



Product Suite New Features

Release 3.30

IBM Mainframe z/OS, VSE & VM/CMS Systems

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

SELCOPY Product Suite 3.30 New Features.....	1
Documentation Notes.....	1
SELCOPY Product Suite Components.....	2
SELCOPYi.....	2
SELCOPY.....	2
SELCOPY (SLC) C++ Version.....	2
CBLVCAT.....	2
SELCOPYi Text Editor.....	3
Processing Enhancements.....	3
Edit RECFM=U Datasets.....	3
Escaped Quotes in Search/Replace Strings.....	3
Save Window Size/Location.....	3
Line Command Repetition Factor.....	4
Line Commands.....	4
AK - Intermediate After Line Target.....	4
BK - Intermediate Before Line Target.....	4
O, OO, OK, OOK - Overlay Line(s) Target.....	5
SET/QUERY/EXTRACT Options.....	5
ACTIONCOMMENT.....	5
HIGHLIGHT.....	6
SELCOPYi Data Editor.....	7
Processing Enhancements.....	7
Large Format Datasets.....	7
Structure (SDO) Creation.....	7
Structure to Dataset Association.....	8
DB2 Table Edit/Browse Processing Enhancements.....	9
Related Table Edit.....	9
Row Selection.....	10
Remote Location Table Edit/Browse.....	11
Distinct Type Columns.....	11
Variable Length Character Columns.....	11
NULLable Field Columns.....	12
New Row Highlighting.....	12
DB2 Table Edit Utility Popup Menu.....	12
Primary Commands.....	13
BROWSE.....	13
CREATE STRUCTURE.....	14
EDIT.....	15
ERROR.....	16
INFO.....	17
POP.....	17
PUSH.....	17
REDIT.....	17
SORT.....	18
SORTINDEX.....	19
Line Commands.....	19
E - Expand SQL Error.....	19
RE - Related DB2 Table Edit.....	19
SET/QUERY/EXTRACT Options.....	20
AUTOSTRUCTURE.....	20
COLOUR.....	20
COMMIT.....	21
DFPSCALE.....	21
EDITPRIMEKEY.....	22
NULLCHAR.....	22
NULLIFBLANK.....	22
STRUCTURE.....	23
VENDCHAR.....	23
VERSION.....	24
VSHOWEND.....	24
VSTRIP.....	24
Functions.....	25
SQLERROR().....	25
Panels.....	25
Create Structure from Copybook(s).....	25
Structured Data Browse/Edit.....	25
Select Display Fields.....	25
DB2 Row Selection.....	26
DB2 Edit Object / DB2 Browse Object.....	26
DB2 Select Table Index.....	26
SELCOPYi Utilities.....	28
CSV Generation.....	28
CSVGEN Primary Command.....	28

Contents

SELCOPYi Utilities

CSV Generation Panels.....	29
Data Set Information (DSI).....	30
DSNTYPE=LARGE Datasets.....	30
DB2 Table Information.....	30
DB2 Object Creation Panels.....	31
Create Index.....	31
Create Clone Table.....	31
General Enhancements.....	32
DB2 SQL Execution (EXECSQL).....	32
EXECSQL Primary Command.....	32
EXECSQL Panels.....	33
DB2 Table Compare.....	34
COMPTABLE Primary Command.....	34
COMPFILE Primary Command.....	34
Compare Base/Result Table Panels.....	35
File Compare.....	35
File Copy.....	36
File Erase.....	36
File Merge.....	37
MERGE Primary Command.....	37
Merge Datasets Panel (=8.18).....	37
Print.....	38
PRINT Primary Command Enhancements.....	38
SDE Print File Panel.....	39
SDEMAIN (SELCOPYi SDE batch).....	39
SDE Rexx Macro Support.....	39
SELCOPY Debugger.....	40
JCL Job Stream Input.....	40
XML Generation.....	41
XMLGEN Primary Command Enhancements.....	41
XML Generation Panels.....	42

SELCOPYi Lists.....44

General List Enhancements.....	44
PFKey Assignment.....	44
SMS Storage Groups (=3.14).....	44
LSG Primary Command.....	44
SMS Storage Groups List.....	44
SMS Storage Group Volumes (=3.15).....	45
LSGV Primary Command.....	45
SMS Storage Group Volumes List.....	45
Library Member Move/Copy.....	46
DB2 Objects.....	47
Index Keys.....	47
Packages.....	47
Package Dependencies.....	48
Package Privileges.....	48
Related Tables.....	49
List DB2 Object Enhancements.....	49

SELCOPYi Panels.....50

Processing Enhancements.....	50
Dataset Selection.....	50
Embedded Table Primary Commands & Options.....	50
CCOLOUR SET/QUERY/EXTRACT.....	50
REFRESH.....	51
SELECTALL.....	51
VLMAX.....	51
VLMIN.....	51

SELCOPYi Miscellaneous Enhancements.....52

3390 Extended Address Volumes (EAV).....	52
DASD Volume and Dataset Lists.....	52
Large Format Datasets.....	53
FIRSTUSE utility.....	53
Automatic Execution.....	53
INI Dataset Update.....	53
Window Manipulation.....	53
Initial Windows State.....	53
MOVEWINDOW/SIZEWINDOW Primary Commands.....	54
PFKey Definitions.....	54
PFSHOW Primary Command.....	54
PFSHOWSTYLE Primary Command.....	54
TITLEBAR and BORDER PFKey Display.....	54
Regular Expression Enhancement.....	55
SELCOPYi Session Count.....	55

Contents

Maintenance Applied.....56

SELCOPY Product Suite 3.30 New Features

Documentation Notes

Information in this New Feature List details changes introduced to CBL software since SELCOPY Product Suite 3.20.

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

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

- SELCOPY Product Suite Customisation Guide
- SELCOPY User Manual
- CBLVCAT User Manual
- SELCOPYi Reference and User Guide
- SELCOPYi Text Editor (CBLe) Manual
- SELCOPYi Structured Data Editor Manual
- SELCOPYi Training Manual

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, 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.

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. 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 manuals.

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

- ZOS** - 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 ZOS, VSE and CMS operating systems.

SELCOPY Product Suite Components

SELCOPY Product Suite 3.30 includes updates to the following product components.

SELCOPYi

z/OS Systems

SELCOPYi has been updated extensively for z/OS systems. Details of SELCOPYi 3.30 new features comprise the remainder of this document.

z/VM and z/VSE Systems

Please be aware that features of SELCOPYi 3.30 detailed in this document are specific to **z/OS systems only**.

