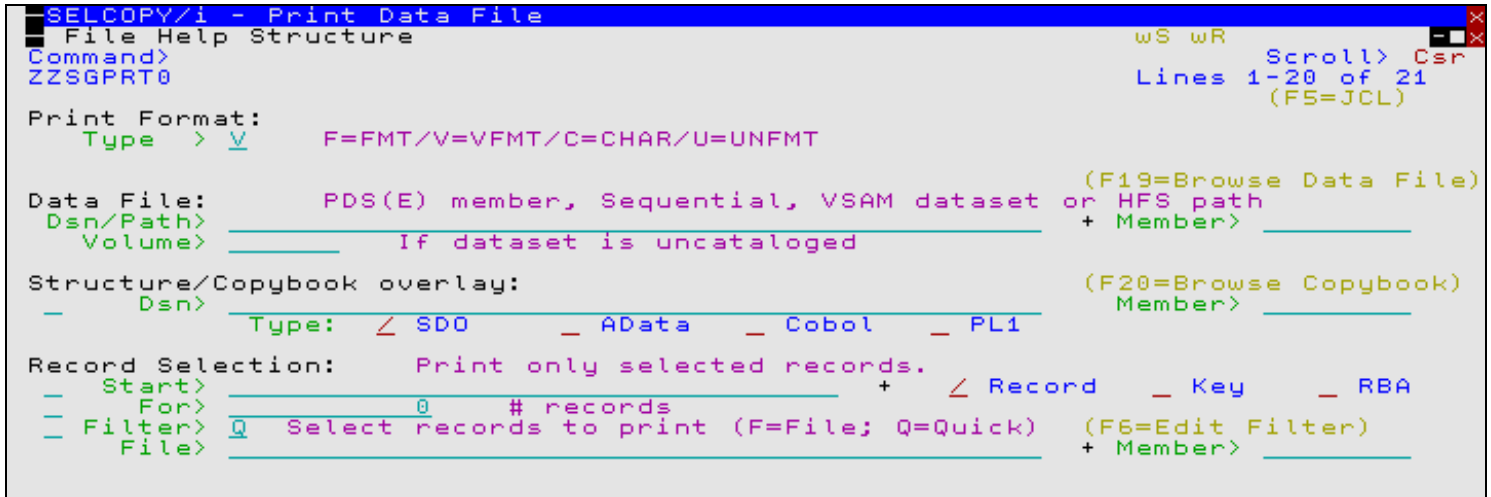


Figure 169. SELCOPY/i - Print Data File.

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.

The Print Data File panel view (ZZSGPRT0) is an **interactive panel window**, opened in selection of option 11. "Print" in the SELCOPY/i Primary option menu. Its purpose is to generate a job which may be submitted to print the data file in batch.

Pressing the <Enter> key or, if configured, **double-clicking the left mouse button** will proceed to the next Print Data File panel view based on the selected print format.

PF18 may be pressed in any of the Print Data File panels to generate the PRINT File batch job based on the existing values set in the supporting panel views.

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 wildcard 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.

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.

Print Data File - VFMT Options

```

SELCOPY/i - Print Data File - VFMT Options
File Help
Command>
ZZSGPRT0
wS wR
Scroll> Csr
Lines 1-20 of 21
(F5=JCL)

VFMT Print - Vertical Formatted (Table-Mode) Options:
Order > A A=AcrossThenDown/D=DownThenAcross/T=Trunc
Offset> P P=POS/X=HEX/R=REL Show> I F=FORMAT/O=OFFSET/P=PICTURE/T=TYPE

Show Hex Data > N Y/N Show TTR/RBA > N Y/N
Show FILLER fields> N Y/N Show Scale Line > Y Y/N
Show Shadow Lines > N Y/N Show Record Length> N Y/N
Translate ASCII > N Y/N Show Record Number> Y Y/N

-----
Choose the action required by entering a number in the input field then
pressing ENTER, or alternatively point-and-shoot at the option description.
-----

Option> 3
1. Select Record-types to Print
2. Select Column Names to Print
3. Continue
  
```

Figure 170. SELCOPY/i - 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.

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 accommodate 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 accommodate 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 continuation 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**.

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> F=FORMAT/O=OFFSET/P=PICTURE/T=TYPE

Determines the contents of the field description header line.

F	Displays length/format where <i>length</i> identifies the field's length within the unformatted record, and <i>format</i> 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 <i>position</i> and <i>length</i> 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.

This option corresponds to the SDE primary command **SHOW**.

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 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**.

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 to Print

By default records of any record-type are included in the printed output. Use this option 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 included in the printed output. (If **DOWNTHEACROSS** has been selected, only one record-type can be selected.)

2. Select Column Names to Print

By default all fields (from all selected record-types) are included in the print output Use this option 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.

3. Continue

Continue to the **Print Data File - Destination** view.

Print Data File - FMT Options

```

SELCPY/i - Print Data File - FMT Options
File Help
Command>
ZZSGPRT0
ws wR
Scroll> Csr
Lines 1-20 of 21
(F5=JCL)

FMT/MAP Print - Formatted (Single-record Mode) Options:
Show > F F=FORMAT/O=OFFSET/P=PICTURE/T=TYPE Show> N L=LEVEL/N=NUMBER
Offset> P P=POS/X=HEX/R=REL

Show Hex Data > N Y/N
Show FILLER fields> N Y/N
Show Group Items > Y Y/N
Translate ASCII > N Y/N

-----
Choose the action required by entering a number in the input field then
pressing ENTER, or alternatively point-and-shoot at the option description.
-----

Option> 3

1. Select Record-types to Print
2. Select Column Names to Print
3. Continue

```

Figure 171. SELCOPY/i - 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.

Show> F=FORMAT/O=OFFSET/P=PICTURE/T=TYPE

Determines the contents of the field description display.

F	Displays length/format where <i>length</i> identifies the field's length within the unformatted record, and <i>format</i> 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 <i>position</i> and <i>length</i> 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.

This option corresponds to the SDE primary command **SHOW FORMAT/OFFSET/PICTURE/TYPE**.

Show> L=LEVEL/N=NUMBER

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)

This option corresponds to the SDE primary command **SHOW LEVEL/NUMBER**.

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**.

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 to Print

By default records of any record-type are included in the printed output. Use this option 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 included in the printed output.

2. Select Column Names to Print

By default all fields (from all selected record-types) are included in the print output Use this option 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.

3. Continue

Continue to the **Print Data File - Destination** view.

Print Data File - Select Record-Types

```

SELCOPY/i - SELCOPY/i Print File: Choose record-types to print
File Help
Command>
ZZSGPRTV
Select record-types to print from Structure/Copybook: PF3=Continue PF1=Help
Name> CBL.CBLI.SDO(MBRLISTR) +
SDE PrintFile - VIEW clause. 18 Rows
Record Type Use When
+
000000 *** Top of Data ***
000001 Source RecTypeX=X'0030'
000002 JobId13 JIDVer='1.3.0'
000003 JobId RecTypeX=X'0000'
000004 ADataId RecTypeX=X'0001'
000005 CompUnit RecTypeX=X'0002'
000006 Output RecTypeX=X'000A'
000007 OPT RecTypeX=X'0010'
000008 ESD RecTypeX=X'0020'
000009 DCDS RecTypeX=X'0034'
000010 Instruction RecTypeX=X'0036'
000011 RLD RecTypeX=X'0040'
000012 Symbol RecTypeX=X'0042'
000013 XRef RecTypeX=X'0044'
000014 RegXRef RecTypeX=X'0045'

```

Figure 172. SELCOPY/i - 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.

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

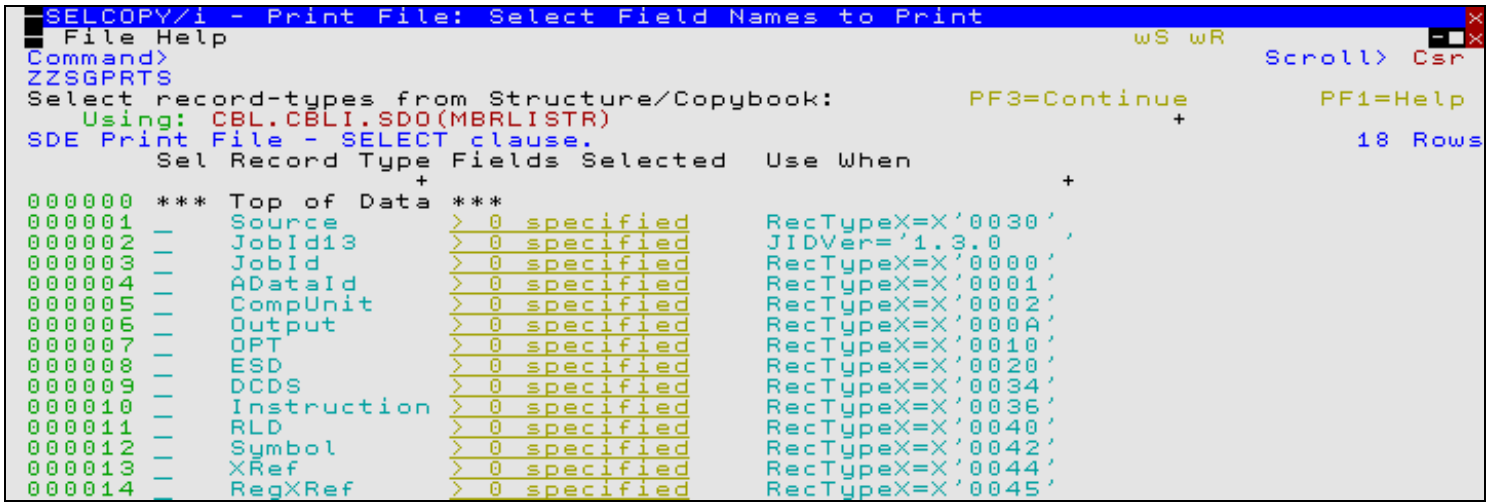


Figure 173. SELCOPY/i - Print Data File - Select Field Names.

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.

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 displayed. 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

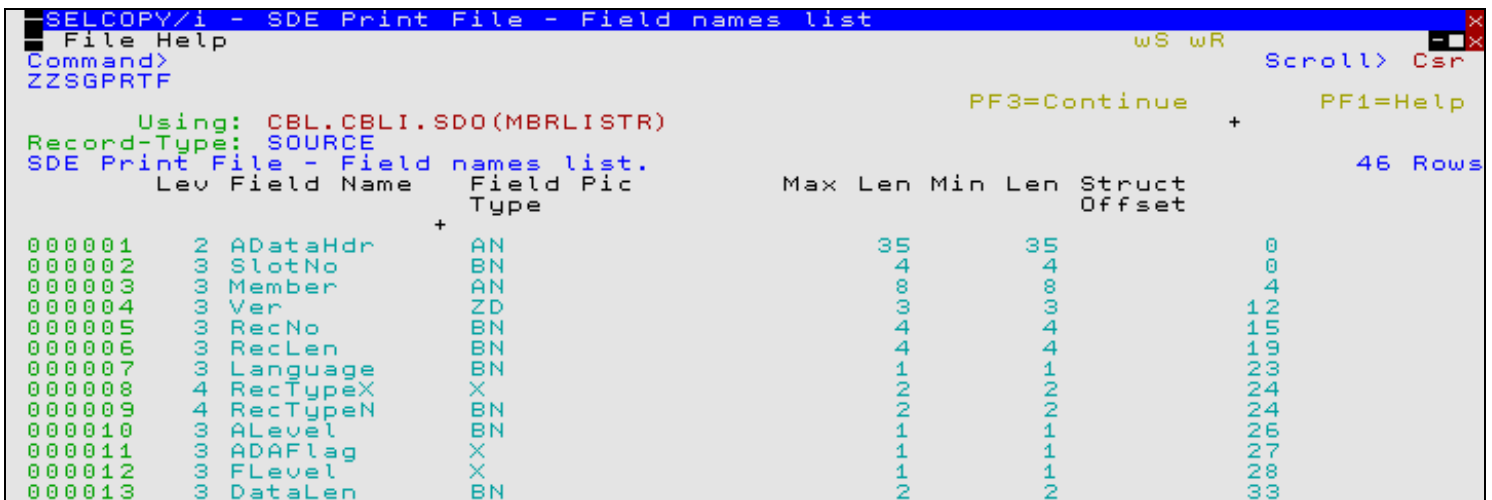


Figure 174. SELCOPY/i - Print Data File - Field Names List.

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.

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 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

```
SELCOPY/i - Print Data File - CHAR Options
File Help
Command>
ZZSGPRT0
Lines 1-20 of 21
(F5=JCL)

CHAR Print - Unformatted (Table-Mode) Options:
Order > A   A=AcrossThenDown/T=Trunc
Offset > P   P=POS/X=HEX/R=REL

Show Hex Data      > N   Y/N
Translate ASCII    > N   Y/N

Show TTR/RBA      > N   Y/N
Show Scale Line   > Y   Y/N
Show Record Length > Y   Y/N
Show Record Number > Y   Y/N
```

Figure 175. SELCOPY/i - 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.

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 accommodate 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 accommodate 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

```

SELCOPY/i - Print Data File - UNFMT Options
File Help
Command>
ZZSGPRT0
UNFMT Print - Unformatted (Single-record Mode) Options:
Offset> P P=POS/X=HEX/R=REL
Show Hex Data > N Y/N
Translate ASCII > N Y/N
wS wR
Scroll> Csr
Lines 1-20 of 21
(F5=JCL)

```

Figure 176. SELCOPY/i - 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.

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

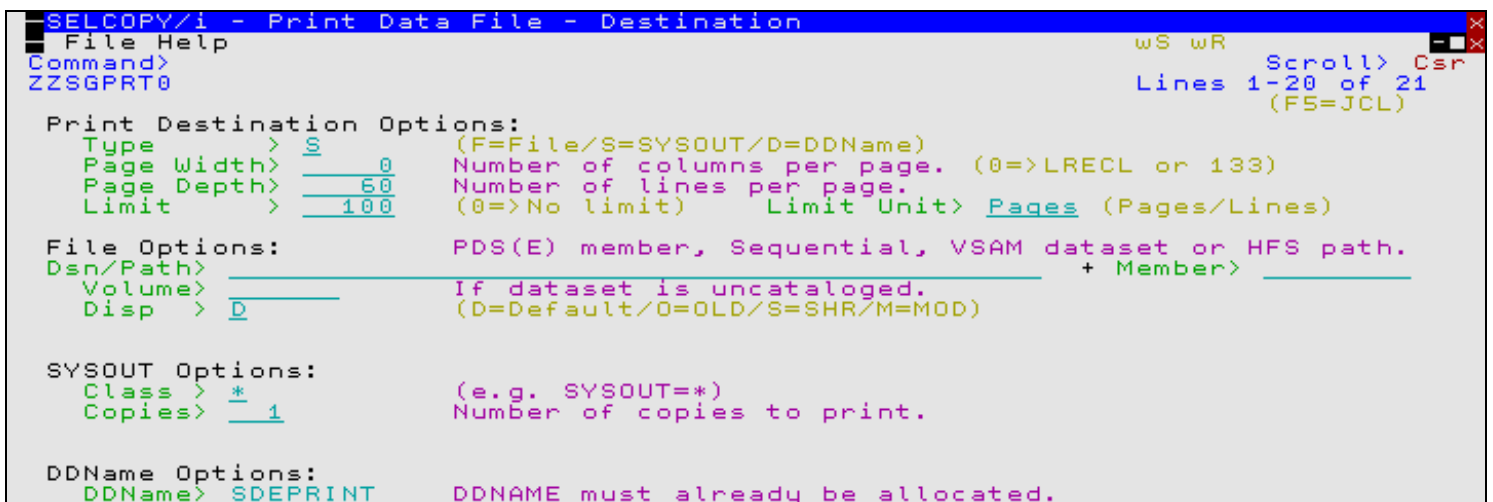


Figure 177. SELCOPY/i - Print Data File Destination.

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.

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) indicates 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.

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 SELCOPY1.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 subsystem 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 SELCOPY/i DB2 panels, encompassing the DB2 functionality available in SELCOPY/i.

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.

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 thereafter 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.

```

DB2(CBLA): Primary Option Menu
File Help
Command>
ZZS2PRIM
                                     Lines      Scroll> Csr
                                     1-13 of 13

 1. DB2           Execute DB2 Commands.      User: NBJ2
 2. SQL           Execute SQL Commands.      Version: 3.1B
 3. Edit          Edit Tables and Views.     Date: 2011/08/31
 4. Browse        Browse Tables and Views.   Time: 16:45:16
 5. Create        Create DB2 Objects.       OpSys: z/OS 1.11.0
 6. Drop          Drop DB2 Objects.
 7. List          List DB2 Objects.
 8. Audit         Audit Trail functions.
 9. Utilities     DB2 Utilities.

DB2 SubSystem> CBLA
Current SQLID> NBJ2
DB2 Version: 9.1.0

Create Audit File> /

```

Figure 178. DB2 Primary Option Menu Panel.

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 DB2 Primary Option menu panel.

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 |

Panel Input/Output Fields

User: An output field displaying the user's userid.

Version: An output field displaying the version of SELCOPY/i.

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. SELCOPY/i 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 SELCOPY/i 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 SELCOPY/i DB2.

Create Audit File>

An option check box that indicates that SELCOPY/i 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 SELCOPY/i 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.

```

DB2(DB9G): Execute DB2 Commands
View Refresh Back Forward FDB Text Help
Command>                               Scroll> Csr
ZZS2XDB2
DB2 command:  Use PF2 with the cursor in the command input field to expand the
               input area for long commands.

Command>  -display bufferpool(bp1)
Byte Limit> 4096      Limit the size of the response. 0 means no limit.

-----Output-----
DB2 Command:
-display bufferpool(bp1)

Command return code (decimal) 0 reason code x'00000000'
          IFI return code (decimal) 0 reason code x'00000000'
Bytes returned 706      Bytes not returned 0

DB2 Command Response:
DSNB401I  -DB9G BUFFERPOOL NAME BP1, BUFFERPOOL ID 1, USE COUNT 0
DSNB402I  -DB9G BUFFER POOL SIZE = 0 BUFFERS  AUTOSIZE = NO
          ALLOCATED = 0      TO BE DELETED = 0
          IN-USE/UPDATED = 0      BUFFERS ACTIVE = 0
DSNB406I  -DB9G PGFIX ATTRIBUTE -
          CURRENT = NO
          PENDING = NO
          PAGE STEALING METHOD = LRU
DSNB404I  -DB9G THRESHOLDS -
          VP SEQUENTIAL = 80
          DEFERRED WRITE = 30      VERTICAL DEFERRED WRT = 5, 0
          PARALLEL SEQUENTIAL =50      ASSISTING PARALLEL SEQT= 0
DSNB9022I -DB9G DSNB1CMD '-DISPLAY BUFFERPOOL' NORMAL COMPLETION

Line 1 of 21 | Col 1 of 69 | Views 1 | select *

```

Figure 179. 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 panel (ZZS2XSQL) is an [interactive panel window](#), opened on selection of option 2. in the DB2 Primary 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 Execute SQL Statements 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 control 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. 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.

```

DB2(CBLA): Execute SQL statements
View Refresh Back Forward FDB Text Help
ZZS2XSQL
Exec SQL:      Use PF2 with the cursor in the statement input field to expand
               the input area for long SQL statements.

Statement> SELECT * FROM ZZS.ZZSSYSMOD
Row Limit> 20      Limit the number of SELECTed rows. 0 means no limit.

SYSMOD--  -FMID--  -ZAP--  -----TITLE-----
QS00001  AZZS300  -    1 SELCOPY/i: FSU error when attempting multiple changes.
QS00002  AZZS300  -    2 SELCOPY/i: SQLCODE=-126 on SELECT ORDER BY using DB2 op
QS00003  AZZS300  -    3 SELCOPY/i: 0C4 on DB2 command with long field string.
QS00004  AZZS300  -    4 SELCOPY/i: 0C4 on DB2 Edit/Browse tables of same name.
QS00005  AZZS300  -    5 SELCOPY/i: Command DB2 SSN=xxxx fails to open new DB2 m
QS00006  AZZS300  -    6 SELCOPY/i: ALLOC dialog TYPE=SEQ, DIR Blocks=0 => DSORG
QS00007  AZZS300  -    7 SELCOPY/i: FCOPIY ALLOC dialog for LOADLIB model ignored
QS00008  AZZS300  -    8 SELCOPY/i: DB2 EDIT VARCHAR field update error.
QS00009  AZZS300  -    9 SELCOPY/i: Only 1st VARCHAR char displayed on DB2 BROWS
QS00010  AZZS300  -   10 SELCOPY/i: FSU/FCOPY unresponsive following restart of
QS00011  AZZS300  -   11 SELCOPY/i: DSNT408I SQLCODE=-104 on DB2 EDIT SELECT(col
QI00001  AZZI300  -    1 SELCOPY: DB2 Input error on null VARCHAR value in last
QI00002  AZZI300  -    2 SELCOPY: Loop on DB2 UPD when table has been closed.
QS00012  AZZS300  -   12 SELCOPY/i: Message ZZSZ008W deemed excessive and so wit
QS00013  AZZS300  -   13 SELCOPY/i: DB2 LIST SQLCODE=-519 following failed DB2 L
QS00014  AZZS300  -   14 SELCOPY/i: IEBCOPY Dialog returns "IEBCOPY Control Card
QS00015  AZZS300  -   15 SELCOPY/i: User not prompted for LRECL of inserted reco
QS00016  AZZS300  -   16 SELCOPY/i: ZZSD015E after SD edit UPD of a non-existent
QS00017  AZZS300  -    0 SELCOPY/i: SDE Panel Bug fixes.
QS00018  AZZS300  -    0 SELCOPY/i: Compare File/Library trailing blank diffs.

Command>
Line 1 of 20 | Col 1 of 716 | Views 1 | select *

```

Figure 180. Execute SQL Statements Panel - SQL Query.

```

DB2(CBLA): Execute SQL statements
View Refresh Back Forward FDB Text Help
Command>
ZZSX005E  DSNT408I SQLCODE = -204, ERROR: CBL.ZZSQURY IS AN UNDEFINED NAME
Exec SQL:      Use PF2 with the cursor in the statement input field to expand
               the input area for long SQL statements.

Statement> CREATE TABLE CBL.ZZSIQ LIKE CBL.ZZSQURY
Row Limit> 100      Limit the number of SELECTed rows. 0 means no limit.

-----Result-----
2011/09/05 16:54:10 EXECUTE IMMEDIATE
CREATE TABLE CBL.ZZSIQ LIKE CBL.ZZSQURY

DSNT408I  SQLCODE = -204, ERROR: CBL.ZZSQURY IS AN UNDEFINED NAME
DSNT418I  SQLSTATE = 42704 SQLSTATE RETURN CODE
DSNT415I  SQLERRP = DSNX0FE SQL PROCEDURE DETECTING ERROR
DSNT416I  SQLERRD = -200 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I  SQLERRD = X'FFFFFFFF38', X'00000000', X'00000000',
X'FFFFFFFF', X'00000000', X'00000000', SQL DIAGNOSTIC
INFORMATION

2011/09/05 16:54:10 COMMIT issued SQLCODE=0

Line 1 of 13 | Col 1 of 80 | Views 1 | select *

```

Figure 181. Execute SQL Statements 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 is **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**.