SELCOPY

The SELCOPY 3.30 (Assembler version) batch program has been updated with important fixes that are specific to both z/OS and z/VSE systems. These are as follows:

Ref: APAR QI32001 (s320z01)

Correct possible ERR501 or ERR502 condition that can occur when reading a file which has been closed and re-opened, and for which a specific BLKSIZE value exists within the SELCOPY control statements.

This is particularly pertinent to operation in z/VSE systems where file BLKSIZE is not maintained by the system and so is usually specified on a SELCOPY READ statement.

Ref: APAR QI32002 (s320z02)

SELCOPY I/O now treats IMS status code FH as an indication of end-of-file (EOF) but sets return code 8 to indicate an abnormal EOF. This allows the opportunity to test for this condition and act accordingly. e.g.

```
IF EOF                      * If end-of-input ...
OR POS STATUS = 'FH'        * ...or DEDB data area not accessible.
THEN PRINT FROM PCB         * Print the DB PCB control block.
THEN CHKP IMS               * Take a checkpoint
THEN GOTO EOJ               * End job.
```

IMS status code FH indicates that a DEDB area was inaccessible when SELCOPY attempted to execute an IMS DB function for the DEDB data base.

Previously, when status code FH was encountered on a READ operation, SELCOPY returned ERROR 541 (Unexpected Status Code) and processing ended without allowing the user to test for the condition and act accordingly (issue CHKP).

SELCOPY (SLC) C++ Version

The SELCOPY (SLC) 3.30 C++ Version batch program has been updated with new features which are detailed in the [SELCOPY C++ Version \(SLC\) 3.30 New Features](#) document. This publication includes features for SELCOPY (SLC) on all supported platforms including Microsoft Windows and i386 Linux.

Please note that licensees of SELCOPY on mainframe systems are entitled to download and install SELCOPY for Windows on Microsoft Windows workstations at the same installation. Please refer to the [CBL web site](#) for product download.

CBLVCAT

The CBLVCAT 3.30 batch program has undergone no changes since version 3.20 and these versions are functionally equivalent.

A known problem exists in IBM VSAM code when interactive execution of CBLVCAT (in SELCOPYi) is performed on z/OS 2.1 systems. This type of CBLVCAT execution uses the supplied SVC module (ZZV00SVC).

Please refer to IBM [APAR OA43983](#) for problem description and required PTF SYSMOD UA72646 which resolves this issue.

SELCOPYi Text Editor

SELCOPYi 3.30 includes a number of enhancements to the SELCOPYi text editor.

Processing Enhancements

The following new features have been introduced which relate to general operation of the SELCOPYi text editor.

Edit RECFM=U Datasets

Support text edit of undefined record format (RECFM=U) physical sequential datasets. Note that edit of these types of dataset is already supported by the Data Editor.

The maximum record length (LRECL) value used for edit of a RECFM=U dataset is the allocated physical record size (block size). If the dataset has not yet been allocated a block size (i.e. BLKSIZE=0) and the allocated LRECL value is not zero (0), then BLKSIZE=LRECL is used. Otherwise, if the LRECL value is also zero, an error is returned.

Edit of a RECFM=U format PDS/PDSE member is prohibited in order to protect module/program members of a load library from accidental update. If an attempt is made to edit this type of entry using the text editor, a read only edit view of the member is opened and warning ZZSE200W is returned. Similarly, if the data editor is used, a browse view of the data is opened and warning ZZSD667W is returned.

When editing text for a new, as yet unallocated dataset name, SET DSORG and SET RECFM may be used to set the dataset organisation and record format respectively. An attempt to set a DSORG=PO and RECFM=U will return error message ZZSE201E, ZZSE202E or ZZSD668E.

Escaped Quotes in Search/Replace Strings

Text edit primary commands FIND, CHANGE, EXCLUDE, ONLY, ALL, LESS and MORE, locate text in the focus edit view using a search string and, in the case of CHANGE, replaces the text with a specified replace string.

A search and replace string may be specified as a delimited string or character string. Both types involve enclosing (delimiting) a string in either apostrophe (') or quotation mark (") characters.

Since occasions arise where the required search or replace string text includes both of these delimiter characters, support has been included to be able to escape a delimiter character within the string text. The escape character is the same as the string delimiter character. e.g. In the following, apostrophe is the delimiter/escape character so that two adjacent apostrophes are treated as a single occurrence of an apostrophe within the string text.

```
FIND 'He said "It''s my brother''s car."'
```

Note that use of an escape character is already supported by Data Editor FIND, CHANGE, etc. primary commands and expressions.

Save Window Size/Location

When operating in a windowed (i.e. non-maximised) SELCOPYi window display, support already exists to save and restore the window location and size of a text edit or data edit/browse document window view used to display data belonging to a specific file.

This is achieved using the wS (window save) and wR (window restore) selectable items displayed to the right of the Text Editor menu bar items. By default, window save and restore operations are also assigned to F13 and F14 respectively when the cursor is positioned in the title bar of an edit or browse document window view.

In previous releases, performing a window save would prompt the user to accept or update the suggested name under which the window location and size is recorded. Since the default name on window restore is the fileid (DSN) associated with the focus window, so too is the name suggested by SELCOPYi on window save.

Since, in most cases, there is no need to use any other window save/restore name, the prompt to accept fileid as the window save/restore name is an irritation and so has been suppressed in SELCOPYi 3.30. Instead, fileid is used automatically and a "Window location saved" message is displayed.

Specification of a different name is still possible using the supplied WINX macro. Enter EM WINX to display the macro source and syntax options.

Line Command Repetition Factor

Text edit supports specification of line command pairs in the number prefix area which represent a block of lines on which the command will operate (e.g. CC, MM, RR).

Where a line command block supports a repetition factor (i.e. RRn , $((n,))n$, $<<n$ and $>>n$ where n is the repetition factor), then support has been added to allow specification of the repetition factor on either the first or last of the line command pairs delimiting the block of lines. In previous releases, specification of the repetition factor was restricted to the line command block delimiter that occurred first within the edit view.

Line Commands

The following Text Editor prefix area line commands have been introduced in SELCOPYi 3.30.