Edit Tables and Views

The Edit Object panel (ZZS2EDIT) is an **interactive panel window**, opened on selection of option 3. in the DB2 Primary options menu or on issuing prefix command "E" in a DB2 table/view/alias/synonym list.

This panel 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 SDE edit view. General features of the SDE editor and features specific to SDE DB2 table edit views are documented in detail in the **SELCOPY/i Structured Data Editor (SDE)** manual.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the table edit. Prior to editing the table data, SELCOPY/i will open a new connection to the DB2 subsystem specified by the DB2 Primary option menu panel and, if selected, will allocate a new audit log data set. This isolates actions specific to the DB2 table edit from all other DB2 actions and table editing performed on the subsystem.

```

SELCOPY/i - DB2(CBLA): Edit Object
File Run Command Help
Command>
ZZS2EDIT
ws wR
Scroll> Csr
Lines 1-39 of 39

DB2 Object:

Owner> ZZS
Name> ZZSSYSMOD + (optional) +

Row/Column Selection Options:
From> 1 Start row within the result set.
For> 100 Number of Rows to be edited.

Select> SYSMOD, LUPDATE, LUPTIME, DESCR +
Where> SYSMOD LIKE '%RS%' +
Order By> LUPDATE DESC, LUPTIME DESC +

COMMIT Options:
 / Commit on SAVE with no errors
 - Commit on SAVE
 - Commit on exit from edit session

Load Options:
 / Skip locked rows
 / Execute Commit following load

Concurrency (Isolation) Options:
 / Use DB2 Default Isolation level
 - Uncommitted Read (least restrictive)
 - Cursor Stability
 - Read Stability
 - Repeatable Read (most restrictive)

Initial Display Format:
 / Table - Single

Profile Macro:
 / Use Default.
 - Use Specified Macro.
 - Do not use a Profile Macro.

Explicit Table Lock:
 / None
 - Share mode
 - Exclusive mode

Use/Keep Locks:
 / None
 - Share
 - Update
 - Exclusive

Miscellaneous Options:
 / Create Audit File
 - Do not protect Prime Key

Macro Name> SDEPROF

```

Figure 182. DB2 Edit Object Panel.

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** Executes the table edit for the specified input field parameters. Hitting <Enter> will perform the same action.
- Command** Opens a text edit view for a temporary data set containing the SDE **EDIT command** syntax generated for the selected input field values. 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.)
- Help** Open the general help for the DB2 Edit Object panel.

Field Entries

DB2 Object:
Fields that together identify the DB2 table or view in the current location which is to be edited.

Owner>
The owner of the required table or view.

Name>
The name of the required table or view.

Row/Column Selection Options:
Fields that together identify the DB2 results table rows and columns to be edited and the order in which they appear.

From>
Specifies the row number of the first row to be displayed for edit in the DB2 results table returned by the SQL query. This row becomes row 1 within the SDE window edit view. Rows that occur before this row number are not included within the edit session.
Default is row 1.

For>
Specifies the maximum number of rows to be displayed from the DB2 results table returned by the SQL query. Rows that fall outside the range of rows identified by the From and For fields are not included within the edit session.
Default is to display all selected rows.

Select>
Specifies a DB2 SQL SELECT clause to be included in the prepared SQL select statement from which the DB2 results table is derived. See "*DB2 SQL Reference*" for syntax of the SQL select clause.

Where>
Specifies a DB2 SQL WHERE clause to be included in the prepared SQL select statement from which the DB2 results table is derived. See "*DB2 SQL Reference*" for syntax of the SQL where clause.

Order By>
Specifies a DB2 SQL ORDER BY clause to be included in the prepared SQL select statement from which the DB2 results table is derived. See "*DB2 SQL Reference*" for syntax of the SQL order-by clause.

COMMIT Options:
Identifies when a DB2 COMMIT should be performed for changed table data. Mutually exclusive options are as follow:

- ◇ **Commit on SAVE with no errors**
COMMIT when SAVE is executed without errors. This option is default.
- ◇ **Commit on SAVE**
COMMIT when SAVE is executed regardless of any save errors.

◇ **Commit on exit from edit session**

COMMIT only when closing the last edit view of the table data.

Load Options:

Options relating to actions performed at load of results table rows.

Skip locked rows

This option field is ignored unless an **isolation level** of **Cursor Stability (CS)** or **Read Stability (RS)** is in effect. Skip locked rows indicates that any selected rows that are already locked by another process should be skipped and are not 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

Indicates that a COMMIT is to be performed following the initial load of the results table rows selected for edit, thus releasing all DB2 table locks performed during load of the data. This includes any **explicit table locks** that may have been applied.

Default is **not** to perform a COMMIT following load of the table rows.

Explicit Table Lock:

Applicable to edit of DB2 base tables only, indicates whether or not a DB2 SQL LOCK TABLE command should be executed before loading rows from the table and, if so, the mode of lock to be performed.

Note that **caution** should be taken when selecting a table locking option as other users and applications may be prevented from accessing the table. If the table is locked then the lock is held until the next COMMIT is actioned. See "*DB2 SQL Reference*" for details on the effects of LOCK TABLE options.

Mutually exclusive options are as follow:

◇ **None**

No explicit table locking prior to load. This option is default.

◇ **Share mode**

Share mode table locking prevents anything other than read-only operations from being performed on the table whilst the lock is in effect.

◇ **Exclusive mode**

Unless a process is running with an isolation level of Uncommitted Read (UR) in which case read-only (dirty read) operations may be performed, exclusive mode table locking prevents another process from performing any operation on the table whilst the lock is in effect.

Concurrency (Isolation) Options:

Specifies a DB2 SQL isolation-clause to be included in the prepared SQL select statement. See "*DB2 SQL Reference*" for details on the SQL Isolation clause and "*DB2 Performance Monitoring and Tuning Guide*" for details on the effects of isolation level on concurrency and protection of DB2 table data.

Mutually exclusive options are as follow:

◇ **Use DB2 Default Isolation level**

The default isolation level as set by the BIND of the SELCOPY/i DB2 package and plan. (See sample job ZZSDB2B which sets default isolation of Cursor Stability.)

◇ **Uncommitted Read**

Uncommitted Read (UR) acquires few locks but may result in edit of uncommitted data.

◇ **Cursor Stability**

Cursor Stability (CS) allows maximum concurrency with data integrity.

This isolation method will release a lock for a row after it has been read. If an attempt is made to update or delete a row (on save), then the row is locked, the predicate re-evaluated to ensure that it still qualifies for the results table and only then, if values match, is the row updated or deleted.

◇ **Read Stability**

Read Stability (RS) locks only those rows that satisfy the results table predicate, releasing the locks only when a COMMIT is performed. Rows that do not satisfy the predicate are eligible for update or delete by other applications. New rows may also be inserted by other applications.

The mode of lock performed on the rows is indicated by the **Use/Keep Locks** options.

◇ **Repeatable Read**

Repeatable Read (RR) is most restrictive since it prevents other applications from performing update, insert or delete even on rows that do not satisfy the results table predicate. Locks on all accessed rows are held until a COMMIT is performed.

The mode of lock performed on the rows is indicated by the **Use/Keep Locks** options.

Use/Keep Locks:

Applicable only if isolation level Read Stability or Repeatable Read is selected, this option identifies the mode of lock to be used and kept on locked pages and rows.

Mutually exclusive options are as follow:

- ◇ **None**
Use and keep a row using the default lock mode.
- ◇ **Share**
Concurrent processes can read but not change the locked row and may acquire SHARE or UPDATE mode locks on the row.
- ◇ **Update**
Concurrent processes can acquire a SHARE mode lock on the row and may only read the data without acquiring a page or row lock.
- ◇ **Exclusive**
Concurrent processes cannot acquire any mode of lock on the row.

Initial Display Format:

Identifies the initial display format of the DB2 table rows in the **SDE window view**.

Mutually exclusive options are as follow:

- ◇ **Table**
Display the DB2 table rows in tabular **multi record view**.
- ◇ **Single**
Display the DB2 table rows one at a time in **single record view**.

Miscellaneous Options:

Miscellaneous edit options.

Create Audit File

Open a new SELCOPY/i DB2 audit data set to record changes to the edited table made during this edit session. See DB2 **Audit Trail Functions** for details.

Do not protect Prime Key

If selected, specifies that data occupying columns that comprise the table's primary key is eligible for edit and update.

Profile Macro:

Indicates whether or not an SDE data edit profile macro is to be executed when the SDE window view is opened and, if so, the macro to be executed.

Mutually exclusive options are as follow:

- ◇ **Use Default.**
Use the default SDE edit profile macro (SDEPROF).
- ◇ **Use Specified Macro.**
Use an SDE edit profile macro with macro name specified by the **Macro Name>** field.
- ◇ **Do not use a Profile Macro.**
No SDE 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 SDE edit profile macro to be executed when the table is edited.

Browse Tables and Views

The Browse Object panel (ZZS2BROW) is an **interactive panel window**, opened on selection of option 4. in the DB2 Primary options menu or on issuing prefix command "B" in a DB2 table/view/alias/synonym list.

This panel allows the user to configure DB2 table browse options prior to loading rows from the DB2 results table and presenting the column data in an SDE window view. General features of the SDE editor and features specific to SDE DB2 table browse views are documented in detail in the **SELCOPY/i Structured Data Editor (SDE)** manual.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the table browse.

```

DB2(DB9G): Browse Object
File Run Command Help
Command>
ZZS2BROW
Lines 1-21 of 21
DB2 Object:
  Owner> CBL
  Name> APIFUNC
Row Selection Options:
  From>
  For>
  Select> FUNCMOD, FUNCNAME, FUNCTITLE
  Where> FUNCMOD#REF=0
  Order By> FUNCMOD
Initial Display Format:
  / Table
Profile Macro:
  / Use Default.
  - Use Specified Macro.
  - Do not use a Profile Macro.
Macro Name> SDEPROF
  
```

Figure 183. DB2 Browse Object Panel.

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** Executes the table browse for the specified input field parameters. Hitting <Enter> will perform the same action.
- Command** Opens a text edit view of a temporary data set containing the SDE **BROWSE command** syntax generated for the selected input field values. 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.)
- Help** Open the general help for the DB2 Browse Object panel.

Field Entries

- DB2 Object:** Fields that together identify the DB2 table or view in the current location which is to be browsed.
- Owner>** The owner of the required table or view.
- Name>** The name of the required table or view.
- Row Selection Options:** Fields that together identify the DB2 results table rows and columns to be browsed and the order in which they appear.
- From>** Specifies the row number of the first row to be displayed for browse in the DB2 results table returned by the SQL query. This row becomes row 1 within the SDE window view. Rows that occur before this row number are not included within the browse session. Default is row 1.
- For>**

Specifies the maximum number of rows to be displayed from the DB2 results table returned by the SQL query. Rows that fall outside the range of rows identified by the From and For fields are not included within the browse session.
Default is to display all selected rows.

Select>

Specifies a DB2 SQL SELECT clause to be included in the prepared SQL select statement from which the DB2 results table is derived. See "*DB2 SQL Reference*" for syntax of the SQL select clause.

Where>

Specifies a DB2 SQL WHERE clause to be included in the prepared SQL select statement from which the DB2 results table is derived. See "*DB2 SQL Reference*" for syntax of the SQL where clause.

Order By>

Specifies a DB2 SQL ORDER BY clause to be included in the prepared SQL select statement from which the DB2 results table is derived. See "*DB2 SQL Reference*" for syntax of the SQL order-by clause.

Initial Display Format:

Identifies the initial display format of the DB2 table rows in the **SDE window view**.

Mutually exclusive options are as follow:

◊ **Table**

Display the DB2 table rows in tabular **multi record view**.

◊ **Single**

Display the DB2 table rows one at a time in **single record view**.

Profile Macro:

Indicates whether or not an SDE data edit profile macro is to be executed when the SDE window view is opened and, if so, the macro to be executed.

Mutually exclusive options are as follow:

◊ **Use Default.**

Use the default SDE edit profile macro (SDEPROF).

◊ **Use Specified Macro.**

Use an SDE edit profile macro with macro name specified by the **Macro Name>** field.

◊ **Do not use a Profile Macro.**

No SDE 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 SDE edit profile macro to be executed when the table is edited.

Create DB2 Objects

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 | 8. View |
| 2. Work File Database | 9. Alias |
| 3. User Database | 10. Synonym |
| 4. Tablespace | 11. Type |
| 5. Work File Tablespace | 12. Trigger |
| 6. Base Table | 13. Sequence |
| 7. Materialized Query Table | 14. Role |

Create Storage Group

The Create Storage Group panel (ZZS2CSG0) is an **interactive panel window**, opened on selection of option 1. in the Create DB2 Object options menu. and may be used to create a new storage group in the current server.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE STOGROUP statement in the SELCOPY/i foreground.

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

Executes the SQL CREATE STOGROUP generated by the specified input field parameters. Hitting <Enter> will perform the same action.

Command

Opens a text edit view for a temporary data set containing **SQL command** CREATE STOGROUP syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL CREATE STOGROUP statement syntax generated for the selected input field values.

Help

Open the general help for the Create Storage Group panel.

Field Entries**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.

Volume Serial Numbers>

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 or 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.

Create Work File Database

The Create Work File Database panel (ZZS2CDBW) is an **interactive panel window**, opened on selection of option 2. in the Create DB2 Object options menu. Applicable only in a data sharing environment, this panel may be used to create the one and only work file database in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE DATABASE AS WORKFILE statement in the SELCOPY/i foreground.

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** Executes the SQL CREATE DATABASE generated by the specified input field parameters. Hitting <Enter> will perform the same action.
- Command** Opens a text edit view for a temporary data set containing **SQL command** CREATE DATABASE syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL CREATE DATABASE statement syntax generated for the selected input field values.
- Help** Open the general help for the Create Work File Database panel.

Field Entries

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).

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 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.

Create for 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.

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.

Create User Database

The Create User Database panel (ZZS2CDBU) is an **interactive panel window**, opened on selection of option 3. in the Create DB2 Object options menu. This panel may be used to create a user database in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE DATABASE statement in the SELCOPY/i foreground.

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

Executes the SQL CREATE DATABASE generated by the specified input field parameters. Hitting <Enter> will perform the same action.

Command

Opens a text edit view for a temporary data set containing **SQL command** CREATE DATABASE syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL CREATE DATABASE statement syntax generated for the selected input field values.

Help

Open the general help for the Create User Database panel.

Field Entries

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 DSNDDB.

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 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.

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 proceeding 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 wildcard 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 object panels.

```

SELCPY/i - DB2(CBLA): Create Work File Tablespace
File Help                                     wS wR
Command>                                     Scroll> Csr
ZZS2CTSW                                     Lines 1-20 of 21
View: Work File Table Space Values
Use wildcard "%" to select a database and/or model table space.

Table Space Name> TSWORK01                   In Work File Database> DSNDB07
Model Table Space> DSN32K00

Table Space data sets:
└ DB2 managed
  Storage Group> SYSDEFLT                    Enter pattern with wildcard "%" to select.
  Primary Alloc> -1                          Minimum pri alloc in KB. (-1 => DB2 default)
  Secondary Alloc> -1                        Minimum sec alloc in KB. (-1 => DB2 default)
  Action on DROP > - Erase                    / Do not erase
- User managed
  Catalog Alias> CBLDB2A                    Enter pattern with wildcard "%" to select.

Page and Segment Sizes:
Buffer Pool> BP32K                          Select a 4K or 32K buffer pool. (Page size)
Segment Size> 16                             #pages per segment. (Must be a multiple of 4, 4-64)

1. Help (PF1)          2. Next (ENTER)      3. Back (PF3)        4. Exit (PF15)

```

Figure 184. 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 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 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 name may be selected from the **Select Catalog Alias** panel.

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.

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 proceeding 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.

```

SELCPY/i - DB2(CBLA): Create Tablespace
File Help
Command>
ZZS2CTS0
View: Table Space Name & Type
Enter a new table space name, select its type and optionally model values
for the table space from an existing table space in the current DB2 server.

Table Space Name> USERTS01      Model Table Space> _____
In Database> DEMODB             Model Database > _____
Model TSType:

Table Space Type:      Enter "/" to select the required table space type.
- Segmented            Single, segmented dataset - multiple tables.
                       (Each segment has data from 1 table.)
- Partitioned          Multiple, non-segmented datasets - 1 table.
                       (Rows distributed by column key ranges.)
- Universal Partition-by-Growth Multiple, segmented datasets - 1 table.
                       (Datasets added as table grows.)
/ Universal Partition-by-Range Multiple, segmented datasets - 1 table.
                       (Rows distributed by column key ranges.)
- Large Object          Multiple datasets of large object data.

1. Help (PF1)          2. Next (ENTER)        3. Back (PF3)         4. Exit (PF15)

```

Figure 185. 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 (partitioned 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**

```

SELCOPY/i - DB2(CBLA): Create Tablespace
File Help                                     wS wR
Command>                                     Scroll> Csr
ZZS2CTS0                                     Lines 1-20 of 21
View: Universal Part-by-Range                Tablespace: USERTS01 In Database: DEMODB

Table Space data sets:
Data set Size > 4G                          DSSIZE>4G => SMS extended addressability.
└ DB2 managed
  Storage Group > DEMOSG                     Enter pattern with wildcard "%" to select.
  Primary Alloc > -1                         Minimum pri alloc in KB. (-1 => DB2 default)
  Secondary Alloc > -1                       Minimum sec alloc in KB. (-1 => DB2 default)
  Action on DROP > Erase                     └ Do not erase
  Define on CREATE > Yes                     └ No
- User managed
  Catalog Alias >                           Enter pattern with wildcard "%" to select.

Partitions, Page and Segment Sizes:
Buffer Pool > BP32K                          Default, 4K, 8K, 16K or 32K page size buffer pool.
Segment Size > 4                             #pages per segment. (Must be a multiple of 4, 4-64)
#Partitions > 5                               Number of table space partitions (data sets.)
- Use individual partition data set attributes. PFS>Show Partitions

1. Help (PF1)          2. Next (ENTER)      3. Back (PF3)        4. Exit (PF15)

```

Figure 186. 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 name may be selected from the **Select Catalog Alias** panel.

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.

```

SELCOPY/i - DB2(CBLA): Tablespace Partition Attributes
File Help
Command>
ZZS2CTSP
Table Space Partition Attributes: Tablespace: USERTS01 In Database: DEMODB
Each entry in the table below represents a table partition with attributes
inherited from the Create Table Space panel. Select and update partition
that are to allocated differently. PFS=Select All, PF6=Deselect All
#Partitions: 5
DB2 CREATE TABLESPACE Partition attributes. 5 Rows
Ptn Sel Using StoGroup Pri (KB) Sec (KB) Erase Free Free Track Comp GBP
Page Pcnt Cache
<--> - <----> <----> <----> <----> <--> <--> <--> <--> <--> <---->
*** Top of Data ***
 1 - STOGROUP DEMOSG -1 -1 No 0 5 Yes No Changed
 2 - STOGROUP DEMOSG -1 -1 No 0 5 Yes No Changed
 3 - STOGROUP DEMOSG -1 -1 No 0 5 Yes No Changed
 4 - STOGROUP DEMOSG -1 -1 No 0 5 Yes No Changed
 5 - STOGROUP DEMOSG -1 -1 No 0 5 Yes No Changed
*** End of Data ***

```

Figure 187. 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.

```

SELCOPY/i - DB2(CBLA): Create Tablespace
File Help                               wS wR
Command>                               Scroll> Csr
ZZS2CTS0                               Lines 1-20 of 21
View: Table Space Options (1/2)         Tablespace: USERTS01 In Database: DEMODB

Logging:
  / Record changes to data in the DB2 log.

Priority in which table data sets are closed when open threshold reached:
  / Close first (CLOSE YES)             _ Close after CLOSE YES data sets (CLOSE NO)

Lock Size:          Size of locks in the tablespace or lock escalation threshold.
  / Any size selected by DB2           _ Tablespace      _ Page      _ Row

Lock Maximum:      Maximum number of simultaneous locks before escalation occurs.
  / Use system defined threshold.      (DSNTIPJ specified LOCKD PER TABLE(SPACE))
  _ Use user defined threshold>       @          0 => locks not counted.

Data Sharing Group Buffer Pool Caching:
  / Changed data pages                  _ All data pages read          _ No caching

1. Help (PF1)          2. Next (ENTER)          3. Back (PF3)          4. Exit (PF15)
  
```

Figure 188. 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.

```

SELCOPY/i - DB2(CBLA): Create Tablespace
File Help                                     wS wR
Command>                                     Scroll> Csr
ZZS2CTS0                                     Lines 1-20 of 21
View: Table Space Options (2/2)             Tablespace: USERTS01 In Database: DEMODB

Table Space Row Data:
Maximum Rows> 255                          Maximum number of rows per page. (1-255)
Compress Rows> _ Yes / No                  Perform data compression on table space rows.

Free Space:
Free page frequency> 0                      Free page after n pages. (0-255, 0=>No free)
Percent free/page > 5                      Percentage of each page left free on LOAD.

Table Space string data default CCSID Encoding Scheme:
_ UNICODE / EBCDIC _ ASCII                 Default set as database default.

Table Space Page Modification:
_ Do not track pages which have been changed in the space map.

1. Help (PF1)           2. Next (ENTER)       3. Back (PF3)         4. Exit (PF15)

```

Figure 189. 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.

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 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 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 proceeding 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.



```

SELCOPY/i - DB2(CBLA): Create Table
File Help                               wS wR
Command>                                Scroll> Csr
ZZS2CT00                                Lines 1-20 of 21
View: Table Name & Location              Action: Create Base Table
Enter a new table name and optionally a table space at the current server.

A blank TableSpace value will create the table in an implicitly defined
tablespace in the named (or a default) database.
A blank Database value with a non-blank TableSpace will use that explicitly
defined tablespace in the default database DSNDB04. If an explicitly named
tablespace does not already exist then a TableSpace error is returned.

Use wildcard character "%" in the database and/or tablespace field values
to select from a list of filtered, eligible entries.

Owner> _____ + (optional)
Name> _____ +

DataBase> _____ (optional) TableSpace> _____ (optional)
TSType: Implicitly defined. (Segmented or Universal Partitioned)

1. Help (PF1)      2. Next (ENTER)      3. Back (PF3)      4. Exit (PF15)

```

Figure 190. 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 **DSNnnnnn**. 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 DSNnnnnn) 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 mKB 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 mGB)**

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 mGB)**

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.

```

SELCOPY/i - DB2(CBLA): Create Table
File Help                               wS wR
Command>                               Scroll> Csr
ZZS2CT00                               Lines 1-20 of 21
View: Table Definition                   Table Owner:
Select the method by which the table and its columns will be defined.

/ Do not model on an existing table.
- Model on a single table or view using LIKE.
- Model on a generated result table.
- Model on one or more tables/views for panel field load.

Table Options:                          (Options affecting column definition.)
EDITPROC> _____ Transform row data procedure. (optional)

Use of a table edit procedure to transform table row data on execution
of SQL data change, LOAD or row retrieval operations, prevents definition of
LOB, XML, ROWID, Identity and Security Label columns within the new table.

1. Help (PF1)          2. Next (ENTER)          3. Back (PF3)          4. Exit (PF15)

```

Figure 191. 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 existence of a primary key constraint are updated accordingly.


```

SELCOPY/i - DB2(CBLA): Create Table
File Help                                     wS wR
Command>                                     Scroll> Csr
ZZS2CT00                                     Lines 1-20 of 21
View: Columns & Constraints                   Table Owner: + Name: NBJ2TAB +
Select the column and constraint definition sub-panel to be opened next.
If no options are selected, processing continues at the next Create Table
panel view.

If no column definitions have been entered (either explicitly or implicitly
via a model table/view), the Column Definitions sub-panel will be opened
regardless of the current selection.

On return from a sub-panel, the next table component in the list will be
selected by default.

Table Column and Constraint Definitions:
- Column Definitions.                #columns:      0                (mandatory)
- Primary Key Constraint.            primary key:   No                (optional)
- Unique Constraints.                #unique keys:  0                (optional)
- Referential Constraints.           #foreign keys: 0                (optional)
- Check Constraints.                 #checks:      0                (optional)

1. Help (PF1)          2. Next (ENTER)      3. Back (PF3)       4. Exit (PF15)

```

Figure 192. 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 **SELECTAIL** (assigned to <PF6> 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 (default <PF2>) and, if necessary, the appropriate field **expanded** (default <PF2>) to accommodate the input value.

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.

```

SELPCOPY/i - DB2(CBLA): Create Table - Column Definitions
File Help
Command>
ZZS2CT01
Define Table Columns: Table Owner: + Name: NBJ2TAB +
Insert column entries and select a column type (Built-in type, Distinct type,
Identity Column, Security Label, Row Change Timestamp.) ZOOM a row to expand
field entries and display additional options applicable to the column type.
PF6=Select/Deselect ALL
SQL CREATE TABLE Column Specification. 16 Rows
Sel Column Name Type Len/Pr Sc Def Default Value / Field Proc
- <---+---1---+---> <---+---> <---+> <> <> <---+---1---+---2---+--->
000 *** Top of Data ***
001 IQ CHAR 8 0 NX
002 AMKEY CHAR 7 0 NX
003 TYPE CHAR 1 0 NX
004 OPSYS CHAR 3 0 ND 'ZOS'
005 COMPONENT VARCHAR 12 0 DN
006 STATUS CHAR 1 0 ND 'N'
007 DESCR VARCHAR 32000 0 DN
008 KEYWORDS VARCHAR 50 0 DN
009 CREDATE DATE 4 0 ND
010 CETIME TIME 3 0 ND

```

Figure 193. 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.

TIMESTAMP

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 which 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 display the sub-panel.

In zoomed view, enter "/" (slash) or any non-blank character in this field and press <Enter> to display 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 column 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 <PF6>) 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.

```

SELCPY/i - DB2(CBLA): Column FIELDPROC Parameters
File Help                                     wS wR
Command>                                     Scroll> Csr
ZZS2CT02
Column Name: IQ                               + Table Owner:           + Name: NBJ2TAB           +
Each row identifies a single parameter and its position in the parameter list.
Field Procedure Parameters.                   4 Rows
Constants
<---+---1---+---2---+---3---+---4---+---5---+---6---+---7---
000000 *** Top of Data ***
000001 '%Date%'
000002 100001
000003 200000
000004 'Encrypt-200'
000005 *** End of Data ***

```

Figure 194. 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 <PF6> by default). This toggles between selection and deselection of all entries in the table.
- **RESTORETab** (assigned to <PF5> 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.

```

SELPCOPY/i - DB2(CBLA): Create Table - Primary Key Definition
File Help                               wS wR
Command>                                Scroll> Csr
ZZS2CT03
Primary Key Constraint:                   Table Owner:           + Name: NBJ2TAB +
Name> IQ-PKEY                             + (optional)

Columns>
The table below lists all new column definitions that are eligible for use
as a Primary Key column. Delete and/or rearrange the columns to define the
primary key columns in the required order of precedence.
PF5=Restore Columns, PF6=Select/Deselect ALL 8 Rows
Primary Key Constraint Columns.
Sel Column Name      Type      Len/Pr  Sc  DistSchema      DistName
- <---+---1---+> <---+--> <---+> <> <---+---1---+> <---+---1---+>
000000 *** Top of Data ***
000001 S IQ CHAR 8 0
000002 - AMKEY CHAR 7 0
000003 - TYPE CHAR 1 0
000004 - OPSYS CHAR 3 0
000005 - STATUS CHAR 1 0
000006 - CREDATE DATE 4 0
  
```

Figure 195. 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 <PF6> 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.

```

SELCOPY/i - DB2(CBLA): Create Table - Unique Constraint Definitions
File Help                               wS wR                               Scroll> Csr
Command>
ZZS2CT04
Unique Constraints:                       Table Owner:                       + Name: NBJ2TAB                       +
Each entry in the table below represents a Unique constraint definition.
Optionally enter a unique constraint name then select the columns to be
assigned to the unique key.
PF6=Select/Deselect ALL
Unique Constraints.                         PF6=Select/Deselect ALL
Sel Constraint Name (optional)             Key Columns                          2 Rows
- <---+---1---+---2---+--->             <---+---1--->
000000 *** Top of Data ***
000001 S  SYSMOD                          > 1 specified
000002 S  SYSMODX                          > 2 specified
000003 *** End of Data ***

```

Figure 196. 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:

- **SELECTAIL** (assigned to <PF6> by default). This toggles between selection and deselection of all entries in the table.
- **RESTORETab** (assigned to <PF5> 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.

```

SELCOPY/i - DB2(CBLA): Create Table - Unique Constraint Columns
File Help
Command>
ZZS2CT05
Unique Key Columns:
Constraint Name> SYSMODX
Table Owner: Sel> / + Name: NBJ2TAB +
Add this constraint. (optional)
Columns>
The table below lists all new column definitions that are eligible for use
as a Unique Key column. Delete and/or rearrange the columns to define the
unique key columns in the required order of precedence.
PF5=Restore Columns, PF6=Select/Deselect ALL 2 Rows
Unique Constraint Columns.
Sel Column Name Type Len/Pr Sc DistSchema DistName
- <---+---1---+> <---+--> <---+> <> <---+---1---+> <---+---1---+>
000000 *** Top of Data ***
000001 S SYSMOD CHAR 7 0 SYSIBM CHAR
000002 S IQ CHAR 8 0 SYSIBM CHAR
000003 *** End of Data ***

```

Figure 197. 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 displays UP, DOWN, LEFT and RIGHT.

In addition to the standard primary table commands, this sub-panel also supports primary command **SELECTALL** (assigned to <PF6> 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.

```

SELCPY/i - DB2(CBLA): Create Table - Referential Constraint Definitions
File Help                               WS WR                               Scroll> Csr
Command>
ZZS2CT06
Referential Constraints:                 Table Owner:                   + Name: NBJ2TAB +
Entries in the table below represent Referential constraint definitions.
Optionally enter a unique constraint name and ZOOM the relevant column entry
to configure the Parent/Foreign key association.                                PF6=Select/Deselect ALL
Referential Constraints.                                                         1 Row
Sel Constraint Name                    Parent Key          Foreign Key          Delete Enf
(optional)                            Columns            Columns              Action           Action
- <---+-----1-----2--> <---+-----1---> <---+-----1---> <---+---> <-->
000000 *** Top of Data ***
000001 S FMID                          > 1 specified      > 1 specified      RESTRICT Yes
000002 *** End of Data ***

```

Figure 198. 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 display 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 display 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. Alternatively, 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.

```

SELCPY/i - DB2(CBLA): Create Table - Referential Constraint Parent Key Colum
File Help                               wS wR
Command>
ZZS2CT08                                Scroll> Csr
Parent Table Definition:
Owner> ZZS
Name> ZZSFMID
Select Parent Key Columns:
Select a Primary/Unique key constraint assigned to the Parent Table to
identify the parent key columns. Use the wild card character (%) to select
the constraint name from a list of possible entries.
Constraint Name> FMID
Parent Key Columns assigned to the Primary/Unique Key constraint.
Parent Key      Key   Data      Len/Pr  Sc  Distinct  Distinct  Distinct  1 Row
Column          Col#  Type                               Schema    Name      Type
<---+---1--->  <>  <---+--->  <--->  <>  <---+---+  <---+---+  <---+---+
000000 *** Top of Data ***
000001 FMID      1  CHAR              7  0  SYSIBM          CHAR
000002 *** End of Data ***

```

Figure 199. 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 **RESTORETab** (assigned to <PF5> 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.

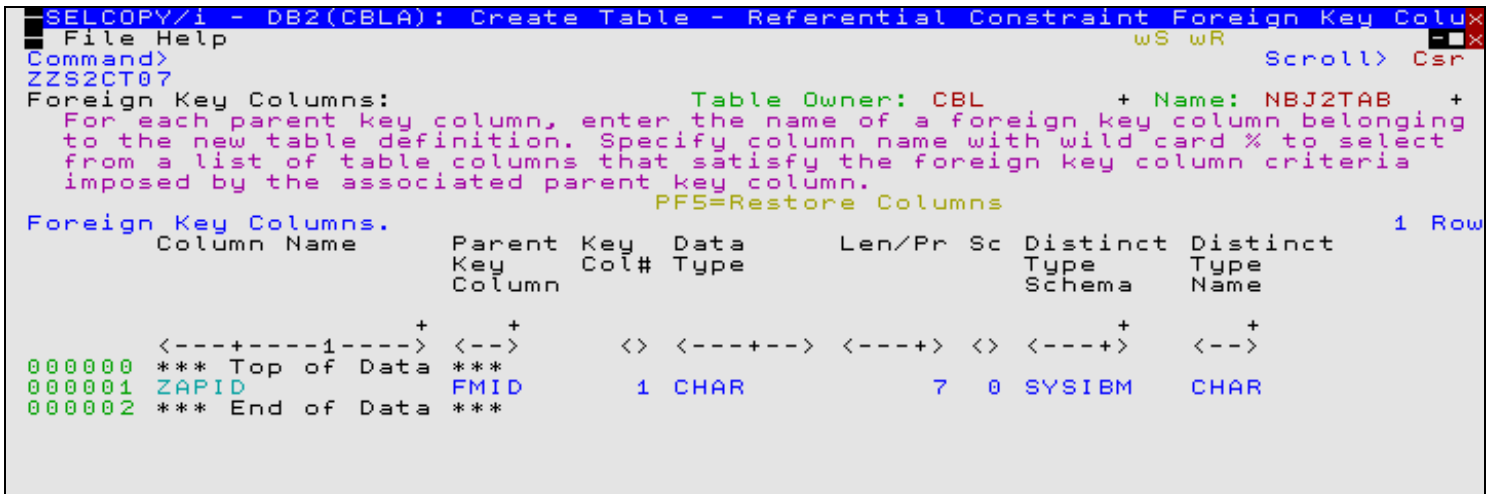


Figure 200. DB2: Create Table - Referential Constraint Foreign Key Columns.

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.


```

SELCPY/i - DB2(CBLA): Create Table - Check Constraint Definitions
File Help                               wS wR                               Scroll> Csr
Command>
ZZS2CT09
Check Constraints:                       Table Owner:                       + Name: NBJ2TAB +
Each entry in the table below represents a Check constraint definition.
Optionally enter a check constraint name then enter the check condition
for columns in the new table definition.

Check Constraints.                               3 Rows
Sel Constraint Name Check Condition (SQL search condition)
(optional)
- <---+-----1-----> <---+-----1-----+-----2-----+-----3-----+-----4-----+----->
000000 *** Top of Data ***
000001 S OPSYS OPSYS='CMS' OR OPSYS='VSE' OR OPSYS='ZOS' OR OPSYS='ZOS' OR OPSYS='ZOS' OR OPSYS='ZOS'
000002 S STATUS STATUS='N' OR STATUS='I' OR STATUS='F'
000003 S TYPE TYPE='A' OR TYPE='B' OR TYPE='D' OR TYPE='L' OR TYPE='L' OR TYPE='L'
000004 *** End of Data ***

```

Figure 201. 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 new 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.

```

SELCOPY/i - DB2(CBLA): Create Table
File Help                               wS wR
Command>                                Scroll> Csr
ZZS2CT00                                Lines 1-20 of 21
View: Model using LIKE                   + Name: NBJ2TAB +
Enter model table/view name and copy column attributes for the
CREATE TABLE LIKE operation.

Use wildcard character "%" in the model table/view owner and/or name field
values to select from a list of filtered, eligible tables/views.

Model Table/View:
Owner>      ZZS                               + (optional)
Name>      ZZSIQ                               +

Copy Column Attributes:
Identity Column>      / Exclude      - Include
Row Change Timestamp Column>  / Exclude      - Include

1. Help (PF1)          2. Next (ENTER)          3. Back (PF3)          4. Exit (PF15)

```

Figure 202. 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.

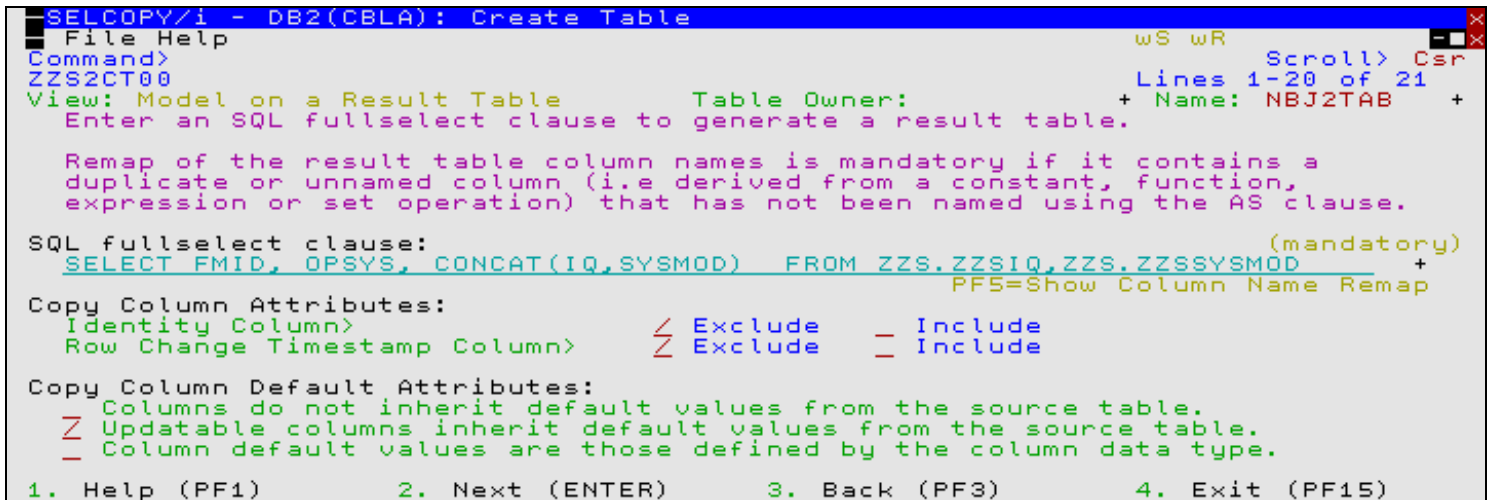
Model on Result Table

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 <PF5>) 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.



```

SELCOPY/i - DB2(CBLA): Create Table
File Help
Command>
ZSS2CT00
View: Model on a Result Table Table Owner:
Enter an SQL fullselect clause to generate a result table.

Remap of the result table column names is mandatory if it contains a
duplicate or unnamed column (i.e derived from a constant, function,
expression or set operation) that has not been named using the AS clause.

SQL fullselect clause:
SELECT FMID, OPSYS, CONCAT(IQ,SYSMOD) FROM ZSS.ZZSIQ,ZSS.ZZSSYSMOD
(mandatory)
PF5>Show Column Name Remap

Copy Column Attributes:
Identity Column> / Exclude - Include
Row Change Timestamp Column> / Exclude - Include

Copy Column Default Attributes:
Z Columns do not inherit default values from the source table.
Z Updatable columns inherit default values from the source table.
- Column default values are those defined by the column data type.

1. Help (PF1) 2. Next (ENTER) 3. Back (PF3) 4. Exit (PF15)

```

Figure 203. 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 <PF2>) 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.

```

SELCOPY/i - DB2(CBLA): Result Table Column Name Remap
File Help
Command>
ZVS2CT10
Column Name Remap:
Remap column name must exist for each column selected by the SQL FullSelect
statement. Column names are remapped in the order specified.
FullSelect SQL: SELECT FMID, OPSYS, CONCAT(IQ,SYSMOD) FROM ZVS.ZVSIQ,ZZ
Results Table Column Map.
New Column Name
Old Column Name
<---+---1---+---2---+--->
000000 *** Top of Data ***
000001 FUNC
000002 SYSTEM
000003 FIXREF
000004 *** End of Data ***

```

Figure 204. 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.

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.

```

SELCOPY/i - DB2(CBLA): Create Table
File Help                               wS wR
Command>                               Scroll> Csr
ZZS2CT00                               Lines 1-20 of 21
View: Table Modelling                   + Name: NBJ2TAB +
Select an existing table/view in the current server on which to model the
new table. Use "%" wildcard for table/view selection.

Model Table/View:
Owner> ZZS                               + (optional)
Name> ZZSIQ                               +

Model Table/View Panel Field Load Options:
Enter "/" to identify the panel fields to be loaded from the model.
/ Database Name                          / Primary key constraint      _ Misc Table Options
/ TableSpace Name                        / Unique key constraints
/ Column Definitions                     / Referential constraints
_ EDIT/VALIDATE Proc                     / Check constraints

Action:
Add modelled values to existing panel fields.
Z Replace values in existing panel fields.

1. Help (PF1)          2. Next (ENTER)      3. Back (PF3)        4. Exit (PF15)

```

Figure 205. 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.

Owner>

The model DB2 table or view owner id.

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.

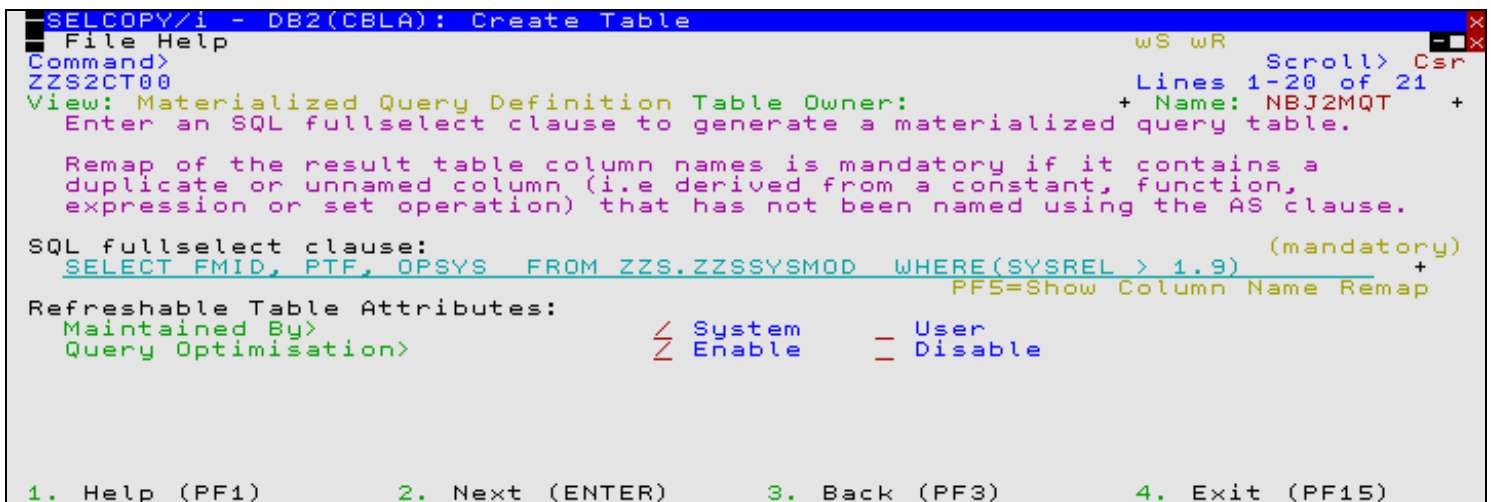
Materialized Query Definition

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 <PF5>) 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.



```

SELCOPY/i - DB2(CBLA): Create Table
File Help                               wS wR
Command>                                Lines 1-20 of 21
ZZS2CT00                                + Name: NBJ2MQT +
View: Materialized Query Definition Table Owner:
Enter an SQL fullselect clause to generate a materialized query table.

Remap of the result table column names is mandatory if it contains a
duplicate or unnamed column (i.e derived from a constant, function,
expression or set operation) that has not been named using the AS clause.

SQL fullselect clause:
SELECT FMID, PTF, OPSYS FROM ZZS.ZZSSYSMOD WHERE(SYSREL > 1.9) (mandatory)
                                                                    PF5>Show Column Name Remap
Refreshable Table Attributes:
Maintained By>                               / System      - User
Query Optimisation>                          Z Enable      = Disable

1. Help (PF1)           2. Next (ENTER)       3. Back (PF3)         4. Exit (PF15)

```

Figure 206. 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 <PF2>) 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 that 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.

```

SELCOPY/i - DB2(CBLA): Create Table
File Help                               wS wR
Command>                               Scroll> Csr
ZS2CT00                                Lines 1-20 of 21
View: Implicit TableSpace Options      Table Owner: CBL      + Name: NBJ2TAB  +
Enter options applicable to the implicitly defined tablespace.

TableSpace Type:
 / Mode default. New-function=>Partition-by-Growth, Conversion=>Segmented.
- Universal Partition-by-Range. (Uses table-controlled partitioning.)
 #Partitions>      1      (1-4096) Partition Size is based on this value.
- Universal Partition-by-Growth.
 Partition Size>    0 GB   (0-64 where 0=>default) TableSpace DSSIZE.

String Data CCSID Encoding Scheme:
- UNICODE          / EBCDIC          - ASCII          DataBase:

Default CCSID value has either been set by a model table/view, implied by
the selected database or is equal to the DB2 sub-system default (DSNTIPF).

1. Help (PF1)          2. Next (ENTER)          3. Back (PF3)          4. Exit (PF15)
  
```

Figure 207. 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.)