AK - Intermediate After Line Target

Syntax:

```

      +-- 1  +--
      |      |
>>+--- A  +---+-----+-----><
      |      |      |
      +--- AK +---+-----+-----
          +-- n  +--
  
```

Description:

The A and AK line commands identify the line **after** which text marked for copy (C) or move (M) will be inserted.

Support for AK is introduced in SELCOPYi 3.30 and identifies the line as being an intermediate line target, i.e. just one of a number of line targets at which the text will be moved or copied. Another line target (A, AK, B, BK, O, OK, OO or OOK) must exist on a subsequent line to identify an additional destination for the copied or moved text. If this is also an intermediate line target, then another line target must follow. This repeats until a non-intermediate target is specified to identify the last target in the copy/move transaction.

The repetition factor n identifies the number of times the copied/moved text will be inserted after the line.

BK - Intermediate Before Line Target

Syntax:

```

      +-- 1  +--
      |      |
>>+--- B  +---+-----+-----><
      |      |      |
      +--- BK +---+-----+-----
          +-- n  +--
  
```

Description:

The B and BK line commands identify the line **before** which text marked for copy (C) or move (M) will be inserted.

Support for BK is introduced in SELCOPYi 3.30 and identifies the line as being an intermediate line target, i.e. just one of a number of line targets at which the text will be moved or copied. Another line target (A, AK, B, BK, O, OK, OO or OOK) must exist on a subsequent line to identify an additional destination for the copied or moved text. If this is also an intermediate line target, then another line target must follow. This repeats until a non-intermediate target is specified to identify the last target in the copy/move transaction.

The repetition factor n identifies the number of times the copied/moved text will be inserted before the line.

HIGHLIGHT

Syntax:

```

                                +- ON --+
>>+-----+-----+ HIlite --+-----+-----+><
    |      |      |      |      |      |
+- SET -----+ +- HIlight -+ +- OFF -+ +- AUTO -----+
                                |
                                +- ... -----+

>>--- Query -----+ HIlite --+-----+-----+><
                |      |
                +- HIlight -+

>>--- EXtract -- / +- HIlite --+ / -----+-----+><
                |      |
                +- HIlight -+

```

Description:

HIGHLIGHT parameter AUTO is now supported in SELCOPYi 3.30.

Text edit syntax colour highlighting is sensitive to the language or format of the edited source. If HIGHLIGHT AUTO is selected, the language assigned to the source text is determined automatically by SELCOPYi. This selection is based on the contents of the first non-blank line and, if this is inconclusive, the last qualifier of the dataset name.

SELCOPYi Data Editor

Whereas the SELCOPYi Text Editor provides powerful facilities for editing character based, plain text data, the SELCOPYi Data Editor provides facilities to perform edit on data comprised of different data type formats mapped by established structures (e.g. as defined by COBOL, PL1, Assembler and DFSORT).

Use of structures extends to other structured data edit (SDE) utilities (e.g. File Copy, File Search/Update/Remap, File Compare, Print, XML and CSV Generation) which all support processing of data at a record field level.

The SELCOPYi Data Editor also supports edit of DB2 tables, edit of data sets too large to be loaded into available storage and also has special provision for edit of VSAM data sets.

SELCOPYi 3.30 includes a number of enhancements to the SELCOPYi data editor.

Processing Enhancements

The following new features have been introduced which relate to general operation of the SELCOPYi data editor.

Large Format Datasets

Physical sequential (DSORG=PS) datasets allocated as DSNTYPE=LARGE may grow beyond the 65,535 tracks per volume limit for normal PS files, with a maximum of 16,777,215 tracks per volume.

In order to accomodate a number of tracks greater than 99999, the record information identification column display, controlled by the SET/QUERY/EXTRACT RECINFO option, has been amended for non-VSAM, non-HFS datasets.

The record identification information displayed for non-VSAM, non-HFS datasets is comprised of a relative volume number (if a multi-volume dataset), volume track number, track physical record (block) number and physical record offset.

In previous releases, these were represented by columns with header name (width): Vol (3), TTR (5 and 3) and Offset (6). As of SELCOPYi 3.30, these values are represented by columns with header name (width): Vol (3), Track (8), Blk (3) and Off (5).

Structure (SDO) Creation

SELCOPYi Structured Data Edit (SDE) uses Structure Definition Object (SDO) structures to apply formatting to record data and DB2 table rows. These SDO structures are in a format used only by SELCOPYi SDE applications.

In addition to record formatting or DB2 table row information (record-types), an SDO may also contain the following information:

- A 64-character structure title and/or 512-character description.
- Record-type selection criteria for data comprising records or record segments mapped by different record formatting.
- Column selection, column sequencing and column width definitions.
- DB2 row selection (WHERE clause) and row sequencing (ORDER BY clause). Note that, for SDE applications that use non-DB2 table SDOs, record selection (filtering) based on WHERE expressions may be achieved using a Filter file.
- DB2 commit, isolation level, locked row, audit and primary key options.
- Record-type colouring based on WHERE expressions.

An SDO may be loaded from an existing library member or sequential data set, or generated in local storage via a DB2 SQL query or an existing COBOL or PL1 copybook as required by the Data Editor or the SDE utility. Once in storage, an SDO remains there until dropped either explicitly, via the DROP command or a panel recompile specification, or implicitly when the SELCOPYi session is ended.

If generated automatically by an SDE application or created via the primary command CREATE STRUCTURE with parameter TEMPORARY, the SDO structure is **temporary** and is not automatically saved to DASD.

SELCOPYi introduces the following enhancements to SDO structure creation:

- Permanent updates to column selection, column sequencing and/or column width definitions within a temporary SDO structure will open the "Specify new Structured Data Object (SDO) name" panel in order to make a permanent copy of the structure. Note that permanent updates may be applied using the "Select Display Fields" panel or the PERMANENT SELECT/COLWIDTH primary command.
- SAVESTRUCTURE primary command has been updated to support specification of the following:

1. The in-storage SDO structure name to be saved. Default is the current (i.e. last referenced structure).
 2. The fileid (library member or sequential DSN) to which the in-storage SDO structure will be saved.
- In addition to creating SDO structures from COBOL and PL1 copybook source members or ADATA datasets generated from a compile of these types of copybook, SDO structures may now also be created from the following source input:
 1. HLASM (Assembler) DSECTs.
 2. HLASM ADATA datasets that include one or more Assembler DSECTs.
 3. DFSORT SYMNAMEs files.