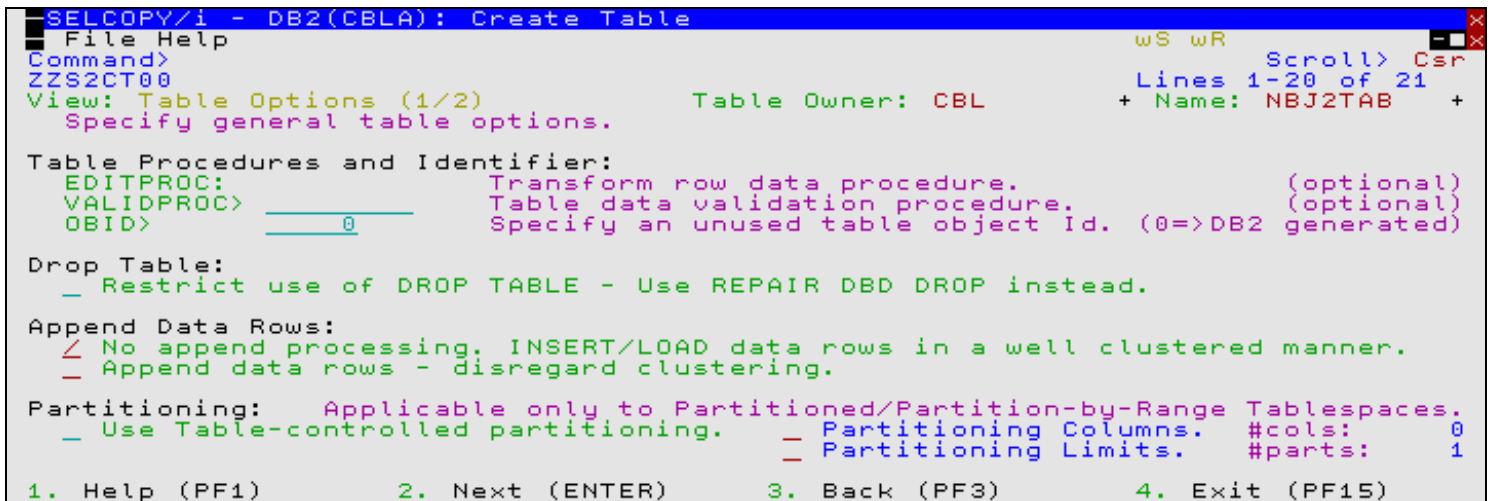
Note that changing the encoding scheme may invalidate column definitions that have been already been added to the **Column Definitions** sub-panel.

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 implicit tablespace options has been completed.



```

SELCOPY/i - DB2(CBLA): Create Table
File Help                               wS wR
Command>                                Scroll> Csr
ZZS2CT00                                Lines 1-20 of 21
View: Table Options (1/2)                Table Owner: CBL
Specify general table options.           + Name: NBJ2TAB +

Table Procedures and Identifier:
EDITPROC>                               Transform row data procedure.           (optional)
VALIDPROC> _____                   Table data validation procedure.       (optional)
OBID> _____                         Specify an unused table object Id. (0=>DB2 generated)

Drop Table:
_ Restrict use of DROP TABLE - Use REPAIR DBD DROP instead.

Append Data Rows:
_ / No append processing. INSERT/LOAD data rows in a well clustered manner.
_ Append data rows - disregard clustering.

Partitioning: Applicable only to Partitioned/Partition-by-Range Tablespaces.
_ Use Table-controlled partitioning.     _ Partitioning Columns. #cols: 0
                                           _ Partitioning Limits. #parts: 1

1. Help (PF1)      2. Next (ENTER)    3. Back (PF3)     4. Exit (PF15)

```

Figure 208. 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 value 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:

- **SELECTAIL** (assigned to <PF6> by default). This toggles between selection and deselection of all entries in the table.
- **RESTORETab** (assigned to <PF5> 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*.

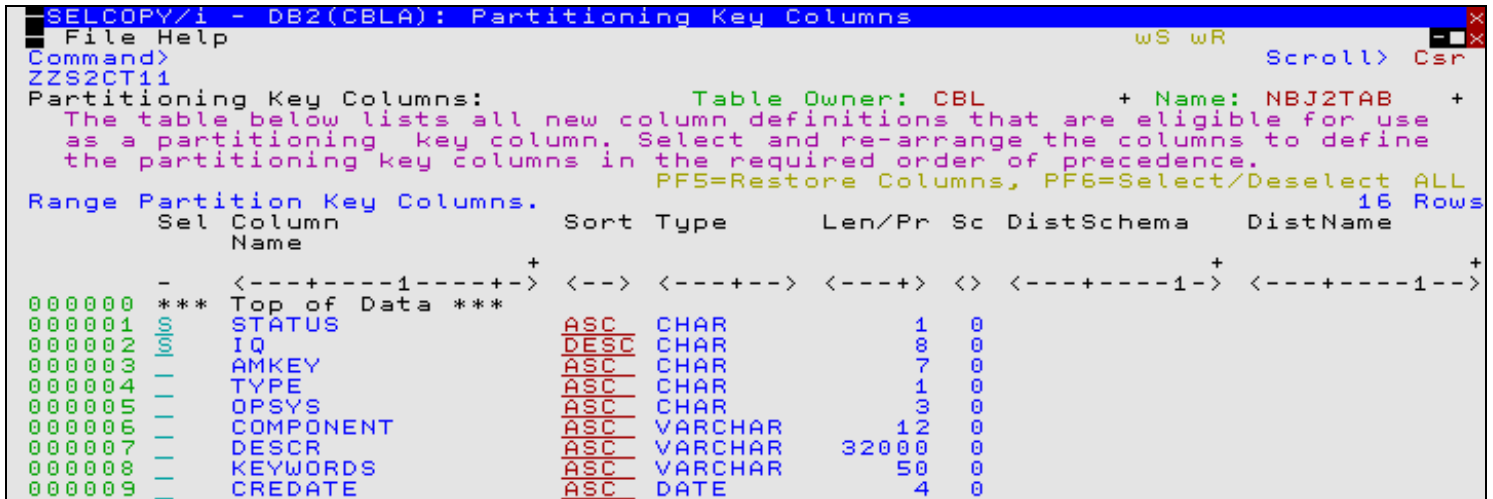


Figure 209. DB2: Create Table - Partitioning Key Columns.

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.

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 positive 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 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.

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*.

```

SELCOPY/i - DB2(CBLA): Tablespace Range partitions
File Help
Command>
ZZS2CT12
TableSpace Partitions: Table Owner: CBL Name: NBJ2TAB
Each entry in the table below represents a table partition definition.
For each partition, enter column values that define the partition limit.

Range Partitions. 5 Rows
PtnNo Partition Column
Limits
<> <---+---1---+---2---+---3---+---4---+---5---+---6---+---7--->
** Top of Data ***
1 > 0 specified
2 > 0 specified
3 > 0 specified
4 > 0 specified
5 > 0 specified
** End of Data ***

```

Figure 210. 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 display 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 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.

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 (default <PF2>) and, if necessary, the appropriate field **expanded** (default <PF2>) to accommodate the input value.

Corresponds to SQL CREATE TABLE syntax ENDING AT (*constant*, MAXVALUE, or MINVALUE, ...).

```

SELCOPY/i - DB2(CBLA): Define Partitioning Key Column Values
File Help                                     wS wR
Command>                                     Scroll> Csr
ZZS2CT13
Partitioning Key Column Values:               Table Owner: CBL      + Name: NBJ2TAB      +
Enter limit values for each column in partitioning key. Enter MAXVALUE or
MINVALUE to indicate the maximum or minimum possible value for the column.

Partition Number:      1

Range Partitioning Column Limits.
Limit Value  Column Name  Type      Len/Pr  Sc  Sort  DistSchema  DistName  2 Rows
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
00 *** Top of Data ***
01 'N'          STATUS      CHAR      1      0  ASC
02 'IQ003001'  IQ          CHAR      8      0  DESC
03 *** End of Data ***
  
```

Figure 211. 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.

Data Type: (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.

```

SELCPY/i - DB2(CBLA): Create Table
File Help
Command>
ZZS2CT00
View: Table Options (2/2) Table Owner: CBL Name: NBJ2TAB
Specify general table options.

Audit Table Access: (Requires appropriate Trace Class to be active)
  / No auditing to occur when the table is accessed. (NONE)
  - Auditing occurs for first insert, update, delete. (CHANGES)
  - Auditing occurs for first operation of any utility/application. (ALL)

DB2 SQL Table Access:
  / Base access on current statistics. (NOT VOLATILE)
  - Index access is used whenever possible. (VOLATILE)

Data Capture Logging:
  / No additional logging for actions on the table. (NONE)
  - Log additional information for table data/column changes. (CHANGES)

1. Help (PF1) 2. Next (ENTER) 3. Back (PF3) 4. Exit (PF15)
  
```

Figure 212. 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.

```

SELCOPY/I - DB2(CBLA): Create Table
File Help SQL JCL Command
Command>
ZZS2CT00
View: Generate SQL Table Owner: CBL + Name: NBJ2TAB +
Select action to be performed on the generated SQL CREATE TABLE statement.
For each option, the utility used to run the SQL is shown in parentheses.
If SQL is copied to a file, enter the fileid below. If to a new data set, a
Model DSN may be specified for allocation of the new data set. In fileid
fields, wildcard characters "*", "**", "%" may be used to select from a list.

Action:
/ Copy to a file (see below) and display it in a text edit view. (EXECSQL)
- Display an in-storage (temporary) copy in a text edit view. (EXECSQL)
- Display as an executable line command. (CMDTEXT - PF4).
- Display within generated batch JCL. (SUBMIT)
- Execute immediately.

SQL Output File: (Library member, Sequential data set or HFS path)
DSN/Path> USEROBJ.SELCOPYI.SQL.SOURCE + Member> NBJ2TAB
Model DSN> USEROBJ.SQL.JCL (optional)
Append> / Select this option to append SQL to existing file content.

1. Help (PF1) 2. Action (ENTER) 3. Back (PF3) 4. Exit (PF15)

```

Figure 213. 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 organisation 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 View

The Create View panel (ZZS2CVI0) is an **interactive panel window**, opened on selection of option 4. in the Create DB2 Object options menu. This panel may be used to create a DB2 view on tables or views in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE VIEW statement in the SELCOPY/i foreground.

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**
Executes the SQL CREATE VIEW generated by the specified input field parameters. Hitting <Enter> will perform the same action.
- Command**
Opens a text edit view for a temporary data set containing SQL command CREATE VIEW syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL CREATE VIEW statement syntax generated for the selected input field values.
- Help**
Open the general help for the Create View panel.

Field Entries

- View:**
Fields that together identify the DB2 view name to be created at the current server location. The view name must not match the name of an existing table, view alias or synonym.
- Owner>**
The owner (schema) of the new view.
The view owner value has a maximum length of 128 characters.
- Name>**
The name of the new view.
The view name has a maximum length of 128 characters.
- This parameter field corresponds to SQL CREATE VIEW parameter *view-name*.
- Full select**
Specifies an SQL fullselect that defines the view. At any time, the 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*.
- Common table expression**
Specifies a common table expression, for use with the full select part of the view definition.
This parameter field is optional and corresponds to SQL CREATE VIEW parameter WITH *common-table-expression*.
- Column names**
Specifies a comma separated list of column names that are used to name each column in the view. If specified, number of column names must match the number of columns returned by the results table of the fullselect.
If not specified, the columns names of the view inherit those of the results table generated by the fullselect.
This parameter field corresponds to SQL CREATE VIEW parameter *column-name*.
- With Check Option:**
Determines whether or not a check that an inserted or updated row conforms to the definition of the view is performed and, if so, the extent of the checking performed.
- ◇ **Default**
The definition of the view is **not** used to perform checks on uninserted or updated rows.
 - ◇ **Cascaded**
Updated and inserted rows must satisfy the search conditions of the view and all underlying views regardless of whether those underlying 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 view and all underlying views that have been

defined with the CHECK option.
This option corresponds to SQL CREATE VIEW parameter WITH LOCAL CHECK OPTIONS.

Create Alias

The Create Alias panel (ZZS2CAL0) is an **interactive panel window**, opened on selection of option 5. in the Create DB2 Object options menu. This panel may be used to create in the current DB2 subsystem, a DB2 alias for a table or view.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE ALIAS statement in the SELCOPY/i foreground.

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** Executes the SQL CREATE ALIAS generated by the specified input field parameters. Hitting <Enter> will perform the same action.
- Command** Opens a text edit view for a temporary data set containing **SQL command** CREATE ALIAS syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL CREATE ALIAS statement syntax generated for the selected input field values.
- Help** Open the general help for the Create Alias panel.

Field Entries

Alias: Fields that together identify the DB2 alias name to be created at the current server location. The alias name must not match the name of an existing table, view alias or index.

Owner>
The owner (schema) of the new alias.
The alias owner value has a maximum length of 128 characters.

Name>
The name of the new alias.
The 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 for which the alias is defined.

Location>
The DBMS location where the table or view is defined.
The table or view location name has a maximum length of 16 characters.

Owner>
The owner (schema) of the table or view.
The table or view owner name has a maximum length of 128 characters.

Name>

The SQL identifier name of the table or view.
The table or view name has a maximum length of 128 characters.

This parameter field corresponds to SQL CREATE ALIAS parameter FOR *table-name* or FOR *view-name*.

Create Synonym

The Create Synonym panel (ZZS2CSY0) is an **interactive panel window**, opened on selection of option 6. in the Create DB2 Object options menu. This panel may be used to create a DB2 synonym for a table or view in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE SYNONYM statement in the SELCOPY/i foreground.

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** Executes the SQL CREATE SYNONYM generated by the specified input field parameters. Hitting <Enter> will perform the same action.
- Command** Opens a text edit view for a temporary data set containing **SQL command** CREATE SYNONYM syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL CREATE SYNONYM statement syntax generated for the selected input field values.
- Help** Open the general help for the Create Synonym panel.

Field Entries

Synonym: Specifies the DB2 synonym name to be created at the current server location. The synonym name must not match the name of an existing table, view, alias or synonym owned by the user.
The synonym name has a maximum length of 128 characters.

This parameter field corresponds to SQL CREATE SYNONYM parameter *synonym*.

Table, View or Alias Name: Fields that together identify the DB2 table, view or alias at the current server for which the synonym is defined.

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.

These parameter fields are mandatory and correspond to SQL CREATE SYNONYM parameter FOR *authorisation-name.table-name* or FOR *authorisation-name.view-name*.

Create Distinct Type

The Create Distinct Type panel (ZZS2CTY0) is an **interactive panel window**, opened on selection of option 7. in the Create DB2 Object options menu. This panel may be used to create a DB2 distinct type for a specific data type.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE TYPE statement in the SELCOPY/i foreground.

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** Executes the SQL CREATE TYPE generated by the specified input field parameters. Hitting <Enter> will perform the same action.
- Command** Opens a text edit view for a temporary data set containing **SQL command** CREATE TYPE syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL CREATE TYPE statement syntax generated for the selected input field values.
- Help** Open the general help for the Create Distinct Type panel.

Field Entries

Distinct Type: Fields that together identify the DB2 distinct type name to be created at the current server location. The distinct type name must not match the name of an existing distinct type, built-in type, BOOLEAN or a system reserved keyword.

Schema>
The schema of the new distinct type.
The schema name has a maximum length of 128 characters.

Name>
The SQL identifier name of the distinct name.
The distinct name has a maximum length of 128 characters.

These parameter fields are mandatory and correspond to SQL CREATE TYPE parameter *distinct-type-name*.

Select Both Source Field Type and Specific Data Type:
Fields that together identify the DB2 source built-in data type. Mutually exclusive options are as follow:

◇ **Signed Numeric Field Type**
Built-in type is numeric. Mutually exclusive options are as follow:

- SMALLINT
- INTEGER
- BIGINT
- FLOAT SINGLE
- FLOAT DOUBLE
- DECFLOAT(16)
- DECFLOAT(34)
- DECIMAL

Precision>
Applicable only to built-in data type DECIMAL, specifies the precision (total number of digits) supported by the decimal field. This value may be 1 to 31.

Scale>

Applicable only to built-in data type DECIMAL, specifies the scale (number of digits to the right of the decimal point) supported by the decimal field. This value may be 0 to value of the precision field.

• String Field Type

Built-in type is non-numeric data. Mutually exclusive options are as follow:

- ◆ CHARACTER
- ◆ VARCHAR
- ◆ CLOB
- ◆ GRAPHIC
- ◆ VARGRAPHIC
- ◆ DBCLOB
- ◆ BINARY
- ◆ VARBINARY
- ◆ BLOB

Length>

Specifies the fixed or maximum length of the data field. Maximum value for each data type is as follow:

CHARACTER	255
VARCHAR	Max row size - 8
CLOB	2147483647, 2097152K, 2048M or 2G
GRAPHIC	127
VARGRAPHIC	(Max row size - 2) / 2
DBCLOB	1073741823, 1048575K, 1024M or 1G
BINARY	255
VARBINARY	Max row size - 8
BLOB	2147483647, 2097152K, 2048M or 2G

Data SubType:

Applicable only to built-in data types CHARACTER, VARCHAR and CLOB, specifies the character data sub-type. Mutually exclusive options are as follow:

- **Default** - Use the default subtype for the CCSID encoding.
- **SBCS** - Single-byte data.
- **MIXED** - Mixed data.
- **BIT** - Bit data. (Not valid for CLOB).

Data Encoding:

Applicable only to built-in data types SBCS/MIXED CHARACTER or GRAPHIC, specifies the encoding scheme for the distinct type. Mutually exclusive options are as follow:

- **Default** - Default encoding scheme.
- **EBCDIC**
- **ASCII**
- **UNICODE**

• Date/Time Field Type

Built-in type is a date, time or timestamp. Mutually exclusive options are as follow:

- ◆ DATE
- ◆ TIME
- ◆ TIMESTAMP

• Row Identifier Type

Built-in type is a row id. (**ROWID**)

Create Trigger

The Create Trigger panel (ZZS2CTR0) is an **interactive panel window**, opened on selection of option 8. in the Create DB2 Object options menu. This panel may be used to create a trigger in a schema and build a trigger package in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE TRIGGER statement in the SELCOPY/i foreground.

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
Executes the SQL CREATE TRIGGER generated by the specified input field parameters. Hitting <Enter> will perform the same action.

Command
Opens a text edit view for a temporary data set containing SQL command CREATE TRIGGER syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL CREATE TRIGGER statement syntax generated for the selected input field values.

Help
Open the general help for the Create Trigger panel.

Field Entries

Trigger: Fields that together identify the DB2 trigger name to be created at the current server location. The trigger name must not match the name of an existing trigger.

Schema>
The schema of the new trigger.
The schema name has a maximum length of 128 characters.

Name>
The SQL identifier name of the trigger name.
The trigger name has a maximum length of 128 characters.

These parameter fields are mandatory and correspond to SQL CREATE TRIGGER parameter *trigger-name*.

Order: Identifies when the trigger will be implemented when invoked. Mutually exclusive options are as follow:

- ◇ **Before (No cascade)**
Action is triggered before any insert/delete/update action on the subject table. No other triggers are activated since the triggered action cannot include updates.
This option corresponds to SQL CREATE TRIGGER parameter NO CASCADE BEFORE.
- ◇ **After**
Action is triggered after any insert/delete/update action on the subject table.
This option corresponds to SQL CREATE TRIGGER parameter AFTER.
- ◇ **Instead**
Action is triggered instead of any insert/delete/update action on the subject view.
This option corresponds to SQL CREATE TRIGGER parameter INSTEAD OF.

Type: Identifies the type of trigger. Mutually exclusive options are as follow:

- ◇ **Insert**
Action is triggered whenever there is an insert operation on the subject table.
This option corresponds to SQL CREATE TRIGGER parameter INSERT.
- ◇ **Update**
Action is triggered whenever there is an update operation on the subject view.
This option corresponds to SQL CREATE TRIGGER parameter UPDATE.
- ◇ **Delete**
Action is triggered whenever there is a delete operation on the subject table.
This option corresponds to SQL CREATE TRIGGER parameter DELETE.

Executed: Specifies the conditions for which DB2 executes the triggered action. Mutually exclusive options are as follow:

◇ **For each row**

Action is triggered for each row of the subject table/view modified by the triggering operation. This option corresponds to SQL CREATE TRIGGER parameter FOR EACH ROW.

◇ **For each statement**

Action is triggered once only by the triggering operation. This option corresponds to SQL CREATE TRIGGER parameter FOR EACH STATEMENT.

Update Columns:

Applicable to trigger type **UPDATE** only, optionally specifies that activation of the trigger should be restricted only to operations that update one or more of these named columns. Column names must be comma separated. This parameter field corresponds to SQL CREATE TRIGGER parameter UPDATE OF *column-name*.

Subject Table/View:

Fields that together identify the subject DB2 base table of a BEFORE or AFTER trigger definition, or the subject DB2 view of an INSTEAD OF trigger definition.

Owner>

The owner (schema) of the table or view.
The table or view owner name has a maximum length of 128 characters.

Name>

The SQL identifier name of the table or view.
The table or view name has a maximum length of 128 characters.

These parameter fields are mandatory and correspond to SQL CREATE TRIGGER parameter ON *table-name* or ON *view-name*.

Triggered SQL:

Specifies a list of semi-colon separated SQL statements that are to be executed by the triggering action. This parameter field is mandatory and corresponds to SQL CREATE TRIGGER parameters BEGIN ATOMIC *triggered-SQL-statement*; ... END

Search Conditions:

Specifies the condition that must be met for the triggered action to be executed.

If not specified, the triggered SQL statements will always be actioned.
This parameter field corresponds to SQL CREATE TRIGGER parameter WHEN (*search-condition*).

Referencing Old:

Optionally specifies the correlation names for the transition variables and the table names for the transition tables prior to any changes.

Correlation Name>

Identifies a name by which the triggered action can refer to values in the row before the triggering operation has occurred.
This parameter field corresponds to SQL CREATE TRIGGER parameters REFERENCING OLD AS *correlation-name*.

Table identifier>

Identifies a name by which the triggered action can refer to the table before the triggering operation has occurred.
This parameter field corresponds to SQL CREATE TRIGGER parameters REFERENCING OLD_TABLE AS *table-identifier*.

Referencing New:

Optionally specifies the correlation names for the transition variables and the table names for the transition tables following modifications.

Correlation Name>

Identifies a name by which the triggered action can refer to values in the row after the triggering operation has occurred.
This parameter field corresponds to SQL CREATE TRIGGER parameters REFERENCING NEW AS *correlation-name*.

Table identifier>

Identifies a name by which the triggered action can refer to the table after the triggering operation has occurred.
This parameter field corresponds to SQL CREATE TRIGGER parameters REFERENCING NEW_TABLE AS *table-identifier*.