Structure to Dataset Association

SELCOPYi 3.30 introduces a facility to define and store an association between a data file name mask and a structure file (SELCOPYi SDO structure or COBOL, PL1, Assembler, ADATA source member).

On subsequent data edit/browse of any dataset (or HFS file) that matches this name mask, the defined association may be used to identify and automatically apply the structure used to format the file's data records. If implemented, this means that a structure need not be referenced directly in the Structured Data Edit/Browse panel or via the USING parameter of the EDIT/BROWSE primary commands.

Implementation of this facility and management of the structure to dataset name mask associations may be actioned via the Associations (option 6) item of the Data Edit Settings panel (=0.4) or via new data editor SET/QUERY/EXTRACT options **AUTOSTRUCTURE** and **STRUCTURE**. Note that association definitions are saved in the user's own SELCOPYi table library member, ZZSDSUSE, which gets automatically generated the first time an association is defined.

The table of association definitions may be viewed and modified by typing the **EDIT (E)** primary command from the "Manage Copybook Associations" panel (=0.4.6) or executing the Data Editor option STRUCT with no parameters (i.e. from outside the data editor, execute **SD STRUCT**). This table is presented as a structured data edit view which supports all the usual data editor primary and line-commands such as FIND, CHANGE, SELECT, WHERE, etc.

Each table row defines a relationship between a structure (**MappingFile**) and one or more data files (**DataFileMask**). Furthermore, if the structure refers directly to a structure which is COBOL, PL1, Assembler or ADATA source, then this must be indicated in the **Lang** column by COBOL, PL1, HLASM or ADATA respectively. For SDO structures, Lang may be specified as SDO or left as blank.

Standard pattern matching wildcards may be specified in the data file mask as follow.

- | | |
|----|---|
| * | 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.) |

An optional library member name mask may also be specified in "()" (parentheses) following a library DSN mask. This supports wildcards as follow.

- | | |
|---|--|
| * | 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. |

In identifying an association for a given data file name, a search of the table is performed in the following order of precedence, stopping when a match is found.

1. Search data file name masks that contain no wild card characters.
2. For library datasets with member name specification, search data file name masks that contain a parenthesised member name mask. Both the library DSN mask and member name mask can contain wildcard characters.
3. For library datasets with member name specification, search data file name masks that contain no wild card characters in the library DSN and no parenthesised member name mask.
4. Search all remaining data file name masks that contain wild card characters.

Note that, within the scope of the above search order, if the data file matches more than one data file name mask then the first matching association definition within the table will be selected.

By default, this facility is set on at a level so that any specification of a structure on data edit or browse will automatically be saved as an association definition for the specific data set name. Likewise, any data edit or browse without a structure reference will automatically interrogate the associations table and use the structure associated with a matching DSN mask. Option AUTOSTRUCTURE controls whether or not association definitions are saved, applied or both.

```

SELCOPY/i - Edit NBJ2.SELCOPYI.TLIB(ZZSDSUSE) using SYS00128.ZZSDSUSE.FDB0000
File Edit Actions Options Utilities Window SwapList Help  wS wR
Command>
Type SEL to control column widths etc. Type MAP for single record display mode.
Record type: ZZSDSUSE  Variable(6,1094)  Offset=0  Data elements=4
DataFileMask  Lang  MappingFile
<---+---1---+---2---+---3--->  <--->  <---+---1---+---2---+--->
0001 *...SDO LAC.CBL.SDO(SDO)
0002 *...ZOPS.** LAC.CBLI.SDO(ZOPS1)
0003 *.AM** CBL.CBLI.SDO(AM)
0004 *.CBLATRA** LAC.CBLI.SDO(CBLATRA)
0005 CBL.PL1.TXT.F110 PL1 CBL.PL1.COPYBOOK(IPIC002)
0006 *.SELCTRN.ZZST1DAT.** NBJ.SELCTRN.SDO(ZZST1)
0007 *.SELCTRN.ZZST2DAT.** NBJ.SELCTRN.SDO(ZZST2)
0008 *.SELCTRN.ZZST3DAT.** NBJ.SELCTRN.SDO(ZZST3)
0009 *.SELCTRN.ZZST5DAT.** NBJ.SELCTRN.SDO(ZZST5)
0010 /mnt/l08/bin/c/zos/CBL.**.SYSPUNCH CBL.CBLI.SDO(HFSV)
0011 /mnt/l08/bin/c/zos/CBL.**.SYSREC CBL.CBLI.SDO(HFSV)
0012 CBL.CBLI.ADA(*) LAC.CBLI.SDO(ASMADATA)
0013 CBL.CBLI.MBRLIST.** CBL.CBLI.SDO(MBRLISTR)
0014 CBL.CBLIMST.STDTEST.*.REC CBL.CBLI.SDO(PLAYBACK)
0015 CBL.CBLIMST.STDTEST.*.RECS CBL.CBLI.SDO(PLAYBACS)
0016 NBJ.SELCTRN.ZZST2DAT COBOL NBJ.SELCTRN.SAM1(ZZST2CPC)
Se | Line=1 | Col=1 | Alt=0,0;0 | Size=16 | Recl=1094 | Fmt=V | Files=1 | Views

```

Figure 1. SDE: Structure to Dataset Associations Table.

DB2 Table Edit/Browse Processing Enhancements

The SELCOPYi DB2 table edit/browse feature has received major new enhancements in SELCOPYi 3.30.

Related Table Edit

SELCOPYi 3.30 introduces specific support for editing DB2 tables which have relationships defined by referential integrity (RI) constraints.

An RI constraint establishes a parent-dependent relationship between two tables by means of a foreign key. A foreign key is a set of columns in the dependent table which correspond to a unique key in the parent table.

When an RI constraint exists certain types of change to the tables involved are not permitted. Breaking these rules leads to an **RI error** of one of the following types:

- Missing parent key **SQLCODE -530**. This happens when an insert or update of a row in a dependent table has a foreign key which does not represent an existing row in the parent table. In other words you cannot make orphans by inserting or updating foreign keys in the dependent table.
- Parent key update error **SQLCODE -531**. This happens when an update of a row in a parent table changes a key which is a foreign key in a dependent table and has dependent rows in that table. In other words you cannot make orphans by changing parent keys.
- RI delete rule violation **SQLCODE -532**. When the RI constraint was defined with a delete rule of **RESTRICT** or **NO ACTION**, this error will happen when deleting a row in the parent table which has dependent rows in the dependent table. In other words for this type of RI constraint deletes must be done from the bottom up.

If any of these RI violations should occur when data is saved (i.e. when SQL INSERT, DELETE and/or UPDATE is performed), each row in error will display the SQL code in the prefix area as R-530, R-531 or R-532. Note that non-RI SQL codes that occur are also displayed in the prefix area as W+nnn (successful execution SQL codes) and E-nnn (SQL error codes).

Execution of the new **RE** line command or **REDIT** primary command against a row flagged as an RI violation will open a related table edit window containing only those rows in the parent or dependent table for which the violation occurs. This allows the option of amending the related table data to preserve integrity. The RE line command and REDIT primary command may be used in a number of ways with tables related by RI constraints:

- List all parent and dependent relationships for the current table.
- Edit or browse the parent row or dependent rows of the **focus row** for any RI constraint.
- After an RI error when saving changes, edit the parent row or dependent rows of the row which had the error.

Alternatively, the **E** line command or **ERROR** primary command, introduced in SELCOPYi 3.30, may also be used to display the "DB2 Save SQL Error" panel for the focus row on which an SQL code has been flagged. The format of this panel differs based on the format of the SQL error.

- For non-RI SQL error codes, the panel displays only the details of the error message returned by DB2.

```

SELCOPY/i - DB2 Save SQL Error
File Edit Actions Options Utilities Window SwapList Help  WS wR  Scroll> Csr
Command>
DB2 has reported an SQL save error. The formatted message is:
DSNT408I SQLCODE = -803, ERROR: AN INSERTED OR UPDATED VALUE IS
INVALID BECAUSE INDEX IN INDEX SPACE ZZSSMOD# CONSTRAINS
COLUMNS OF THE TABLE SO NO TWO ROWS CAN CONTAIN DUPLICATE
VALUES IN THOSE COLUMNS.
RID OF EXISTING ROW IS X'0000000201'.
DSNT418I SQLSTATE = 23505 SQLSTATE RETURN CODE
DSNT415I SQLERRP = DSNXRINS SQL PROCEDURE DETECTING ERROR
DSNT416I SQLERRD = -110 13172739 0 13817814 -490143744 0 SQL
DIAGNOSTIC INFORMATION
DSNT416I SQLERRD = X'FFFFFFF92' X'00C90003' X'00000000'
X'00D2D7D6' X'E2C90000' X'00000000' SQL DIAGNOSTIC
INFORMATION

```

Figure 2. DB2: Save SQL Error - General.

- For RI SQL error codes, the panel also displays the referential constraint name, the delete rule, the parent and dependent table names and includes an embedded table detailing the parent and foreign key column names and the foreign key value. An Action input field prompts the user to do one of the following:

EXIT	Exit to the focus DB2 edit view.
CANCEL	Cancel from the focus DB2 edit view.
REDIT	Perform related table edit to update/delete dependent table rows.
INSERT	Perform related table edit to insert a parent table row.
MODEL	Perform related table edit to insert a parent table row which is modelled on an existing parent row.

```

SELCOPY/i - DB2 Save SQL Error
File Edit Actions Options Utilities Window SwapList Help  WS wR  Scroll> Csr
Command>
DB2 has reported a missing parent key error SQLCODE -530.
Relationship: ZZSSMOD_REF1 Delete rule: Restrict
Parent      : CBL.ZZSLIBS
Dependent   : CBL.ZZSSMOD

An UPDATE operation on the dependent table attempted to change the columns
of a foreign key to values which do not exist as a key in the parent table.

Action> _____ CANCEL Cancel the edit session
                  EXIT  Return to the edit session
                  MODEL Model the parent row on selected row(s)
                  INSERT Insert a default parent row with key values

Foreign Key
Parent column  Dependent column  Column value
+             +
000001 APILIB APILIB XNVLIB
000002 *** End of Data ***
1 Row

```

Figure 3. DB2: Save SQL Error -530 - Missing Parent Key.

Row Selection

The "DB2: Select table rows by column value" modal panel provides a tabular method of building a simple DB2 WHERE clause (search condition) comprising any number of logically connected (Basic, BETWEEN, IN, LIKE or NULL) predicates and/or search conditions based on named columns belonging to the nominated DB2 table or view. Specification of balanced left and right parentheses allows grouping and nesting of search conditions.

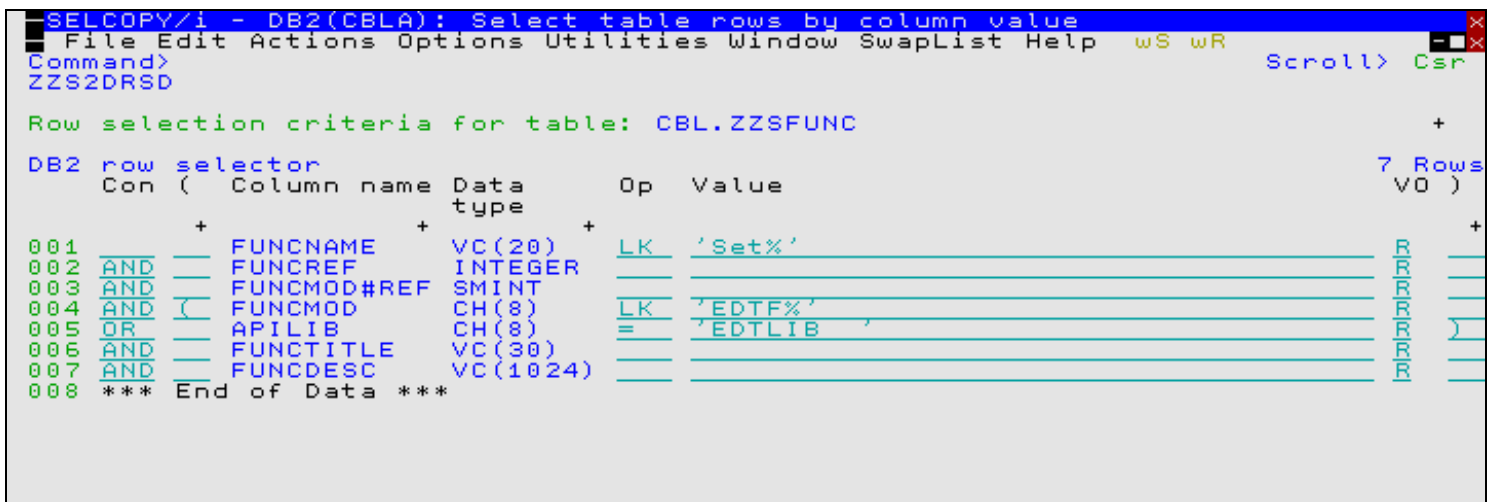


Figure 4. DB2: Row Selection Panel.