Create Sequence

The Create Sequence panel (ZZS2CSQ0) is an **interactive panel window**, opened on selection of option 9. in the Create DB2 Object options menu. This panel may be used to create a DB2 sequence in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE SEQUENCE statement in the SELCOPY/i foreground.

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** Executes the SQL CREATE SEQUENCE generated by the specified input field parameters. Hitting <Enter> will perform the same action.
- Command** Opens a text edit view for a temporary data set containing **SQL command** CREATE SEQUENCE syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL CREATE SEQUENCE statement syntax generated for the selected input field values.
- Help** Open the general help for the Create Sequence panel.

Field Entries

Sequence: Fields that together identify the DB2 sequence name to be created at the current server location. The sequence name must not match the name of an existing sequence or one that is generated by DB2.

Schema>
The schema of the new sequence.
The schema name has a maximum length of 128 characters.

Name>
The SQL identifier name of the new sequence.
The sequence name has a maximum length of 128 characters.

These parameter fields are mandatory and correspond to SQL CREATE SEQUENCE parameter *sequence-name*.

Sequence Initialisation:
Options relating to initial sequence values.

Data Type>
Specifies the data type used for the sequence value. Acceptable values are: SMALLINT, INTEGER, BIGINT or DECIMAL (scale zero).

If not specified, the default data type is INTEGER.
This parameter field corresponds to SQL CREATE SEQUENCE parameter *AS data-type*.

Start Value>
Specifies the first positive or negative value in the sequence.

If not specified, the start value is the **MINVALUE** for ascending sequences or the **MAXVALUE** for descending sequences.
This parameter field corresponds to SQL CREATE SEQUENCE parameter *START WITH numeric-constant*.

Increment Value>

Specifies the interval between successive values in the sequence. A positive value or 0 (zero) indicates an ascending sequence, a negative value indicates a descending sequence.

If not specified, the increment value is +1.

This parameter field corresponds to SQL CREATE SEQUENCE parameter INCREMENT BY *numeric-constant*.

Sequencing Options:

General options relating to the sequence.

Specify minimum

Activate this option as required by entering "/" in the selection field.

If not activated, no minimum value limit is set so that the default minimum value for a descending sequence is the minimum value for the data type.

This parameter field corresponds to SQL CREATE SEQUENCE parameter NO MINVALUE.

MIN Value>

Specifies the minimum value at which a descending sequence either cycles or stops generating values, or an ascending value cycles to when the maximum value is encountered.

This parameter field corresponds to SQL CREATE SEQUENCE parameter MINVALUE *numeric-constant*.

Specify maximum

Activate this option as required by entering "/" in the selection field.

If not activated, no explicit maximum value limit is set so that the default maximum value for an ascending sequence is the maximum value for the data type.

This parameter field corresponds to SQL CREATE SEQUENCE parameter NO MAXVALUE.

MAX Value>

Specifies the maximum value at which an ascending sequence either cycles or stops generating values, or a descending value cycles to when the minimum value is encountered.

This parameter field corresponds to SQL CREATE SEQUENCE parameter MAXVALUE *numeric-constant*.

Enable caching

Activate this option as required by entering "/" in the selection field.

If not activated, no pre-allocated values are kept in memory for faster access.

This parameter field corresponds to SQL CREATE SEQUENCE parameter NO CACHE.

Cache Value>

Specifies the maximum number of sequence values to be kept in cache memory.

The minimum value that may be specified is 2.

This parameter field corresponds to SQL CREATE SEQUENCE parameter CACHE *integer-constant*.

Cycle values

Activate this option as required by entering "/" in the selection field.

If activated, ascending sequence values will cycle to the minimum value when the maximum value threshold is encountered, and descending sequence values will cycle to the maximum value when the minimum value threshold is encountered. If not activated, no further sequence values are generated when minimum/maximum threshold limits are encountered.

This parameter field corresponds to SQL CREATE SEQUENCE parameter CYCLE and NO CYCLE.

Sequence in Order of request

Activate this option as required by entering "/" in the selection field.

If activated, sequence numbers will be generated in order of request. If not activated, sequence numbers do not need to be generated in order of request.

This parameter field corresponds to SQL CREATE SEQUENCE parameter ORDER and NO ORDER.

Create Role

The Create Role panel (ZZS2CRO0) is an **interactive panel window**, opened on selection of option 9. in the Create DB2 Object options menu. This panel may be used to create a DB2 role in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the CREATE ROLE statement in the SELCOPY/i foreground.

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** Executes the SQL CREATE ROLE generated by the specified input field parameters. Hitting <Enter> will perform the same action.
- Command** Opens a text edit view for a temporary data set containing **SQL command** CREATE ROLE syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL CREATE ROLE statement syntax generated for the selected input field values.
- Help** Open the general help for the Create Role panel.

Field Entries

- Role:** Identifies the DB2 role name to be created at the current server location. The role name must not match the name of an existing role.
- Name>** The SQL identifier name of the new role.
The role name must not be SYSADM, SYSCTRL or PUBLIC and has a maximum length of 128 characters.
This parameter field is mandatory and corresponds to SQL CREATE ROLE parameter *role-name*.

Drop DB2 Objects

The Drop DB2 Object panel (ZZS2D000) is an **interactive panel window**, opened on selection of option 6. in the DB2 Primary options menu.

This panel 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**.

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 *"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 Drop DB2 Objects option menu panel.

Options

- | | |
|------------------|----------------------------|
| 1. Storage group | 9. Type |
| 2. Database | 10. Function |
| 3. Table space | 11. Stored procedure |
| 4. Table | 12. Trigger |
| 5. View | 13. Global temporary table |
| 6. Alias | 14. Sequence |
| 7. Index | 15. Role |
| 8. Synonym | 16. Trusted context |

Drop Storage Group

The Drop Storage Group panel (ZZS2DSG0) is an **interactive panel window**, opened on selection of option 1. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 storage group in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the **Enter>** field.

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**
Executes the SQL DROP STOGROUP generated by the specified input field parameters.
- Command**
Opens a text edit view for a temporary data set containing **SQL command** DROP STOGROUP syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL DROP STOGROUP statement syntax generated for the selected input field values.
- Help**
Open the general help for the Drop Storage Group panel.

Field Entries

Storage Group Name>
Identifies the name of the DB2 storage group to be dropped from the current server but not one that is in use by a table space or index space. Maximum length of a storage group name is 128 characters.
This parameter field is mandatory and corresponds to SQL DROP parameter STOGROUP *stogroup-name*.

Enter>
Specifies the action on pressing the <Enter> key or, if configured, **double-clicking the left mouse button**.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

- ◇ **Nop**
Indicates that No operation is to be performed.
- ◇ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.
- ◇ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.
- ◇ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Database

The Drop Database panel (ZZS2DDB0) is an **interactive panel window**, opened on selection of option 2. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 database in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the **Enter>** field.

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** Executes the SQL DROP DATABASE generated by the specified input field parameters.
- Command** Opens a text edit view for a temporary data set containing **SQL command** DROP DATABASE syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL DROP DATABASE statement syntax generated for the selected input field values.
- Help** Open the general help for the Drop Database panel.

Field Entries

- Database Name>** Identifies the name of the DB2 database to be dropped from the current server. Dropping a database will also drop all of its table spaces, tables, index spaces and indexes. Maximum length of a database name is 8 characters. This parameter field is mandatory and corresponds to SQL DROP parameter DATABASE *database-name*.
- Enter>** Specifies the action on pressing the <Enter> key or, if configured, **double-clicking the left mouse button**.
- Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:
- ◇ **Nop**
Indicates that No operation is to be performed.
 - ◇ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.
 - ◇ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.
 - ◇ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Table Space

The Drop Table Space panel (ZZS2DTS0) is an **interactive panel window**, opened on selection of option 3. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 table space in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the **Enter>** field.

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** Executes the SQL DROP TABLESPACE generated by the specified input field parameters.
- Command** Opens a text edit view for a temporary data set containing **SQL command** DROP TABLESPACE syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL DROP TABLESPACE statement syntax generated for the selected input field values.
- Help** Open the general help for the Drop Table Space panel.

Field Entries

Table Space: Fields that together identify the DB2 table space name to be dropped from the current server location. Dropping a table space will also drop all of its tables.

Table Space Name>

The table space name of the table space to be dropped.

The table space name must not identify a table space implicitly created for an XML column, and has a maximum length of 8 characters.

Database Name>

The name of the data base in which the table space is defined. If not specified, database name DSNDDB04 is used.

The database name has a maximum length of 8 characters.

This parameter field corresponds to SQL DROP parameter TABLESPACE *database-name.table-space-name*.

Enter>

Specifies the action on pressing the <Enter> key or, if configured, **double-clicking the left mouse button**.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

◇ **Nop**

Indicates that No operation is to be performed.

◇ **RUN**

Execute the generated SQL DROP statement in the SELCOPY/i foreground.

◇ **CMX**

Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.

◇ **JCL**

Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Table

The Drop Table panel (ZZS2DTA0) is an **interactive panel window**, opened on selection of option 4. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 table in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the **Enter>** field.

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	Executes the SQL DROP TABLE generated by the specified input field parameters.
Command	Opens a text edit view for a temporary data set containing SQL command DROP TABLE syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL DROP TABLE statement syntax generated for the selected input field values.
Help	Open the general help for the Drop Table panel.

Field Entries

Table: Fields that together identify the DB2 table name to be dropped from the current server location. 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.

Owner>
The table owner (schema) of the table to be dropped.
The table owner name has a maximum length of 128 characters.

Name>
The SQL identifier name of the table to be dropped.
The table name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter TABLE *table-name*.

Enter> Specifies the action on pressing the <Enter> key or, if configured, **double-clicking the left mouse button**.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

- ◇ **Nop**
Indicates that No operation is to be performed.
- ◇ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.
- ◇ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.
- ◇ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop View

The Drop View panel (ZZS2DVI0) is an **interactive panel window**, opened on selection of option 5. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 view in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the **Enter>** field.

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** Executes the SQL DROP VIEW generated by the specified input field parameters.
- Command** Opens a text edit view for a temporary data set containing **SQL command** DROP VIEW syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL DROP VIEW statement syntax generated for the selected input field values.
- Help** Open the general help for the Drop View panel.

Field Entries

View: Fields that together identify the DB2 view name to be dropped from the current server location. Dropping a view will also drop all synonyms, other views or materialised query tables defined on the view and privileges on the view.

Owner> The view owner (schema) of the view to be dropped.
The view owner name has a maximum length of 128 characters.

Name> The SQL identifier name of the view to be dropped.
The view name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter VIEW *view-name*.

Enter> Specifies the action on pressing the <Enter> key or, if configured, **double-clicking the left mouse button**.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

- ◇ **Nop**
Indicates that No operation is to be performed.
- ◇ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.
- ◇ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.
- ◇ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Alias

The Drop Alias panel (ZZS2DAL0) is an **interactive panel window**, opened on selection of option 6. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 alias in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the **Enter>** field.

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** Executes the SQL DROP ALIAS generated by the specified input field parameters.
- Command** Opens a text edit view for a temporary data set containing **SQL command** DROP ALIAS syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL DROP ALIAS statement syntax generated for the selected input field values.
- Help** Open the general help for the Drop Alias panel.

Field Entries

Alias: Fields that together identify the DB2 alias name to be dropped from the current server location.

Owner> The alias owner (schema) of the alias to be dropped.
The alias owner name has a maximum length of 128 characters.

Name> The SQL identifier name of the alias to be dropped.
The alias name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter ALIAS *alias-name*.

Enter> Specifies the action on pressing the <Enter> key or, if configured, **double-clicking the left mouse button**.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

- ◇ **Nop**
Indicates that No operation is to be performed.
- ◇ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.
- ◇ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.
- ◇ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Index

The Drop Index panel (ZZS2DIN0) is an **interactive panel window**, opened on selection of option 7. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 index in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the **Enter>** field.

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** Executes the SQL DROP INDEX generated by the specified input field parameters.
- Command** Opens a text edit view for a temporary data set containing **SQL command** DROP INDEX syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL DROP INDEX statement syntax generated for the selected input field values.
- Help** Open the general help for the Drop Index panel.

Field Entries

Index: Fields that together identify the name of the user defined DB2 index to be dropped from the current server location. Dropping an index will also drop the index space containing the index.

Owner>
The index owner (schema) of the index to be dropped.
The index owner name has a maximum length of 128 characters.

Name>
The SQL identifier name of the index to be dropped.
The index name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter INDEX *index-name*.

Enter> Specifies the action on pressing the <Enter> key or, if configured, **double-clicking the left mouse button**.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

- ◇ **Nop**
Indicates that No operation is to be performed.
- ◇ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.
- ◇ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.
- ◇ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Synonym

The Drop Synonym panel (ZZS2DSY0) is an **interactive panel window**, opened on selection of option 8. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 synonym in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the **Enter>** field.

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** Executes the SQL DROP SYNONYM generated by the specified input field parameters.
- Command** Opens a text edit view for a temporary data set containing **SQL command** DROP SYNONYM syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL DROP SYNONYM statement syntax generated for the selected input field values.
- Help** Open the general help for the Drop Synonym panel.

Field Entries

- Synonym:** Specifies the DB2 synonym name, for which the user is the owner, to be dropped from the current server location.
- Name>** The SQL identifier name of the synonym to be dropped.
The synonym name has a maximum length of 128 characters.
This parameter field is mandatory and corresponds to SQL DROP parameter SYNONYM *synonym-name*.
- Enter>** Specifies the action on pressing the <Enter> key or, if configured, **double-clicking the left mouse button**.
- Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:
- ◇ **Nop**
Indicates that No operation is to be performed.
 - ◇ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.
 - ◇ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.
 - ◇ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Type

The Drop Type panel (ZZS2DTY0) is an **interactive panel window**, opened on selection of option 9. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 user defined distinct type in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the **Enter>** field.

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
Executes the SQL DROP TYPE generated by the specified input field parameters.

Command
Opens a text edit view for a temporary data set containing **SQL command** DROP TYPE syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL DROP TYPE statement syntax generated for the selected input field values.

Help
Open the general help for the Drop Type panel.

Field Entries

Distinct Type:
Fields that together identify the name of the user defined DB2 distinct type to be dropped from the current server location. The RESTRICT parameter, used to prevent dropping a distinct type if it is used by dependent operations, is default and cannot be overridden by this panel.

Schema>
The schema of the distinct type to be dropped.
The distinct type schema name has a maximum length of 128 characters.

Name>
The SQL identifier name of the distinct type to be dropped.
The distinct type name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter TYPE *distinct-type-name*.

Enter>
Specifies the action on pressing the <Enter> key or, if configured, **double-clicking the left mouse button**.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

- ◇ **Nop**
Indicates that No operation is to be performed.
- ◇ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.
- ◇ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.
- ◇ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Function

The Drop Function panel (ZZS2DFU0) is an **interactive panel window**, opened on selection of option 10. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 user defined function in the current DB2 subsystem by its function name. Dropping a function by its function signature or by its specific name is not supported.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the **Enter>** field.

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** Executes the SQL DROP FUNCTION generated by the specified input field parameters.
- Command** Opens a text edit view for a temporary data set containing **SQL command** DROP FUNCTION syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL DROP FUNCTION statement syntax generated for the selected input field values.
- Help** Open the general help for the Drop Function panel.

Field Entries

Function: Fields that together identify the name of the DB2 function which was created using a CREATE FUNCTION statement and is now to be dropped from the current server location. The RESTRICT parameter, used to prevent dropping a function if it is used by dependent operations, is default and cannot be overridden by this panel.

Schema> The schema of the function to be dropped.
The function schema name has a maximum length of 128 characters.

Name> The SQL identifier name of the function to be dropped.
The function name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter FUNCTION *function-name* RESTRICT.

Enter> Specifies the action on pressing the <Enter> key or, if configured, **double-clicking the left mouse button**.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

- ◇ **Nop**
Indicates that No operation is to be performed.
- ◇ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.
- ◇ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.
- ◇ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Stored Procedure

The Drop Stored Procedure panel (ZZS2DSP0) is an **interactive panel window**, opened on selection of option 11. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 stored procedure in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the **Enter>** field.

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** Executes the SQL DROP PROCEDURE generated by the specified input field parameters.
- Command** Opens a text edit view for a temporary data set containing **SQL command** DROP PROCEDURE syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL DROP PROCEDURE statement syntax generated for the selected input field values.
- Help** Open the general help for the Drop Stored Procedure panel.

Field Entries

Stored Procedure: Fields that together identify the name of the DB2 stored procedure which was created using a CREATE PROCEDURE statement and is now to be dropped from the current server location. The RESTRICT parameter, used to prevent dropping a stored procedure if it is used by dependent operations, is default and cannot be overridden by this panel.

Schema> The schema of the stored procedure to be dropped.
The stored procedure schema name has a maximum length of 128 characters.

Name> The SQL identifier name of the stored procedure to be dropped.
The stored procedure name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter PROCEDURE *procedure-name* RESTRICT.

Enter> Specifies the action on pressing the <Enter> key or, if configured, **double-clicking the left mouse button**.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

- ◇ **Nop**
Indicates that No operation is to be performed.
- ◇ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.
- ◇ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.
- ◇ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Trigger

The Drop Trigger panel (ZZS2DTR0) is an **interactive panel window**, opened on selection of option 12. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 user defined trigger in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the **Enter>** field.

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** Executes the SQL DROP TRIGGER generated by the specified input field parameters.
- Command** Opens a text edit view for a temporary data set containing **SQL command** DROP TRIGGER syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL DROP TRIGGER statement syntax generated for the selected input field values.
- Help** Open the general help for the Drop Trigger panel.

Field Entries

Trigger: Fields that together identify the name of the DB2 trigger to be dropped from the current server location.

Schema>
The schema of the trigger to be dropped.
The trigger schema name has a maximum length of 128 characters.

Name>
The SQL identifier name of the trigger to be dropped.
The trigger name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter TRIGGER *trigger-name*.

Enter> Specifies the action on pressing the <Enter> key or, if configured, **double-clicking the left mouse button**.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

- ◇ **Nop**
Indicates that No operation is to be performed.
- ◇ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.
- ◇ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.
- ◇ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Global Temporary Table

The Drop Global Temporary Table panel is the same panel used for **Drop Table** (ZZS2DTA0) and is opened on selection of option 13. in the Drop DB2 Object options menu.

Drop Sequence

The Drop Sequence panel (ZZS2DSQ0) is an **interactive panel window**, opened on selection of option 14. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 user defined sequence in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the **Enter>** field.

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** Executes the SQL DROP SEQUENCE generated by the specified input field parameters.
- Command** Opens a text edit view for a temporary data set containing **SQL command** DROP SEQUENCE syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL DROP SEQUENCE statement syntax generated for the selected input field values.
- Help** Open the general help for the Drop Sequence panel.

Field Entries

Sequence: Fields that together identify the name of the DB2 sequence to be dropped from the current server location.

Schema> The schema of the sequence to be dropped.
The sequence schema name has a maximum length of 128 characters.

Name> The SQL identifier name of the sequence to be dropped.
The sequence name has a maximum length of 128 characters.

This parameter field corresponds to SQL DROP parameter SEQUENCE *sequence-name*.

Enter> Specifies the action on pressing the <Enter> key or, if configured, **double-clicking the left mouse button**.

Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:

- ◇ **Nop**
Indicates that No operation is to be performed.
- ◇ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.
- ◇ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.
- ◇ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Role

The Drop Role panel (ZZS2DRO0) is an **interactive panel window**, opened on selection of option 15. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 role in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the **Enter>** field.

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** Executes the SQL DROP ROLE generated by the specified input field parameters.
- Command** Opens a text edit view for a temporary data set containing **SQL command** DROP ROLE syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL DROP ROLE statement syntax generated for the selected input field values.
- Help** Open the general help for the Drop Role panel.

Field Entries

- Role:** Specifies the DB2 role name to be dropped from the current server location. The RESTRICT parameter, used to prevent dropping a role if it is used by dependent operations, is default and cannot be overridden by this panel.
- Name>** The SQL identifier name of the role to be dropped.
The role name has a maximum length of 128 characters.
This parameter field is mandatory and corresponds to SQL DROP parameter ROLE *role-name* RESTRICT.
- Enter>** Specifies the action on pressing the <Enter> key or, if configured, **double-clicking the left mouse button**.
- Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:
- ◇ **Nop**
Indicates that No operation is to be performed.
 - ◇ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.
 - ◇ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.
 - ◇ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

Drop Trusted context

The Drop Trusted context panel (ZZS2DTC0) is an **interactive panel window**, opened on selection of option 16. in the Drop DB2 Object options menu. This panel may be used to drop a DB2 trusted context profile in the current DB2 subsystem.

Having configured the input fields, select "Run" from the menu bar or hit <Enter> to execute the action specified in the **Enter>** field.

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** Executes the SQL DROP TRUSTED CONTEXT generated by the specified input field parameters.
- Command** Opens a text edit view for a temporary data set containing **SQL command** DROP TRUSTED CONTEXT syntax generated for the selected input field values. 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 DSNTIAD (via IKJEFT01) to execute the SQL DROP TRUSTED CONTEXT statement syntax generated for the selected input field values.
- Help** Open the general help for the Drop Trusted context panel.

Field Entries

- Trusted context :** Specifies the DB2 trusted context name to be dropped from the current server location. The RESTRICT parameter, used to prevent dropping a trusted context if it is used by dependent operations, is default and cannot be overridden by this panel.
- Name>** The SQL identifier name of the trusted context to be dropped.
The trusted context name has a maximum length of 127 characters.
This parameter field is mandatory and corresponds to SQL DROP parameter TRUSTED CONTEXT *context-name*.
- Enter>** Specifies the action on pressing the <Enter> key or, if configured, **double-clicking the left mouse button**.
Setting the default action for <Enter> is supported to prevent accidental execution of the DROP command. Mutually exclusive options are as follow:
- ◇ **Nop**
Indicates that No operation is to be performed.
 - ◇ **RUN**
Execute the generated SQL DROP statement in the SELCOPY/i foreground.
 - ◇ **CMX**
Open a temporary edit view to display the generated SQL DROP statement. This is the default the first time the panel is opened. Thereafter, the action set by the user persists.
 - ◇ **JCL**
Open a temporary edit view to display JCL that executes the generated SQL DROP statement in batch.

List DB2 Objects

The List DB2 Object 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 in the current DB2 subsystem.

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 amended 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

Name>
Used to specify a filter on storage group name.
A storage group name has a maximum length of 128 characters.

Creator>
Used to specify a filter on storage group creator (owner) authorisation ID.
A storage group creator ID has a maximum length of 128 characters.

VCatName>
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 PF2 by default.
---	--

List Databases

The List Databases panel (ZZS2LDB0) may be used to list databases defined in the current DB2 subsystem.

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 amended 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

Name>
Used to specify a filter on 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.

StGroup>
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 PF2 by default.

List Table spaces

The List Table spaces panel (ZZS2LTS0) may be used to list table spaces defined in the current DB2 subsystem.