The panel is started by any of the following:

- Specify "?" (question mark) for parameter *where-clause* in the DB2 options of the SDE **EDIT**, **BROWSE** or **CREATE STRUCTURE** primary commands.
- Specify "?" in the Where> input field of the **Edit Object** or **Browse Object** panels.
- Specify MODEL in the Action> input field of the "DB2 Save SQL Error" panel. This option is available only if the DB2 table row in error has a missing parent key in a referential constraint relationship. (SQL error -530)

The panel table rows initially display an incomplete, unselected predicate for each column belonging to the DB2 table. A row entry becomes selected on entry of a valid predicate operator value in the Op field.

Selected rows are processed in ascending order of row number when building the search condition. Using table edit primary and/or line commands, the row entries may be re-ordered and duplicated to build the required WHERE clause.

Remote Location Table Edit/Browse

SELCOPYi 3.30 introduces support for edit and browse of tables and views at remote DB2 server locations. This is achieved by specifying the DB2 server location as the first qualifier (part) of a three-part table or view name. The DB2 server location name specified must be one of the unique names defined in the SYSIBM.LOCATIONS CDB table.

To accommodate this feature, various primary commands, panels, lists and reports have been updated as follow:

1. Data Editor **BROWSE**, **CREATE STRUCTURE** and **EDIT** primary commands updated to support a DB2 table or view name having a three-part reference. i.e. location.schema.name
2. DB2 CREATE TABLE utility panels now support modelling on table attributes on a DB2 table or view at a remote DB2 server location.
3. DB2 Object List panels now support "Location" input field in order to list objects at a remote DB2 server location. List prefix commands that display object lists relating to a particular list entry inherit the location value.
4. **DB2 Table Information** report generated by primary command DSINFO (or line command "I" in a DB2 tables list) displays the DB2 table location if not at the server location for the default connection.
5. **Related Table Edit** performed for the current DB2 table edit view, is sensitive to the table's DB2 server location.

Distinct Type Columns

Browse and edit of DB2 tables and views that include columns defined using a distinct data type are supported in SELCOPYi 3.30. Columns of distinct type are displayed with the correct built-in data type and length/precision as defined by the distinct type.

Variable Length Character Columns

DB2 table columns containing character data of variable length may include trailing blanks. To enhance support for these types of column, SELCOPYi 3.30 introduces the following:

1. Option to automatically strip trailing blanks from a column field if that field is updated. This may be set via the **VSTRIP** option or from the **"DB2 Table Edit Utility Popup Menu"**.

- Options to assign and control display of a delimiter character that marks the end of variable character data. This may be set via the **VSHOWEND** and **VENDCHAR** options or from the "DB2 Table Edit Utility Popup Menu"
- Option to assign a character to mark the end of variable character data on input. This may be set via the **VENDCHAR** option or from the "DB2 Table Edit Utility Popup Menu"

NULLable Field Columns

SELCOPYi 3.30 introduces the following features that enhance support for DB2 table columns defined as supporting NULL values:

- Option to control whether a blank field entry is treated as a NULL value. This may be set via the **NULLIFBLANK** option or from the "DB2 Table Edit Utility Popup Menu".
- Option to assign a character that marks the field as being NULL. This may be set via the **NULLCHAR** option or from the "DB2 Table Edit Utility Popup Menu"
- Option to assign a character that marks the field as being NULL on input. This may be set via the **NULLCHAR** option or from the "DB2 Table Edit Utility Popup Menu"

New Row Highlighting

When the **SAVE** operation is executed for edited DB2 table data (either explicitly or implicitly when closing the edit session), each row that has been deleted, inserted or updated since the last time that data was saved (or since the start of the edit session) is processed by generating and executing an appropriate dynamic SQL statement to achieve the required change.

It is, therefore, important to identify which rows in the edit view are considered as having been inserted or updated (especially following a row replicate operation.) The record information flags display (controlled via the **RECINFO** option) will identify the inserted and updated rows, however, SELCOPYi 3.30 also introduces automatic colour highlighting for these rows.

The **COLOUR** option supports parameter **MODIFIED** to identify the colour highlighting that occurs for modified data records or DB2 table rows. For DB2 edit only, this option also applies to inserted (e.g. copied and replicated) table rows.

DB2 Table Edit Utility Popup Menu

In conjunction with some of the DB2 features already documented, the "DB2 Browse/Edit Utilities Menu" panel has been updated to include the following Field Option items:

Configure display of Var-Length and Nullable fields

Opens the new "DB2 Edit Settings" panel to manage display and update of variable length data columns and columns defined as supporting NULL values. This panel sets options applicable to these type of fields (i.e. **NULLCHAR**, **NULLIFBLANK**, **VENDCHAR**, **VSHOWEND**, and **VSTRIP**.)

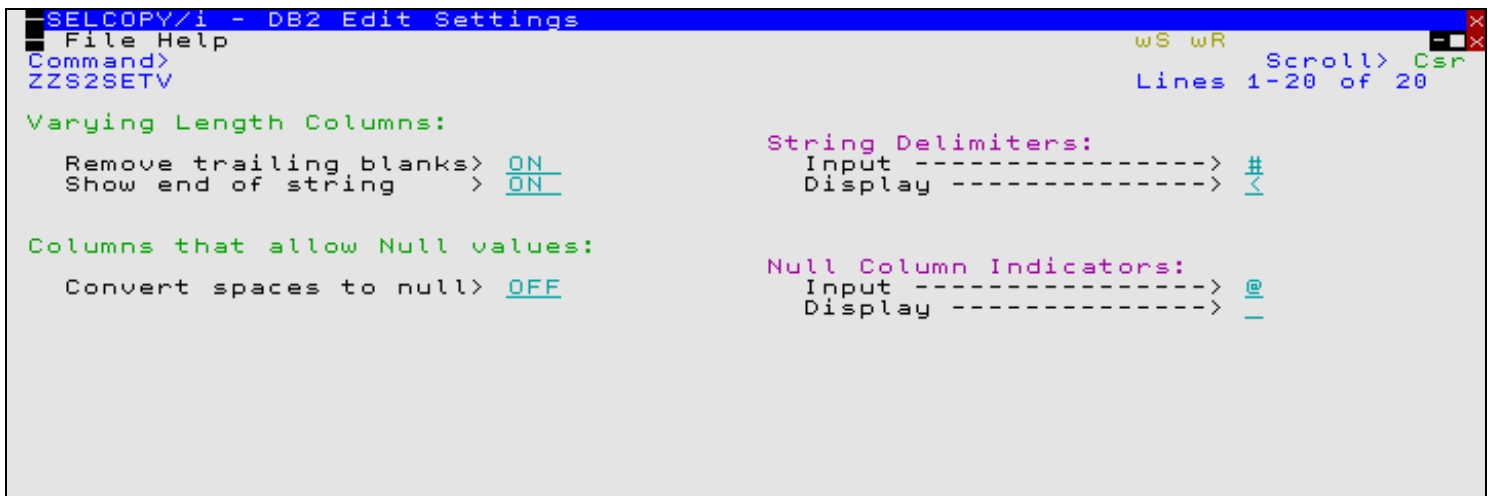


Figure 5. DB2: SELCOPYi DB2 Browse/Edit Utilities menu.

Delete focus field (*field_name*) value

Deletes the current value from the field at the cursor location.

This option is useful for columns containing very long values (i.e. that span multiple screens in multi-row view or multiple lines in single-row view) where you would otherwise have to use the <Erase-EOL> key many times to remove the current value.

Primary Commands

The following Data Editor primary commands have been updated or included in SELCOPYi 3.30.

BROWSE

Note that the following syntax does not include all keywords but serves to highlight use of new and updated parameters.

Syntax:

```
>>-- Browse ---+----- fileid -----| SDE Opts |--->
          |
          +-- DB2 ---+-----+-----| DB2 Opts |--+
                |         |
                +- (ssn) -+

>-----+-----+-----+-----+----->>>
        |               +-----+-----|
        |               v               |
        +-- INItcmd - ( --+ sd_command -+- ) -+

```

SDE Opts:

```

|+-----+
| USING | struct_name |
|+-----+-----+
| STRUCTure = |
|+-----+-----+
| HLASM copybook_name |
|+-----+-----+
| COBOL |
|+-----+-----+
| PL1 |
|+-----+-----+
| ADATA |
|+-----+-----+
| V |
| SYMNames ( DFSORT symbol dataset ) |
|+-----+-----+

> +-----+
| KEYRange - ( lowkey | ) |
| | | , - highkey |
|+-----+-----+
| FROM -- KEY -- string |
| | RBA ----- n ----+ | FOR -- n_recs --+-----+
| | | | | | | | | | RECORDS +-+
| | | | | | | | | |
| REcord +-+
|+-----+-----+

> +-----+
| FILTER filter_fileid | | |
| | | v |
| Filter Clause | SELECT select_syntax |
|+-----+-----+

```

DB2 Opts:

```

|+--+ table_name +--+-----| SQL Query Opts |-----+----->
|      |      |
|+--+ view_name  +--+ USING  +-----+--+ struct_name  +--+
|      |      |      |      |
|      |      |      |      |
|      |      |      |      |
|+--+ SQL ( select_statement ) -----+-----+
|
>+-----+-----+
|      |      |
|+----- SELECT (select_syntax) -----+

```

SQL Query Opts:

```

|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  +- WHERE (where_clause) +- ORDER +------+- (order_by_clause) +-
|      |               |      |               |
|      +--- ? ---+      |      +- BY +- |
|                          |
|                          +- SORT -----+
|
|                          +- SORTIndex +--- index_name --+-----+
|                              |
|                              +- Prime -----+

```

Existing primary command BROWSE has been enhanced to support the following:

- Three-part DB2 table and view names for **remote DB2 server** location.
- New **HLASM** keyword parameter for creation of a temporary **structure (SDO)** from an Assembler DSECT.
- New **SYMNAMES** keyword parameter for creation of a temporary **structure (SDO)** from a DFSORT SYMNAMES source.
- New **SELECT** keyword parameter to customise the sequence and number of columns displayed when the browse view is opened. SELECT for DB2 table browse does not correspond to a DB2 SQL *select-clause*. The executed SQL query used to fetch table rows selects **all** columns from the DB2 table/view and subsequently applies the SELECT clause to re-sequence and display the selected columns.

Unlike parameter **INITCMD**, which may also be used to execute a SELECT command, the SELECT parameter column selection is performed before any **profile** macro is executed.

- New **KEYRANGE** keyword parameter and *lowkey/highkey* arguments to select a range of KSDS records to be browsed. *lowkey* and *highkey* may reference a complete or partial key.
- New **INITCMD** keyword parameter to execute one or more Data Editor primary commands after the **profile** macro (default SDEPROF) is executed but before initial display of the browse view.
- New DB2 option **SQL** parameter used to supply any valid SQL query statement. This is an alternative to specification of a DB2 table or view name with optional SELECT, WHERE and ORDER BY clauses.
- Existing DB2 query option **WHERE** now supports question mark (?) argument to invoke the **DB2: Select table rows by column value** modal panel.
- New DB2 query option **SORTINDEX** to support row sequencing by columns defined by an index belonging to the DB2 table to be browsed.

CREATE STRUCTURE

Note that the following syntax does not include all keywords but serves to highlight use of new and updated parameters.