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 amended 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

Name>

Used to specify a filter on table space name.
A table space name has a maximum length of 8 characters.

Creator>

Used to specify a filter on table space 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 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 PF2 by default.

List Tables

The List Tables panel (ZZS2LTAB) may be used to list tables defined in the current DB2 subsystem.

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 the prefix command "**T**" against an entry in the List Databases, List Table Spaces or List Tablespace partitions panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be amended 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

Name>

Used to specify a filter on table name.
A table name has a maximum length of 128 characters.

Creator>

Used to specify a filter on table creator (owner) authorisation ID.
A table creator ID 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 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.
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.
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 PF2 by default.

List Views

The List Views panel (ZZS2LVI0) may be used to list views defined in the current DB2 subsystem.

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 amended 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

Owner>

Used to specify a filter on view creator (owner) authorisation ID.
A view creator ID 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 PF2 by default.

List Aliases

The List Aliases panel (ZZS2LALI) may be used to list aliases defined in the current DB2 subsystem.

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 amended 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

Owner>
Used to specify a filter on alias creator (owner) authorisation ID.
An alias creator ID 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 PF2 by default.

List Indexes

The List Indexes panel (ZZS2LINX) may be used to list indexes defined in the current DB2 subsystem.

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 7. Indexes from the List DB2 Object options menu.
- Execute the command **LDINDEX** 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 amended 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.

Panel Input Fields

Index Creator>

Used to specify a filter on index creator (owner) authorisation ID.
An index creator ID 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 creator (owner) authorisation ID for the table on which the index is defined.
A table creator ID 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 E.
B	Browse the contents of the table to which the index entry applies.
CL	List columns belonging to the table to which the index entry applies.
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 PF2 by default.

List Synonyms

The List Synonyms panel (ZZS2LSYN) may be used to list synonyms defined in the current DB2 subsystem.

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 amended 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

Creator>

Used to specify a filter on synonym creator (owner) authorisation ID.
A synonym creator 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.

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 PF2 by default.

List Distinct Types

The List Distinct Types panel (ZZS2LTYP) may be used to list distinct types defined in the current DB2 subsystem.

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 amended 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

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.

Prefix Line Commands

List of Distinct Types does not support any prefix commands, therefore no prefix area is included in the display.

List Triggers

The List Triggers panel (ZZS2LTRG) may be used to list triggers defined in the current DB2 subsystem.

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 amended 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

Owner>

Used to specify a filter on trigger creator (owner) authorisation ID.
A trigger creator ID 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.

Prefix Line Commands

List of Triggers does not support any prefix commands, therefore no prefix area is included in the display.

List Global Temporary Tables

The List Global Temporary Tables panel (ZZS2LGTT) may be used to list tables defined as being temporary in the current DB2 subsystem.

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 amended 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

Owner>

Used to specify a filter on global temporary table creator (owner) authorisation ID.
A table creator ID 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 PF2 by default.

List Sequences

The List Sequences panel (ZZS2LSEQ) may be used to list sequences defined in the current DB2 subsystem.

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

Owner>
Used to specify a filter on sequence creator (owner) authorisation ID.
A sequence creator ID 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

List of Sequences does not support any prefix commands, therefore no prefix area is included in the display.

List Roles

The List Roles panel (ZZS2LROL) may be used to list roles defined in the current DB2 subsystem.

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

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

List of Roles does not support any prefix commands, therefore no prefix area is included in the display.

List Trusted Contexts

The List Trusted Context panel (ZZS2LTRC) may be used to list trusted context defined in the current DB2 subsystem.

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 amended 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

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

List of Trusted Contexts does not support any prefix commands, therefore no prefix area is included in the display.

List Columns

The List Columns panel (ZZS2LCOL) may be used to list columns defined in the current DB2 subsystem.

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 amended 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 Creator>

Used to specify a filter on the creator (owner) of the table or view that contains the column.
A table or view creator ID 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.

Name>

Used to specify a filter on column name.
A column 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 PF2 by default.

List Volumes

The List Volumes panel (ZZS2LVOL) may be used to list volumes defined to storage groups in the current DB2 subsystem.

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 amended 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

Owner>

Used to specify a filter on storage group creator (owner) 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

List of Volumes does not support any prefix commands, therefore no prefix area is included in the display.

List Table space parts

The List Table space parts panel (ZZS2LTS1) may be used to list table space partitions defined in the current DB2 subsystem.

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 amended 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

TSName>

Used to specify a filter on the table space name.
A table space name has a maximum length of 8 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.

VCAT Name>

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.

Stor Name>

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 PF2 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 SELCOPY/i 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.)

```

SELCPY/i - DB2(CBLA): Select a Storage Group
View Refresh Back Forward FDB Text Help          wS wR          Scroll> Csr
Command>
ZZS2SSG0
Select a Storage Group:
Enter Name, Creator and Catalog alias (with optional % wild card) to filter
the list. Position cursor and press <Enter> to select an entry.

Name> %
Creator> %
VCAT Name> %

--NAME--  CREATOR  VCATNAME  VPASSWORD  ---SPACE---  SPCDATE  IBMREQD  CREATEDBY  ---
- CBLATEST  NBJ      CBLDB2A   0           0           N        NBJ      000
- CBLAXXXX  NBJ      CBLA      0           0           N        NBJ      000
- CBLI320G  LAC      CBLI320   0           0           N        LAC      000
- DEMOSG    JGE      CBLDB2A   0           0           N        JGE      000
- DSN8G910  IBMUSER  CBLDB2A   0           0           N        IBMUSER  000
- DSN8G91U  NBJ      CBLDB2A   0           0           N        NBJ      000
- LACTEST1  LAC      LAC        0           0           N        LAC      000
- LACTEST2  LAC      LAC        0           0           N        LAC      000
- SYSDEFLT  IBMUSER  CBLDB2A   0           0           N        IBMUSER  000
- ZZSSG001  LAC      CBLDB2A   0           0           N        LAC      000
Line 1 of 10 | Col 1 of 230 | Views 1 | select *

```

Figure 214. DB2: Select a Storage Group.

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 amended to re-apply the filter and so refresh the display of storage group entries fetched from the DB2 catalog table SYSIBM.SYSSTOGRUP. 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 storage group name.
A storage group name has a maximum length of 8 characters.

Creator>
Used to specify a filter on storage group creator (owner) authorisation ID.
A storage group creator ID has a maximum length of 128 characters.

VCAT Name>
Used to specify a filter on VCATNAME catalog name or alias.
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.)

```

SELCOPY/i - DB2(CBLA): Select a DataBase
View Refresh Back Forward FDB Text Help          wS wR          Scroll> Csr
Command>
ZZS2SDB1
Select a DataBase:
Enter Name and Creator (with optional % wild card) to filter the list.
Position cursor and press <Enter> to select an entry.

  Name> %
  Creator> %

--NAME--  CREATOR-  STGROUP-  -BPOOL--  -DBID-  IBMREQD  CREATEDBY  ROSHARE  -----
- CBLI320D  LAC          CBLI320G  BP16K0    298 K    LAC       0001-01-
- DEMODB    JGE          DEMOSG    BP0       276 K    JGE       0001-01-
- DSNADMDB  IBMUSER     SYSDEFLT  BP0       259 N    IBMUSER   0001-01-
- DSNATPDB  IBMUSER     SYSDEFLT  BP0       256 N    IBMUSER   0001-01-
- DSNDB04   SYSIBM      SYSDEFLT  BP0        4 Y     SYSIBM    0001-01-
- DSNMQDB   IBMUSER     SYSDEFLT  BP0       262 N    IBMUSER   0001-01-
- DSNRGFDB  IBMUSER     SYSDEFLT  BP0       263 N    IBMUSER   0001-01-
- DSNRLST   IBMUSER     SYSDEFLT  BP0       258 N    IBMUSER   0001-01-
- DSNXSR    IBMUSER     SYSDEFLT  BP0       257 K    IBMUSER   0001-01-
- DSN8D91A  IBMUSER     DSN8G910  BP0       264 N    IBMUSER   0001-01-
- DSN8D91E  IBMUSER     DSN8G910  BP0       267 K    IBMUSER   0001-01-
Line 1 of 23 | Col 1 of 262 | Views 3 | select * sort NAME

```

Figure 215. 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 amended 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 <PF2> by default.

Select Tablespace

The Select Tablespace panel (ZZS2STSx) may be used to select a tablespace defined in the current DB2 server (sub-system.)

```

SELCOPY/i - DB2(CBLA): Select a Tablespace
View Refresh Back Forward FDB Text Help          wS wR      Scroll> Csr
Command>
ZZS2ST91
Select a Tablespace:
Enter Name, Creator and/or DBName (with % wild card) to filter the list.
Position cursor and press <Enter> to select an entry.

  Name> %
  Creator> %
  DBName> DEMODB

-NAME-- CREATOR DBNAME -DBID- -OBID- -PSID- -BPOOL-- PARTITIONS LOCKRULE PGSI
-- CARD   JGE    DEMODB   276   12    13  BP0      0  A
-- CUST   JGE    DEMODB   276    1    2  BP0      0  A
-- ORDER  JGE    DEMODB   276    4    5  BP0      0  A
-- PAYMENT JGE    DEMODB   276    9   10  BP0      0  A

Line 1 of 4 | Col 1 of 531 | Views 1 | select *

```

Figure 216. 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 amended 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 (or view) defined in the current DB2 server (sub-system.)


```

SELCOPY/i - DB2(CBLA): Select a Table/View
View Refresh Back Forward FDB Text Help          wS wR
Command>
ZZS2STB1
Select a Table/View:
Update Name/Creator fields (using wild card %) to filter the list.

Name> %
Creator>

-----NAME----- CREATOR-  TYPE  -DBNAME-  -TSNAME-  -DBID-  -OBID-  COLCOUNT -
- ACT                DSN8910  T     DSN8D91A  DSN8S91P  264    32     3
- ADMIN_TASKS        SYSIBM   T     DSNADMDB  DSNADMTS  259    3      2
- BIN_REC_INPUT      SYSIBM   G     DSNDB06   SYSPKAGE   6      0      2
- BIN_REC_OUTPUT     SYSIBM   G     DSNDB06   SYSPKAGE   6      0      2
- BUFFERPOOL_STATUS  SYSIBM   G     DSNDB06   SYSPKAGE   6      0     13
- CATALOG            DSN8910  T     DSN8D91X  DSN8S91X  266    49     3
- CONSUQ             NBJ      T     SAMPLEDB  NBJTS4    277    6      10
- CONSUQD            NBJ      T     SAMPLEDB  NBJTS5    277    16     7
- CUSTOMER           DSN8910  T     DSN8D91X  DSN8S91X  266    21     4
- DATA_SHARING_GROUP SYSIBM   G     DSNDB06   SYSPKAGE   6      0     10
- DB2_CMD_OUTPUT     SYSIBM   G     DSNDB06   SYSPKAGE   6      0      2
- DB2_SYSPARM        SYSIBM   G     DSNDB06   SYSPKAGE   6      0      8
Line 1 of 369 | Col 1 of 645 | Views 1 | select *

```

Figure 217. DB2: Select a Table/View.

Variations of this panel exist as follow:

Select Table/View (ZZS2STB0)

Used by create DB2 table panel views 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 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/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.

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 amended 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

Name>

Used to specify a filter on table name.
A table name has a maximum length of 128 characters.

Creator>

Used to specify a filter on table creator (owner) authorisation ID.
A table creator ID 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 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.)

```

SELCOPY/i - DB2(CBLA): Select a Distinct Type
View Refresh Back Forward FDB Text Help          wS wR
Command>
ZZS2SDT0
Schema> NBJ
Name> %
Source Type> %

SCHEMA OWNER ---NAME---- CREATEDBY SOURCESCHEMA SOURCETYPE METATYPE DATATYPEPI
-- NBJ NBJ FLOAT21 NBJ SYSIBM REAL T -21474836
-- NBJ NBJ FLOAT53 NBJ SYSIBM DOUBLE T -21474836
-- NBJ NBJ NBJCHAR1 NBJ SYSIBM CHAR T -21474836
-- NBJ NBJ NBJCHAR2 NBJ SYSIBM CHAR T -21474836
-- NBJ NBJ NBJDEC84 NBJ SYSIBM DECIMAL T -21474836
-- NBJ NBJ NBJTIMESTAMP NBJ SYSIBM TIMESTAMP T -21474836
-- NBJ NBJ NBJTSTMP NBJ SYSIBM TIMESTAMP T -21474836
-- NBJ NBJ NBJVARB NBJ SYSIBM VARBINARY T -21474836
-- NBJ NBJ NBJVARG NBJ SYSIBM VARGRAPHIC T -21474836
-- NBJ NBJ NUM50 NBJ SYSIBM DECIMAL T -21474836
-- NBJ NBJ USERID NBJ SYSIBM CHAR T -21474836
-- NBJ NBJ USERIDX NBJ SYSIBM CHAR T -21474836
-- NBJ NBJ USERU NBJ SYSIBM CHAR T -21474836
Line 1 of 14 | Col 1 of 201 | Views 1 | select *
  
```

Figure 218. 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 amended 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 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 referential constraint (parent/foreign key) relationship which requires that the parent key columns constitute a unique key on the parent table.

```

SELPCOPY/i - DB2(CBLA): Select a Primary or Unique Key Constraint
View Refresh Back Forward FDB Text Help          wS wR
Command>
ZZS2SCN0
Select a Primary or Unique Key Constraint:
Update Constraint/Table fields below (using wild card %) to filter the list.

Table Name>
Table Creator> ZZS
Constraint> %

CONSTNAME  TBCREATOR  --TBNAME--  CREATOR  TYPE  IXOWNER  -IXNAME--  -----CREATE
-- ALIAS    ZZS        ZSSALIASMOD NBJ      P     NBJ      ZSSXALIM   2011-03-17-10.
-- APAR     ZZS        ZSSAPAR     LAC      P     LAC      ZSSXAPAR   2010-12-22-12.
-- APAR     ZZS        ZSSAPARIQ   LAC      P     LAC      ZSSXAPIQ   2010-12-22-12.
-- FMID     ZZS        ZSSFMID     LAC      P     LAC      ZSSXFMID   2010-12-16-14.
-- FMID     ZZS        ZSSCSECT    LAC      P     LAC      ZSSXCSEC   2010-12-22-15.
-- FMID     ZZS        ZSSXMOD     LAC      P     LAC      ZSSXMOD    2010-12-20-10.
-- FMID     ZZS        ZSSXINCLUDE LAC      P     LAC      ZSSXINC    2010-12-20-15.
-- FMID     ZZS        ZSSXALIAS   LAC      P     LAC      ZSSXALIA   2010-12-20-10.
-- FMID     ZZS        ZSSPREL     LAC      P     LAC      ZSSXPREL   2011-11-10-12.
-- FMID     ZZS        ZSSLINK     LAC      P     LAC      ZSSXLINK   2011-12-20-10.
-- IQ      ZZS        ZSSIQMOD    LAC      P     LAC      ZSSXIQMO   2011-02-03-15.
Line 1 of 16 | Col 1 of 119 | Views 1 | select *

```

Figure 219. 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 amended 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.

```

SELPCOPY/i - DB2(CBLA): Display Primary/Unique Key Constraint Columns
View Refresh Back Forward FDB Text Help          wS wR
Command>
ZZS2LCC0
COLNAME  COLSEQ  COLTYPE-  LENGTH  SCALE-  TYPESCHEMA  TYPENAME  SOURCETYPE  CONSTNAM
-- ALIAS  1  CHAR    8        0  SYSIBM     CHAR      ALIAS      ALIAS
-- SMOD   2  CHAR    8        0  SYSIBM     CHAR      ALIAS      ALIAS
-- TYPE   3  CHAR    8        0  SYSIBM     CHAR      ALIAS      ALIAS
Line 1 of 3 | Col 1 of 80 | Views 1 | select *

```

Figure 220. 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 Foreign Key Column

The Select Foreign Key Column panel (ZZS2SC00) may be used to select a foreign key column from the list of columns belonging to the current DB2 table definition.

This panel is used for selection of a column which matches the relevant parent key column and so may be paired in a referential constraint (parent/foreign key) relationship.

```

SELCOPY/i - DB2(CBLA): Referential Constraint Foreign Key Column Selection
View Refresh Back Forward FDB Text Help          wS wR
Command>
ZZS2SC00
Select Foreign Key Column:          Table Owner CBL          + Name NBJ2TAB          +
Enter Column name field with wild card character % to filter the list.

Column Name> %

ColName  DataType  Length  Scale  DistSchema  DistName
--
IQ       CHAR      8       0
SMOD    CHAR      8       0
TYPE    CHAR      8       0

Line 1 of 3 | Col 1 of 51 | Views 1 | select *

```

Figure 221. DB2: Select a Foreign Key Column.

The contents of the foreign key 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 amended to re-apply the filter and so refresh the display of foreign key column entries fetched from the SELCOPY/i in-storage table.

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 to be assigned to the foreign key.
>	Open a new window containing a zoomed vertical display of the entry's fields. Assigned to <PF2> by default.

Audit Trail Functions

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 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 SDE edit profile macro to be executed when the table is edited.

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 SELCOPY/i 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 SELCOPY/i 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 SELCOPY/i 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 SELCOPY/i 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.

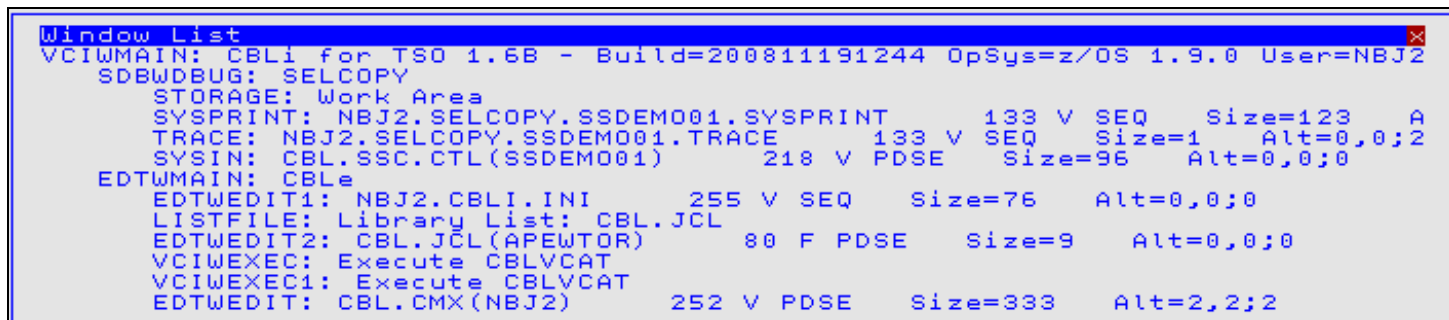
Window List (=W)

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 **CBL 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.



```
Window List
VCIWMAIN: CBLi for TSO 1.6B - Build=200811191244 OpSys=z/OS 1.9.0 User=NBJ2
SDBWDEBUG: SELCOPY
STORAGE: Work Area
SYSPRINT: NBJ2.SELCOPY.SSDEMO01.SYSPRINT      133 V SEQ      Size=123   A
TRACE: NBJ2.SELCOPY.SSDEMO01.TRACE           133 V SEQ      Size=1     Alt=0,0;2
SYSIN: CBL.SSC.CTL(SSDEMO01)                 218 V PDSE      Size=96    Alt=0,0;0
EDTWMAIN: CBL
EDTWEDIT1: NBJ2.CBLI.INI                     255 V SEQ      Size=76    Alt=0,0;0
LISTFILE: Library List: CBL.JCL
EDTWEDIT2: CBL.JCL(APEWTOR)                  80 F PDSE      Size=9     Alt=0,0;0
VCIWEXEC: Execute CBLVCAT
VCIWEXEC1: Execute CBLVCAT
EDTWEDIT: CBL.CMX(NBJ2)                     252 V PDSE     Size=333   Alt=2,2;2
```

Figure 222. Window List window.

SELCOPY/i Command Reference

SELCOPY/i commands may be issued from a command line at the Command> prompt.
All SELCOPY/i main window **menu bar** commands have a command line equivalent.

Command	Description
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 SELCOPY/i Module List.
AUDPRINT	Display formatted DB2 Audit log output.
BACKWARD	Scroll the display backwards by one page.
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 SELCOPY/i command processor.
CBLICANCEL	Exit the SELCOPY/i session without opening the quit session confirmation pop-up message.
CBLNAME	Open a storage window containing the loaded CBLNAME module.
CFOUT	Display SDE formatted Compare File report output.
CLOSE	Close a window.
COMMANDLINE	Set the command line attributes.
COMPFILE	Execute the Compare File Utility.
COMPLIB	Execute the Compare Library Utility.
CURSORSELECT	Perform the default operation based on the cursor position.
DB2	Open the SELCOPY/i 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.
FORWARD	Scroll the display forwards by one page.
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.
HELP	Open the help top level 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.
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 may be passed as a parameter. For VSE a catalog DLBL is required.
LD	Open the dataset details list window. For MVS a partial dataset name may be passed as a parameter. For VSE not yet implemented. For CMS a file name, type and mode 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. Not implemented for VSE.

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.
LLX	Execute the Locat 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. Not implemented for VSE.
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 string may be passed as a parameter.
LX	Open the VTOC extents list window. A volume name may be passed as a parameter. Not yet implemented for VSE.
MAXIMISE	Maximise a window.
MDINEXT	Place focus on the next MDI child window.
MDIPREV	Place focus on the previous MDI child window.
MINIMISE	Minimise a window.
MOVEWINDOW	Move a window.
NEXTMAINWINDOW	Sets the focus window to the next main window within the SELCOPY/i desktop.
NEXTWINDOW	Sets the focus window to the next window in the ring of all windows.
POWER	Open the POWER Command Output window.
PREVMAINWINDOW	Sets the focus window to the previous main window within the SELCOPY/i desktop.
PREVWINDOW	Sets the focus window to the previous window in the ring of all windows.
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 appearance 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.
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. Not yet implemented for VSE.
SYSLL	Open the Link List window. (MVS only)
SYSLPA	Open the LPA Modules window. (MVS only)
SYSTEMENU	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 SELCOPY/i.
TOP	Display the first lines of data.
UP	Scroll the window display upwards.
VCAT	Open the Execute CBLVCAT window. A CBLVCAT command string 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.

Parameters:

AMS_Command
AMS/IDCAMS command to be executed.

Examples:

AMS LISTCAT ENTRY (CBL.MCAT.CMP.P) ALL

AMSDIALOG

Syntax:

```
>>--- AMSDialog ---+--- Ksds -----+
                    |-----+
                    | Esds  -----+
                    | Rrds  -----+
                    | Lds   -----+
                    | ALias -----+
                    | GDG   -----+
```

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.

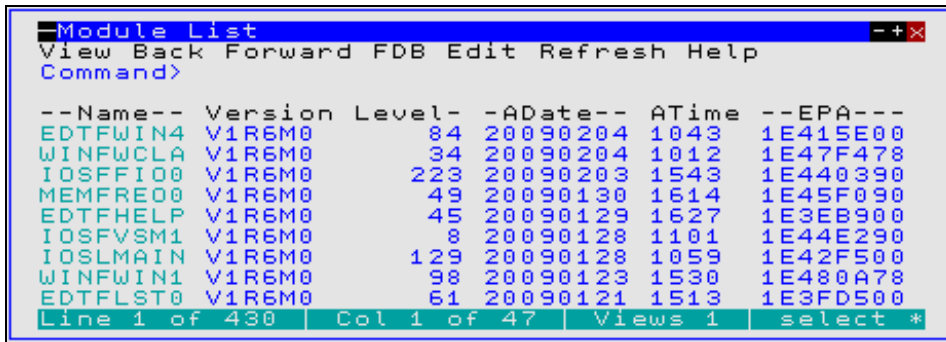


Figure 223. Module List window.

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 CBL 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.

BACKWARD

Syntax:

```
>>--+ Back -----+----->>
      |                   |
      +- BACKWARD -+
```

Description:

Use the BACKWARD command to scroll the window contents backwards by a page.

Note that BACKWARD is the same as UP CURSOR where the cursor is outside the display area.

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 **CBL**e 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 CBL_e text editor main window has been stopped, BROWSE will start the CBL_e 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 the DSN of a sequential or VSAM data set, the member name (with or without the library DSN) of a PDS/PDSE member or an HFS file name. If member name is specified without a library DSN, the DSN of the library member in the current edit view is used.

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.ct1(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.

COMMANDLINE

Syntax:

```

>>--+ COMMANDLINE  -+--+-----+--+-----+--+-----+-----+----->
      |              | |              | |              | |              |
      +- CLN -----+ +- windowname -+ +- SHOW  -+ +- TOP  -+
                                         | |              | |              |
                                         +- HIDE  -+ +- BOT  -+
<-----+-----+-----+-----+-----+-----+-----+-----+-----><
      |              | |              | |              | |              |
      +- PROMPT=prompt -+ +- SCROLL  ----+
                                         | |              | |              |
                                         +- NO SCROLL --+
    
```

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.

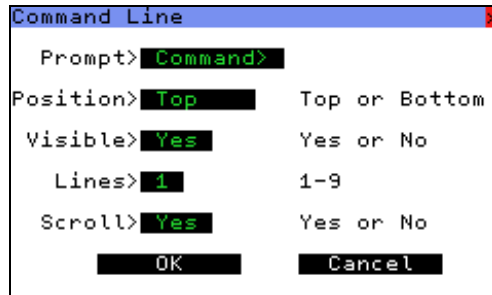


Figure 224. Command Line Dialog Window.

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 CBL 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.

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 *new_structname*

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 SDO|COBOL|PL1|ADATA *old_structname*

Specifies the structure format and structure name used to map record data in *old_fileid* for a formatted compare. See USING *new_structname*.

Records that are assigned a record type defined in *old_structname* will be compared with records assigned the same record type in *new_structname*. Any records in the NEW and OLD files that are assigned a record type that does not exist in both *old_structname* and *new_structname* will be flagged as either inserted or deleted as appropriate.

If a field column in a record type defined in *new_structname* does not exist in the record type of the same name defined in *old_structname* 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_structname* will be used to map record data in *old_fileid*.

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*.

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').

VIEW *rectype, ...*

For **formatted compare** only, VIEW specifies one or more record types (*rectype*) that are defined by the *new_structname* 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_structname*.

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_structname* 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_structname* but not in the *rectype* for *old_structname* then that *fieldname* is not included in the compare. If *fieldname* does not exist in *rectype* for *new_structname* 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_structname* and *new_structname*.

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 mismatching record in the OLD file.

If a read-ahead match is found, the *ra_match* value is then used to identify 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 synchronisation (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 initial 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 *rectype*.

If a different structure is applied to OLD file data (i.e. USING *old_structname*), then each key *fieldname* must exist in record type *rectype* for both *new_structname* and *old_structname*.

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 types Compare or Compare *record_type* 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.

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.

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.

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.

REPORT|RPT *fileid*

REPORT specifies that the COMPFILE 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 (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 eligible 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".

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:

```
>>-- COMPLIB -----+-----+----->>
                    |         |         |
                    +- | CLI Options | -+
```

CLI Options:

```
>-- new_libdsn -----+-----+----- old_libdsn ----->
                    |         |         |
                    |         v         |
                    +- ( --- mbr_mask -+- ) -+

>-----+-----+-----+----->
        |         |         |         |
        +- STRIP -+-----+-----+ BATCH -+
                    |         |         |
                    +- 'char' -+
```

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](#).

COMPLIB executes the SELCOPY program to compare selected library members and generates an output report suitable for edit using the SELCOPY/i (CBL) text editor.

Parameters:

new_libdsn

Specifies the full DSN of the PDS/PDSE library containing the NEW images of library member data to be compared.

mbr_mask

Specifies a member name mask which is to be used by COMPLIB to select members from the **NEW library only** (*new_libdsn*) to be included in the compare operation. Members of the OLD library (*old_libdsn*), whose member names match a specified *mbr_mask*, are **not** selected for compare unless a member of the same name exists in the NEW library.

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 one or more blanks and/or a "," (comma).

Default is to compare **all** members of both the NEW and OLD libraries.

old_libdsn

Specifies the full DSN of the PDS/PDSE library containing the OLD images of library member data to be compared.

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').

BATCH

BATCH specifies that COMPLIB should generate JCL to execute SELCOPY in batch. Concatenated SYSIN input includes the STRIP option status and strip character, the input control statements (ZZSCOMPL) and, if *mbr_mask* was specified, a list of selected members. By default, SYSPRINT is written to SYSOUT=*

The JCL will be displayed in a temporary CBL) text edit window view.

If BATCH is not specified, COMPLIB will execute SELCOPY in the foreground to generate the compare libraries utility output.

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.

CURSORSELECT**Syntax:**

```
>>--- CURSORSELECT -----><
|
| CSELECT -----+
|
| CSEL -----+
|
```

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:**

```
>>-- DB2 -----><
|
| v
| +-----+ | fastpath --+
| | v | |
| +-----+ |
| fieldname=value +---+
|
```

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:

```
>>-- DCMD +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
          |         |         |         |         |         |         |         |         |         |
          +- -SSN ssn_name -+-+ +- -LIMIT n_bytes -+ +- db2_command -+-+
```

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:

```
>>-- DOWN -----><
      |-----| |-----|
      +- windowname -+ +- CURSOR ---+
      |               | +- DATA ----+
      |               | +- HALF ----+
      |               | +- MAX -----+
      |               | +- PAGE ----+
      |               | +- n_lines --+
      |-----| |-----|
```

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 default 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.

DSINFORMATION

Syntax:

```
>>-- DSInformation -----+-----+-----+----->>
                          |         |
                          +-- fileid --+
```

Description:

Display the appropriate **Data Sets Information** panel for the specified *fileid*. *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.

The Data set information panel may also be opened by entering the "I" prefix command against an entry in a file list.

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.

Examples:

```
DSINFO OEM.SELC320.CBL13295.CSI
Display information for the SELCOPY Product Suite SMP/E CSI data set.
```

DSQL

Syntax:

```
>>-- DSQL -+-----+-----+-----+----->>
          |         |         |         |
          +- -SSN ssn_name ---+ +- -LIMIT n_rows -+ +- sql_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.
```

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.

EDIT

Syntax:

```
>>-- Edit ----->>
      |           | |           |
      +-- fileid --+ +- ( --- PROFILE macroname ---+-----+
                               | |
                               +-- NOPROFile -----+ +-| HFS Opts |--+
```

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.

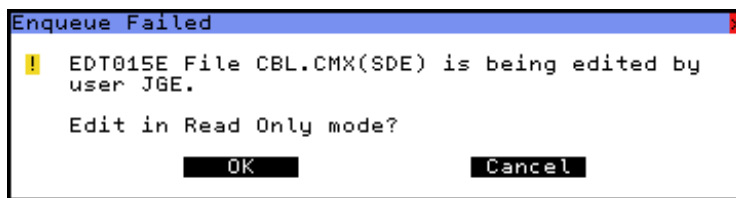


Figure 225. Enqueue Failed Window.

Parameters:

fileid

The fileid of the file to be viewed.

For MVS, *fileid* may be the DSN of a sequential or VSAM data set, the member name (with or without the library DSN) of a PDS/PDSE member or an HFS file name. If member name is specified without a library DSN, the DSN of the library member in the current edit view is used.

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:

```
>>--- EO --- jnm -- jno -- class ---+-----+-----+><
                                     |         |         |
                                     +--- userid ---+-----+
                                     |         |         |
                                     +--- passwd ---+
```

Description:

Use the EO command to Edit (read only) an Output listing from the system's output queues. A new CBLed text editor window is opened if SELCOPY/i INI variable Edit.Instance=Multiple or if no CBLed 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 CBLed command, **SUBMIT**. A job may be submitted to batch from within CBLed 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:

<i>jnm</i>	The required Job Name.
<i>jno</i>	The required Job Number. Note that, when a job is submitted using the CBLed SUBMIT command, the job number is displayed in the job submitted confirmation message.
<i>class</i>	The required List Class.
<i>userid</i>	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.
<i>passwd</i>	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, HFS file or PDS(E) member:

```
>>-- ERAsE -----+-----+----- fileid -----><
                   |               |
                   +- valid ----- : -----+
```

Erase a CMS file on an accessed minidisk:

```
>>-- ERAsE ----- fileid -----><
```

Erase a VSE sequential or VSAM file:

```
>>-- ERAsE -----+ valid ---+ : ----- fileid -----><
                   |         |
                   +- catdsn ---+
```

Description:

Erase (delete) 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.

For VSE, sequential files may only be erased if the CBL software product **CBLVCAT** is licensed. SELCOPY/i uses CBLVCAT's DEL operation to perform the erase.

Parameters:

valid
For MVS uncataloged data sets and VSE sequential disk files, this is the volume serial number of the DASD volume on which the file resides.

catdsn
For VSE VSAM files, this is the full fileid of the VSAM catalog to which the VSAM managed file belongs.

fileid
The full fileid of the file to be erased.

For MVS, 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.

Examples:

```
erase test.exec.a
Erase CMS file TEST.EXEC.A.

erase cbl.cbli.test.file
Erase MVS cataloged data set CBL.CBLI.TEST.FILE.

erase cbl.cbli.testlib(example1)
Erase MVS PDS(E) member CBL.CBLI.TESTLIB(EXAMPLE1).

erase OEM001:cbl.cbli.test.file
Erase MVS uncataloged data set CBL.CBLI.TEST.FILE from DASD volume OEM001.

erase SYSWK1:CBL.SELCOPY.NAM
Erase VSE sequential file CBL.SELCOPY.NAM on SYSWK1. (CBLVCAT must be licensed.)

erase VSESP.USER.CATALOG:CBL.TEST.KSDS
Erase VSE VSAM managed data set CBL.TEST.KSDS cataloged in the VSAM catalog, VSESP.USER.CATALOG.
```

EXECSQL

Syntax:

```
>>- EXECSQL ----->
      |         |         |         |
      +- SSN ssn_name -+ +- LIMIT n_rows -+
> +-----+-----+ | Output DSN Alloc Options | ----->
      |         |         |
      +- OUTDSN outfile -+
      +- NOFORMat -- | Unformatted Options | ---+
> +-----+-----+-----+-----+-----+-----+-----><
      |         |         |         |         |         |
      +- FORMat -----+-----+ +- INDSN infile ---+
      |         |         |         |         |         |
      +- OUTSDO struct_name -----+ +- sql_syntax -----+
```

Output DSN Alloc Options:

```
+-----+-----+-----+-----+-----+-----+-----+
| SPACE ( CYLINDERS 1 1 ) -----+
|-----+-----+-----+-----+-----+-----+-----+
|         |         |         |         |         |         |
|         +--1 ----- 1 -----+
+- SPACE ( +- TRacks -----+-----+ ) -+
      |         |         |         |         |         |
      +- TRks -----+ +- 1 -----+
      |         |         |         |         |         |
      +- CYlinders -+ +- primary -----+
      |         |         |         |         |         |
      +- secondary -+
+- RECFM VB -----+ +- LRECL 4092 ---+ +- BLKSIZE 0 -----+
|-----+-----+-----+-----+-----+-----+-----+
+- RECFM +- FB ---+ +- LRECL reclen -+ +- BLKsize blksize -+
      +- FBA -+
      +- VB ---+
      +- VBA -+
+- UNIT SYSDA ----+
|-----+-----+-----+-----+-----+-----+
+- UNIT unitname -+
```

Unformatted Options:

```
+-----+-----+-----+-----+-----+-----+-----+
| MAXCHAR 250 -----+ +- MAXNUM 33 -----+
|-----+-----+-----+-----+-----+-----+-----+
|         |         |         |         |         |         |
+- MAXCHAR maxcharwidth -+ +- MAXNUM maxnumberwidth -+
```

Description:

The EXECSQL primary command is the command line interface to the SELCOPYi SQL Execution facility.

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.
- 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. SELCOPY/i will perform a CONNECT to this sub-system before executing the SQL syntax.

Default is the SSN value assigned to the SELCOPY/i 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.

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.Dyyyyddd.Thhmmssu.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 SDO structure data set.

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.Dyyyyddd.Thhmmssu.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. Comment data must be enclosed with '/'* and '*/' characters and may span several consecutive lines.

The input file may be an existing sequential data set, a PDS or PDSE library member.

If neither INDSN nor *sql_syntax* is specified, then the focus text edit or structured edit/browse dataset is used for INDSN.

sql_syntax

One or more SQL statements (*sql_syntax*) may be specified in-line following the last EXECSQL keyword parameter.

sql_syntax and INDSN *infile* are mutually exclusive parameters.

Examples:

```
execsql outdsn /u/cbl/nbj/1st/execsql_1st
```

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.

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 and *to_fileid* is a not, then records from all selected members will be copied to the single target data set.

If *from_fileid* contains a library member mask and *to_fileid* is a library (i.e. library copy), then specification of a target library member name is invalid and FCOPY will fail. Note that FCOPY does not support specification of a target library member mask to perform a rename of multiple source members.

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 output structure *in_struct* is specified for *from_fileid*.

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 *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) <<<<<<*"

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 unallocated, then it should be allocated as new using the same DCB geometry and SPACE attributes as the *from_fileid* library.

NEW is ignored for file copy.

If this option is not selected, then the library will not be created and ZZSD353E open error message is returned.

Note that NEW is not a valid option if FCOPY is executed in batch.

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.
```

FORWARD

Syntax:

```
>>--+ Forw -----+-----><
      |             |
      +- FORWARD ---+
```

Description:

Use the FORWARD command to scroll the window contents forwards by a page.

Note that FORWARD is the same as DOWN CURSOR where the cursor is outside the display area.

FS

Syntax:

```
>>--+ FS -----+-----+-----><
      |             | |             |
      +- FILESEARCH -+ +-- filemask -- string ---+
```

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:

>>- FSU +-----+
|
|+-+| Unformatted File Search Opts |---+----| Common 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 |---+

Unformatted File Search Opts:

>---| File Input |-----| Search Opts |----->

Unformatted File Update Opts:

>---| File Input |-----+-----+ Change Opts |----->
|
|+-| Search Opts |---+
+-- NOUpdate --+
|
|-----+----->
|
|+- UPDATE ----+

Unformatted File Copy Opts:

>---| File Input |-----+-----+ Change Opts |---+>
|
|+-| Search Opts |---+
+ MEMBERDLM -- 0 +-
|
|-----+-----+
|
|+- APPend -+ +- STRIP -+- char + + MEMBERDLM +- 0 -+
|
|+- PAD ----+ +- 1 -+
|
|+- FILL ---+ +- 2 -+

Formatted Library Copy Opts:

```

>--| Library Input |--| SDE Opts |-----+----->
      |
      +-----+-----| Change Opts |-->
      |
      +-| Search Opts |-->

>--- OUTPUT lib_dsn -----+----->
      |
      +- NEW -+ +- REPlace -+ +- STRIP --+ char -+
      |
      +- PAD ----+
      +- FILL ---+
  
```

Formatted Library Remap Opts:

```

>--| Library Input |--| SDE Opts |-----+----->
      |
      +-----+-----| Change Opts |-->
      |
      +-| Search Opts |-->

      +- SDO ----+
>-- OUTPUT lib_dsn -- USING -----+----->
      |
      +- COBOL -+
      +- PLI ---+
      +- ADAta -+

>-----+----->
      |
      +- NEW -+ +- REPlace -+
  
```

File Input:

```

      +-----+
      | v      |
>-- INPUT ( -- fileid_mask +- ) -----+----->
      |
      +-| HFS Opts |-->
  
```

HFS Opts:

```

      +- STD -----+
      |
      +- EOL -----+-----+----->
      |
      +- CR -----+
      +- LF -----+
      +- NL -----+
      +- CRLF -----+
      +- LF CR -----+
      +- CR NL -----+
      +- string ---+
      |
      +- LRECL lrecl -+
      |
>-----+----->
      +- RECFM +- F -----+-----+----->
      |
      | +- (0,2,0) -----+-----+----->
      | |
      +- V +- (off,len,origin) -+--+
      |
>-----+----->
      |
      +- RECURSE -+ +- CASEIgn -+
  
```


	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 and library members 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 an accompanying member mask, or 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. A pre-allocated DDNAME (non-HFS) which may represent one or more (concatenated) data sets and/or libraries. (e.g. SYSEXEC)
2. 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.

Because 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 be enclosed in single quotes (apostrophes) or double quotes. e.g.

```
'/u/ibmuser/tmp/*' /* Search all files in this directory. */
```

3. 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 "," (comma) 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(*)
```

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 distinguishable 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.
<i>string</i>	-	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:

<i>off</i>	Offset of the record length field from the start of the record.
<i>len</i>	Length of the record length field.
<i>origin</i>	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 delimited 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 discriminate 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.

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.DATA3. ) USING( DEV.TEST.SDO(SDDATA3) )
  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, *find_parms* parameters ALL/NEXT/FIRST/PREV/LAST all have the same 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*, logical AND or logical OR. Note that a combination of both AND and OR logical operators is invalid. This indicates whether a record must satisfy all *find_parms* operations or only one of the *find_parms* operations in order to be selected.
e.g.

```
FSU INPUT( DEV.USER01.JCL(*) )
  FIND( (c'EXEC' WORD 1 20) AND (c'IKJEFT01' NEXT) AND ('REGION' NEXT) )

FSU INPUT( DEV.USER*.COBOL.COPYBOOK(*) )
  FIND( (c'REDEFINES' WORD) OR ('OCCURS' WORD) OR ('filler.' 12 80) )

FSU INPUT( DEV.TEST.DATA3. )
  USING( DEV.TEST.SDO(SDDATA3) )
  FIND( (25 (#10:#18)) OR ('Ramsay' (EMP_NAME)) )
```

Where logical AND is used, each execution of a FIND operation will perform a scan for the search string within the **entire width** of the record data and is not subject to the position of the data found by a previous, successful FIND *find_parms* operation.

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 or NEXT change the first occurrence and LAST or PREV change 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*, logical AND or logical OR. Note that a combination of both AND and OR logical operators is invalid. 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.
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, each execution of a CHANGE operation will perform a scan for the search string within the **entire width** of selected record data and is not subject to the position of the data found (or changed) by a previous, successful FIND *find_parms* or CHANGE *change_parms* operation.

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.

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.

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.

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 allocated as new 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 library will not be created and ZZSD353E open error message is returned.

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.

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 (not 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 eligible 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.

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.

Parameters:

FSUEND has no parameters.

FSUOUT

Syntax:

```
>>-- FSUOUT-----+-----+-----<<
                |               |
                +-- fsu_report_fileid --+
```


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.Thmmss*) is displayed.

FSUOUT may also be 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:

```

>>-- FSUUNDO  (1)
              +- Panel  -----+ +- Verify  --+
              |-----+-----+-----+-----+
              +- fsu_report_fileid -+ +- Foreground -+ +- Update  --+
              |-----+-----+-----+-----+
              +- Background  --+
              |-----+-----+-----+-----+
              +- Terse  -----+
              |-----+-----+-----+-----+
              +- Extended -+ +- Diagnose  ---+

```

Notes:

- 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 CBL 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
```

HELP

Syntax:

```
>>-- Help +-----+----->>
          |         |
          +-- topic ---+
```

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.

Examples:

HELP

Open the Help window contents page.

HELP CBL*e*

Open the Help window at the CBL*e* command page.

H "OEM.CBL.HTML (TEST)

Open a specific HTML document library member.

H "ZZSISIZE"

Open the SELCOPY/i Help member name ZZSISIZE.

HOME

Syntax:

```
>>-- Home -----<<
```

Description:

Edit the user's personal command centre (CMX) file. A new CBL*e* 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.

Parameters:

IEBCOPYDIALOG has no parameters.

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 **CBL*e* 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.

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 **KEYLISTs**, 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)

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)

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:

```
>>-- KEYLIST --+-----+>>
|
|  ON  -----+
|
|  OFF -----+
|
|  PRoDuct -----+
|
|  SHARed -----+
|
|  PRIVate -----+
|
|  SET  -----+
|
|                    |
|                    +--- key-list ---+
|
```

Description:

Use the KEYLIST command with no parameters to display a **selection-list** of all loaded Function Key list objects. From any KeyList maybe selected, either for view/edit or for use by the window from which the KEYLIST command was entered.

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**.

- 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.

- 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
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 **KEYLISTs**, 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

By default, the associated object name for a selected catalog entry is displayed in the first column. For example, if the selected entry is a VSAM CLUSTER object, a separate list entry may exist for its associated DATA, INDEX and ALTERNATE INDEX objects. Similarly, if the selected entry is a DATA object, a list entry will exist for the associated CLUSTER object but not an INDEX object which is associated with the CLUSTER, not the DATA object.

The Associations 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:

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 unnecessary 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.

For MVS and VSE, the LC (List Catalog entries) command may be 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 VSE, the LC command is only supported where 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 than 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 unnecessary 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.

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 amended *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 unnecessary 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.

See Also:

LC Command

Examples:

```

LD CBL.%%C
LD CBL.SYS*.**  USERCAT.CBLCAT  A
    
```

LEFT

Syntax:

```
>>-- LEFT -----><
      |           |           |
      +- windowname -+   +- CURSOR ---+
                        +- DATA ----+
                        +- HALF ----+
                        +- MAX ----+
                        +- PAGE ----+
                        +- n_cols ---+
```

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 default 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:

```
>>--+ LJQ -----+--+-----+-----><
      |           | |           |
      +- LISTJOBENQ -+  +-- jobname --+
```

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:

```
>>--+ LL -----+--+-----+-----><
      |           | |           |
      +- LISTLIBRARY -+  +-- library --+
      |           |
      +- LM -----+
      |           |
      +- LISTMEMBERS -+
```

Description:

Use the LL (List Library) command to open a **Library List** window and optionally list the members of an MVS PDS/PDSE, VSE LIBR library or CMS minidisk.

The Library List window may also be opened via the List menu of the SELCOPY/i main window menu bar.

Note: Not implemented for CMS.

Parameters:

library The name of the library for which the contents are to be listed.

◇ For **MVS** the library parameter is a PDS (or PDSE) dataset name and optionally one or more member name mask.

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.

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 CBL 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 CBL text editor macro path will be searched for the specified *mbr_mask* member mask(s).

This list of libraries may be displayed using the CBL 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 -----+-----+-----+-----+-----+-----+-----+-----+-----+----->>
      |           |           |           |           |           |           |           |
      +- LISTPATH -----+ +- -C ---+ +- -S ---+ +- hfs_path ---+
      |           |           |           |           |           |           |
      +- LISTP -----+
      |           |           |           |           |           |           |
      +- LPATH -----+
```

Description:

The LP (List Path entries) command may be used to open an **HFS Path List** window to list information about entries that match the specified HFS path.

If no parameters are specified, the list window will be opened with fields populated with parameters entered by the user for the last invocation of the HFS Path list window.

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:

```
>>-- LVR -----+-----><
      |             |
      +--- cblvcat_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:

```
>>--+ LX -----+-----+-----+-----+-----+-----+-----+-----+-----+-----><
      |           |           |           |           |           |           |           |           |
      +- LISTEXTE -----+ +--- volume ---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----><
                                         |           |           |           |           |           |           |
                                         +--- filter ---+-----+-----+-----+-----+-----+-----+-----+-----+-----><
```

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.

filter

Note: The filter parameter is not yet implemented for VSE.

Select only catalog entries that match the specified filter mask. The filter 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 filter that does not contain an * (asterisk) wild card will be appended with *.* to list those data sets whose names begin with the filter string.

This parameter is placed in the Filter 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 ---+-----+-----+-----+-----+-----+-----+-----+-----+-----><
      |           |           |           |           |           |           |           |           |
      +- MAX -----+ +--- windowname ---+-----+-----+-----+-----+-----+-----+-----+-----+-----><
```

Description:

This command maximises the specified window.

This command is equivalent to selecting the **Maximise Button** of the window to be maximised.

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.

MDINEXT

Syntax:

```
>>--+ MDINEXT -----><
|
|      +-- + (plus-sign) --+
|
+-- Window -+-----+><
```

Description:

For use in MDI applications only (e.g. CBL e and SELCOPY Debug), this command sets the **focus window** to be the next 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 last created to the first created.

MDIPREV

Syntax:

```
>>--+ MDIPREV -----><
|
+-- Window - (minus-sign) --+
```

Description:

For use in MDI applications only (e.g. CBL e 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.

MINIMISE

Syntax:

```
>>--+ MINIMISE ---+-----><
|      |      |
+-- MIN -----+ +- windowname -+
```

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.


```
MW BY X=2 Y=-5
```

Window is moved 2 columns to the right and 5 columns upwards.

See Also:

[SIZEWINDOW](#)

NEXTMAINWINDOW

Syntax:

```
>>--+- NEXTMAINWINDOW -+-----><
      |                   |
      +- NMW -----+
```

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 CBLc, 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:

```
>>--+- NEXTWINDOW -+-----><
      |                   |
      +- NW -----+
```

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.

POWER

Syntax:

```
>>-- Power -+-----+-----><
           |         |
           +- power_command -+
```

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:

```
>>--+ PREVMAINWINDOW -+-----><
      |                   |
      +- PMW -----+
```

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 CBLLe, SELCOPY Debug, CBLVLCAT 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 CBLLe 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:

```
>>--+ PREVWINDOW -+-----><
      |                   |
      +- PW -----+
```

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.

QUICKREF

Syntax:

```
>>--+ QUICKRef  ---+-----><
      |           |
      +- QRef  -----+
```

Description:

Display the SELCOPY/i Quick Reference help.

QUIT

Syntax:

```
>>-- QUIT -----><
```

Description:

Use QUIT to exit and close the current SELCOPY/i window.

If the current window is the SELCOPY/i main window, then a pop-up window prompts the user to confirm whether or not to quit the SELCOPY/i session.

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 ----+- valid ----+- : -- 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:

- valid* 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 sequential 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.
    
```

RESTORE

Syntax:

```

>>--+ RESTORE -+-----+<<
      |         |         |
      +- RES -----+ +- windowname -+
    
```

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:

```

>>--- RETRIEVE ---+--- + (plus) ---+-----+<<
                |         |
                +--- - (minus) ---+
    
```


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 SELCOPY/i Structured Data Environment (**SDE**).

The SDATA command allows SDE commands to be issued from any SELCOPY/i window.
If the CBL e text editor main window has been stopped, SDATA will start the CBL e main window and open an edit view for the user's **HOME** CMX file before executing the SDATA command.
Also see the **SDATA** CBL e CLI command.

Parameters:

sde_command
Any **SDE** command.

SDE

Syntax:

```
>>-- SDE -----+-----+-----><
                |         |
                +-- sde_command --+
```

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:

```
>>-- SDFS -----><
```


-LIB *libpath*

For **MVS SELCOPY** jobs only, 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 control which SELCOPY module is executed and also any routines executed via the SELCOPY CALL statement.

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.

-PGM *loadmod*

Use an alternative load MODULE or PHASE name in place of the default name, SELCOPY.

-PSB *psbn*

For **MVS IMS SELCOPY** jobs only, the Program Specification Block name to be used to access DL1 data.

-DLI
-BMP

For **MVS IMS SELCOPY** jobs only, use either the DLI region or BMP region.
Default is -DLI.

-IMSID *ssn*

For **MVS IMS SELCOPY** jobs only, the sub-system name of the IMS Region to which to connect.

-AGN *grpn*

For **MVS IMS SELCOPY** jobs only, the IMS Application Group name.

Examples:

```
SELC -CTL CBL.SSC.CTL(DIRD01)
```

Start the SELCOPY Debug window and load control statements from the DSN CBL.SSC.CTL(DIRD01).

```
SELCOPY -CTL SYSIN
```

Start SELCOPY Debug using control statements in the currently allocated filename SYSIN.

```
SELCOPY -CTL SYS3.CBL.SELCOPY.CTL001 -LIB "DEV.CBL.LOAD DEV.TEST.RTN001.LOAD"
```

Start SELCOPY Debug using control statements in data set SYS3.CBL.SELCOPY.CTL001. Search load libraries DEV.CBL.LOAD then DEV.TEST.RTN001.LOAD ahead of the search chain when locating the SELCOPY executable module and any modules called 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 CBL **ALLOCATE** command is first used to allocate a concatenation of two DSNs to DDname, INCTL. These commands should be entered in a CBL edited CMX file and executed using ACTION (F16).

SETCOLOUR

Syntax:

```

          +-----+
          v       |
>>--+ SETCOLOUR -+--+ ttréf -- ttval -+-----><
          |
          +- SC -----+

```

Description:

Use SETCOLOUR to remap the appearance 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:

		<i>2nd Char</i>			
		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.

		<i>2nd Char</i>			
		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.

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 characteristics as List windows. For example select, sort and filter to display new views of the data are supported.

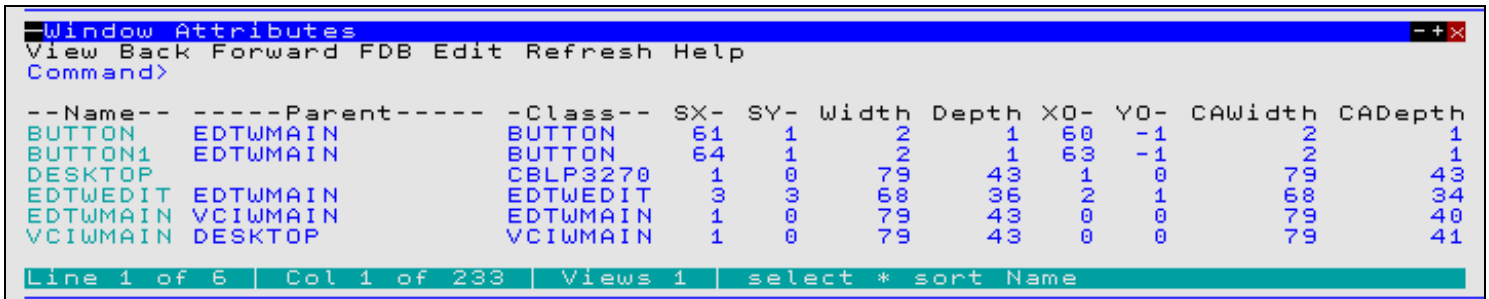


Figure 227. Window Attributes Window.

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
CApDepth	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

SIZEWINDOW

Syntax:

```
>>--+ SIZEWINDOW -+-----+-----+-----+-----+----->>
      |           |           |           |           |           |
      +- SZW -----+ +- windowname -+ +- BY -+ +- D=n -----+
                                     |           |
                                     +-----+ W=m -+
                                     |           |
                                     +- TO -+
                                     |           |
                                     (1)
                                     D=n -- W=m
```

Notes:

1. 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 CBL 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. If not supplied then the window in which the command is issued (via a command line or function key) is assumed.

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.

Examples:

```
SIZEWINDOW TO D=21
```

Focus window is resized to display 21 rows. Number of columns is unaltered.

```
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

SET WINSIZE CBL edit command for resizing the current CBL edit view or frame window.

SVC

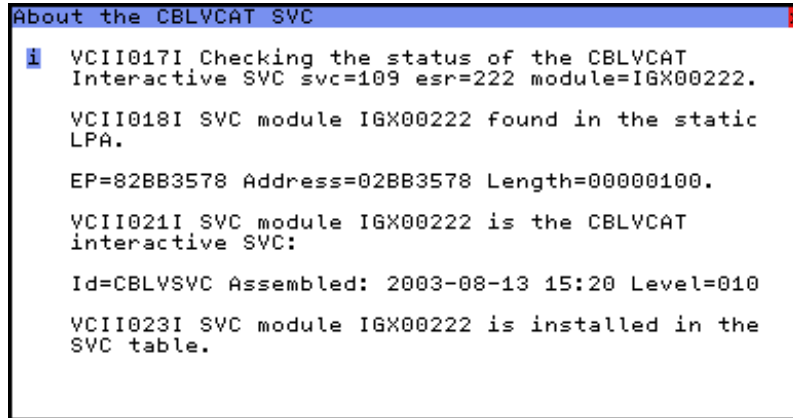
Syntax:

```
>>--- SVC -----><
```

Description:

For MVS only, use the SVC command to display the **CBLV CAT SVC** window containing the current status of the CBLV CAT Interactive (VCI) SVC required for LISTV CAT operations.

The CBLV CAT SVC window may also be opened via the System menu of the SELCOPY/i main window menu bar.



```

About the CBLV CAT SVC
i VCII017I Checking the status of the CBLV CAT
  Interactive SVC svc=109 esr=222 module=IGX00222.

VCII018I SVC module IGX00222 found in the static
  LPA.

  EP=02BB3578 Address=02BB3578 Length=00000100.

VCII021I SVC module IGX00222 is the CBLV CAT
  interactive SVC:

  Id=CBLVSVC Assembled: 2003-08-13 15:20 Level=010

VCII023I SVC module IGX00222 is installed in the
  SVC table.
  
```

Figure 228. CBLV CAT SVC Window.

SYSAPF

Syntax:

```
>>--- SYSAPF -----><
```

Description:

Use the SYSAPF command to open the **APF List** window.

The APF List window may also be opened via the System menu of the SELCOPY/i main window menu bar.

Note: Not valid for CMS and VSE.

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 -----><
      |           |
      +- SYINFO -+
```

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.

The Link List window may also be opened via the System menu of the SELCOPY/i main window menu bar.

Note: Not valid for CMS and VSE.

SYSLPA

Syntax:

```
>>-- SYSLPA -----><
```

Description:

Use the SYSLPA command to open the [LPA Modules](#) window.

The LPA Modules window may also be opened via the System menu of the SELCOPY/i main window menu bar.

Note: Not valid for CMS and VSE.

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.

The Task List window may also be opened via the System menu of the SELCOPY/i main window menu bar.

Note: Not implemented for CMS and VSE.

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 CBL e 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 CBL e 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.

UP

Syntax:

```
>>-- UP -----><
      |         | |         |
      +- windowname -+ +- CURSOR ---+
                        +- DATA ----+
                        +- HALF ----+
                        +- MAX ----+
                        +- PAGE ----+
                        +- n_lines --+
```

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 default 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:

```
>>-- VCat -----><
      |-----|
      |  cblv_syntax  |
      |-----|
      |  <  cblv_ctl  |
      |-----|
```

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:

```
>>+- View -----><
      |-----|
      | Browse +- | fileid -----|
      | (1)      |                |
      |          | +- ( +- PROFILE macroname -----|
      |          | |     | |                |
      |          | +- NOPROFile -----+ +-| HFS Opts |+-
```

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 CBL e text editor main window has been stopped, VIEW will start the CBL e 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 SELCOPY/i **EDIT** command.

Parameters:

fileid

The fileid of the file to be viewed.

For MVS, *fileid* may be the DSN of a sequential or VSAM data set, the member name (with or without the library DSN) of a PDS/PDSE member or an HFS file name. If member name is specified without a library DSN, the DSN of the library member in the current edit view is used.

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 SELCOPY/i 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 CBL e command SET DEFPROFILE (default PROFILE.)

The macro name must exist in a library within the CBL e 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 CBL e CLI command **EDIT** for supported HFS parameters.

VOLSTATS

Syntax:

```
>>--+ VOLSTATS  +--+-----+----->>
      |          |  |         |
      +- VOL -----+  +- volser  -+-
```

Description:

Use the VOLSTATS command to open a DASD Volume Statistics window and optionally display statistics for a specific DASD volume.

The DASD Volume Statistics window may also be opened via the prefix command **V** on any list window entry containing a volser field.

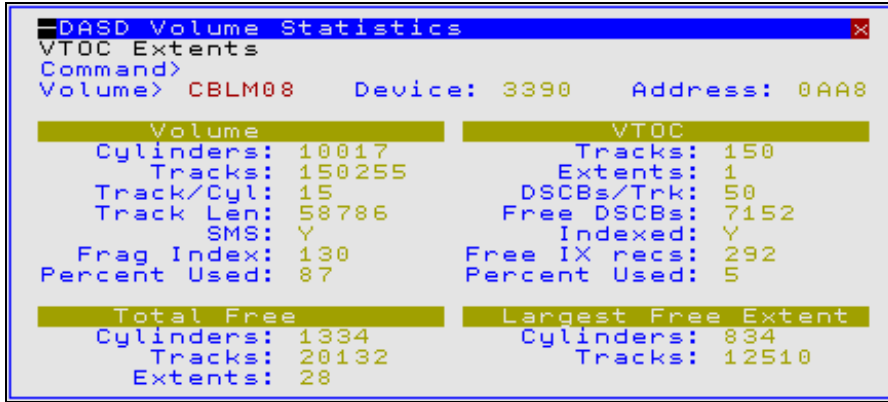


Figure 229. DASD Volume Statistics Window.

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:

```

>>- XMLgen ----->
      +-| Input dataset specification  |--+
>----->
      +-| Output dataset specification  |--+
>----->
      +-| Non-printable characters option |--+
>----->
      +-| XML special characters option  |--+
>----->
      +-| Invalid data values option    |--+
>----->
      +-| Character conversion option    |--+

+- NOSPLit +- NOREDefines +- NOFILLer +- COMments +-
|          |          |          |          |
>----->
+- SPLit ---+ REDefines ---+ FILLer +- NOCOMments +-
|          |          |          |          |

+- INDent 1 +- BROWSE +-
|          |          |
>----->
+- INDent n +- EDIT ---+
|          |          |
+- NOVIEW +-

```

Input dataset specification:

```

>- INDSN ( input_dataset_specification ) ----->

```

Output dataset specification:

```

>- OUTDSN ( output_dataset ) ----->
      +- EOL NL -----+
      |          |          |
      +- MODify +- NL -----+
      |          |          |
      +- APPend +- EOL -----+
      |          |          |
      +- CR -----+
      +- LF -----+
      +- CRLF -----+

```

Non-printable characters option:

```

>-NONPrint ----->
      +- HEX -----+
      |          |          |
      +- CHARS +- ASIS -----+
      |          |          |
      +- SKIP -----+
      |          |          |
      +- REPlace +- '.' -----+
      |          |          |
      +- WITH +- character -----+
      |          |          |
      +- 'character' ---+
      +- "character" ---+
      +- C'character' +-
      +- X'hex_value' +-
      +- HEX -----+

```

XML special characters option:

```

+-- ESCape -----+
|                 |
>- SPECIAL +-----+-----+
|           |           |
+-- CHARS -+ +-- CDATA -----+
|           |           |
|           |           |
+-- HEX -----+
|           |           |
|           |           |
+-- REPlace +-----+-----+
|           |           |
|           |           |
+-- WITH -+ +-- character ----+
|           |           |
|           |           |
+-- 'character' --+
|           |           |
+-- "character" --+
|           |           |
+-- C'character' -+
|           |           |
+-- X'hex_value' -+
|           |           |
+-- HEX -----+

```

Invalid data values option:

```

+-- HEX -----+
|                 |
>- INValid +-----+-----+
|           |           |
+-- VALues + +-- SKIP -----+
|           |           |
|           |           |
+-- REPlace +-----+-----+
|           |           |
|           |           |
+-- WITH -+ +-- character ----+
|           |           |
|           |           |
+-- 'character' --+
|           |           |
+-- "character" --+
|           |           |
+-- C'character' -+
|           |           |
+-- X'hex_value' -+

```

Character conversion option:

```

+-- terminal_ccsid --+
|                 |
>-+-- CONVert +-----+-----+-----+-----+-----+
|           |           |           |           |           |
|           |           |           |           |           |
+-- TO -+ +-- FFrom -+ +-- from_ccsid -----+
|           |           |           |           |           |
|           |           |           |           |           |
+-- ASCII -----+
|           |           |           |           |           |
+-- UNICODE -----+

```

Description:

The XMLGEN command is used to produce an exportable copy of a structured dataset as XML text.

XMLGEN 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 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**

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 current structured dataset (browsed or edited) is used. If there is no current structured dataset the command terminates with an error message.

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.

Output dataset specification

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

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'FFFFFFF' would have its value represented as

```
X&apos;FFFFFFF&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 will have the attribute **NONPRINT_CHAR="REPLACE_HEX"**.

XML special characters option

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 as is 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 will have the attribute **SPECIAL_CHAR="REPLACE_HEX"**.

Invalid data values option

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

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 transformed to the most recent UTF-16 standard.

Output line splitting option

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

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 (FILLER) field option

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

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.

XML nested tag indentation option

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.

INDENT *n*

The indentation value (default 1).

Output view option

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.

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 C98444444444444444444
      168500000000000000000 465000000000000000000
00000002 Amy          Johnston    28          41500.00
      C9A44444444444444444 D989AA99444444444444
      148000000000000000000 168523650000000000000
00000003 Freda         Bloggs      39          27800.00
      C9888444444444444444 C9988A444444444444444
      695410000000000000000 236772000000000000000
00000004 *** End of Data ***
```

The following XMLGEN command then produces the output file CBL.XMLGEN:

```
xmlgen indsn(cbl.xmllexamp using cobol cbl.cob(xmllexamp))
      outdsn(cbl.xmlgen)
      nocoment 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 CBL 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 CBL 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.

SELCOPY/i VTAM commands

Commands may be passed to the SELCOPY/i 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 SELCOPY/i VTAM. e.g.

```
MSG F8,DATA=command
```

Command	Description
MESSAGE	Send a text message to one or more users logged on to SELCOPY/i VTAM.
QUERY	Query the SELCOPY/i VTAM environment.
STOP	Stop SELCOPY/i VTAM.

MESSAGE

Syntax:

```
>>--+ MESSAGE +--+ user +--+ text -----><
      |         |   |   |
      +- MSG -----+ +-- * ----+
```

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:

```
          +- Users -+
          |         |
>>-- Query -+-----+-----><
```

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
```

SELCOPY/i Dump Files

SELCOPY/i dump files are supported for SELCOPY/i running in MVS environments only.

In order to assist CBL software engineers to correct any defects encountered in the SELCOPY/i system and programs, SELCOPY/i dump files exist to store formatted storage dumps.

By default, the **System.AbendTrap** variable is set **ON** in the SELCOPY/i System INI file. Therefore, in the event of a program check or program abend occurring which ultimately halts the CBL interactive environment, a message is sent to the user and control is passed to SELCOPY/i's abend handler routines.
If AbendTrap is set **OFF**, any abnormal program end is handled by the operating system.

Each time the SELCOPY/i abend handler is called, a new dump file is allocated with DSN prefix qualifier(s) determined by the **System.DumpDSNPrefix** variable in the SELCOPY/i 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 SELCOPY/i, 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 SELCOPY/i dump file to CBL file is likely.

Appendix A - SELCOPY/i Window Classes

The following table identifies the SELCOPY/i Windows Classes.

Window Class Name	Window Class Description
VCIWMAIN	SELCOPY/i 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
SDBWDEBUG	SELCOPY Debug main window
SYSIN	SELCOPY Debug SYSIN Text Edit document view
SYSPRINT	SELCOPY Debug SYSLSL/SYSPRINT Text Edit document view
WTOLG	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 [CBL text editor](#) 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 [CBL 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.
- 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.
- J** Submit the library member entry to batch. Executes the CBL CLI SUBMIT command using the entry name as input. (A CBL frame window must be active for this operation to succeed.) 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.
- 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.

```

FILEUTIL: NBJ.CBL.EXEC
Select option using point-and shoot,
by PF key or by entering option number.

>>> --- Create CLI (Command Line Interface) in temp CMX file ...
1. Text EDIT (PF1)
2. ALLOCate (PF2)
3. Exit utilities menu without action (PF3)
4. EQU (%VariableName%) (PF4)

Miscellaneous ...
5. ALLOCate Dialog (fileid as model) (PF5)
6. List ALLOCs for fileid (PF6)
7. Create // DD card (PF7)
8. Create SELCOPY batch job (PF8)
9. Create SELCOPY foreground job (PF9)

```

Figure 230. File Utilities Menu.

Options are selected by entering the required option number at the command prompt or executing the equivalent PFKey.

On selecting one of these 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

V Open 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.

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 the DASD Volume Statistics window for the volume in the list 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 PF2 by default.

Command Cross-Reference

	Prefix Commands																																	
	A	AS	AP	B	C	CF	CL	D	E	EU	EX	F	FO	FS	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	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	Y
LVOL	-	-	-	-	-	-	-	-	-	-	-	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	
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	
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	
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	
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	
Lab	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	Y	Y	
LL	2	-	-	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	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	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 background 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 CBL. 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.