Syntax:

```
>>-- CREATE STRUCTure --- struct_name ----->
>+-----+-----+-----+-----+
|      TITLE short_title    +- DESCRIPTION long_description ---+
|
>+----- | XRef Definition | -----><
|         |                 |         |
+-----+ | CopyBook Definition | +-----+
|         |                 |         |
+-----+ | DB2 Definition   | +-----+
|         |                 |         |
+-----+ | Direct Definition | +-----+
```

XRef Definition:

```
|-----+----->
|      |               +-----+               |
|      v               |               |
| +- LIBrary -- (--- copybook_library +- ) -----+
|
| +-----+
| v       |
|>----- REcord -- ( -- | RecordType Definition | -- ) ---+-----
```

RecordType Definition:

```

|-- NAME --- record_type ----->
      + COBo1 +
      |       |
>-- SOURCE +-----+-----+-----+-----+-----+-----+-----+-----+-----+
      | + PL1 --+   +- copybook_library(member_name) --+
      | + HLAsm +
      |
      +-- ADAta -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
      |
      +-- SDO -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
      |
      +-- STRUcture (-- | Record Definition | --) -----+-----+-----+-----+-----+
      |
      |               +-----+-----+-----+-----+-----+-----+-----+-----+
      |               v                                     |
      +-- SYMNAMES   (-+- DFSORT SYMNAMES dataset -+-) -+

```



```

> +-----+-----+-----+-----+-----+-----+-----+-----+
| KEYRange - ( - lowkey +-----+-----+ ) -----+
|                                     |               |
|                                     +- , - highkey +-
|
+-- FROM +- KEY -- string +-+-----+-----+
|                                     |               |
|                                     +- RBA ----- n ----+ +- FOR -- n_recs --+-----+-----+
|                                     |               |
| +-----+-----+ n ----+                                     +- RECORDS +-
|                                     |               |
| +- REcord +-
|
> +-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |               |
| +- FILTER +-+----- filter_fileid ----+ | +-----+-----+
|                                     |               |
|                                     +-+ Filter Clause |---+ +-+ SELECT select_syntax +-+

```

DB2 Opts:

```

|-----| SQL Query Opts |-----+----->
|
|-- USING -----+----- struct_name --+
      |               |
      +- STRUCTure -+

> +-----+-----+
  |
  +---- SELECT (select_syntax) ----+

```

SQL Query Opts:

```

|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| +- WHERE (where-clause) -+ | +- ORDER +-----+ +- (orderby-clause) -+ | | | | | |
|           |               | |         |             |             |             |
|           +--- ? ---+    | |         +- BY -+      |             |
|                           | |         SORT -----+          |             |
|                           | |         +- SORTIndex +--+ index_name --+-----+ |
|                           | |                   +- Prime -----+          |

```

Existing primary command EDIT has been enhanced to support the following:

- Three-part DB2 table and view names for **remote DB2 server** location.
 - New **HLASM** keyword parameter for creation of a temporary **structure (SDO)** from an Assembler DSECT.
 - New **SYMNAMES** keyword parameter for creation of a temporary **structure (SDO)** from a DFSORT SYMNAMES source.
 - New **SELECT** keyword parameter to customise the sequence and number of columns displayed when the browse view is opened. SELECT for DB2 table edit does not correspond to a DB2 SQL *select-clause*. The executed SQL query used to fetch table rows selects **all** columns from the DB2 table/view and subsequently applies the SELECT clause to re-sequence and display the selected columns.
- Unlike parameter **INITCMD**, which may also be used to execute a SELECT command, the SELECT parameter column selection is performed before any **profile** macro is executed.
- New **KEYRANGE** keyword parameter and *lowkey/highkey* arguments to select a range of KSDS records to be edited. *lowkey* and *highkey* may reference a complete or partial key.
 - New **INITCMD** keyword parameter to execute one or more Data Editor primary commands after the **profile** macro (default SDEPROF) is executed but before initial display of the browse view.
 - Existing DB2 query option **WHERE** now supports question mark (?) argument to invoke the **DB2: Select table rows by column value** modal panel.
 - New DB2 query option **SORTINDEX** to support row sequencing by columns defined by an index belonging to the DB2 table to be browsed.

ERROR

Syntax:

```
>>--+- Error -----+----->>
      +- SQLException --+
```

Description:

New in SELCOPYi 3.30, ERROR displays the SQL Error panel describing the non-zero SQLCODE returned from the last attempt to save the focus row.

This command is only applicable to DB2 table edit. It has the same effect as the **E - Expand SQL Error** line command entered in the prefix area of the focus row.

See **Related Table Edit** for examples of SQL Error display.

INFO**Syntax:**

```
>>-- INfOrMation -----+-----+-----><
      |
      +-----+-----+-----+-----+
      |         fileid -----+
      +-- DB2  --+-----+-----+-----+
                  |         table_name -----+
                  +- (ssn) -+
                  |
```

Description:

New in SELCOPYi 3.30, INFORMATION has the same operation as the SELCOPYi primary command **DSINFORMATION** except that it applies only to Data edit/browse window views. i.e. Dataset information panel/report is displayed for the specified *fileid* or DB2 *table_name* (default is dataset/DB2 table in the focus Data Editor view.)

POP**Syntax:**

```
>>-- POP -----><
```

Description:

New in SELCOPYi 3.30, POP is used in SDE REXX macros to reset user configurable SDE SET options that have been saved by a previously executed **PUSH** operation. the reset includes all options that may be set at the Global, File and View levels.

PUSH**Syntax:**

```
>>-- PUSH -----><
```

Description:

New in SELCOPYi 3.30, PUSH is used in SDE REXX macros to save the current values of user configurable SDE SET options. All options that may be set at the Global, File and View levels are saved. Saved option values may be subsequently restored using the **POP** operation.

REDIT**Syntax:**

```
>>-- REdit ---+-----+-----+-----><
      |
      +--- PArent -----+--- referential_constraint ---+
      |
      +-+ DEpendent +-+
      |
      +- CHILd -----+
      |
```

Description:

New in SELCOPYi 3.30, the REDIT primary command applies only to browse and edit of DB2 tables and is used to list, edit or browse tables that are related to the DB2 table in the focus Data Editor view. See also the **RE** line command.

Relationships between tables are defined by referential constraints. A referential constraint specifies a set of columns as a foreign key in the dependent (or child) table which is a unique or primary key in the parent table.

REDIT opens an **Edit/Browse Related Table** view which is thereafter linked to the focus DB2 edit/browse view from which it has originated. Closing the originating DB2 edit/browse view will close all Edit/Browse Related Table views that are linked to it.

The specified (or implied) referential constraint determines which of the related tables are to be edited or browsed by the REDIT operation. Furthermore, the **focus row** of the originating DB2 edit/browse view identifies the foreign or parent key value for which related rows are displayed. i.e. Only rows related to this parent/foreign key will be loaded into the related table view.

Note that the related table may be a parent table or a child table of the focus DB2 table.

REDIT with no parameters:

When the REDIT command is issued with no parameters the behaviour depends on the context.

If the focus row has had an **SQL relational error** as a result of the last SAVE operation then the relationship involved in the error will be used automatically to supply the REDIT parameters and the related table session will be created.

Otherwise the "List Related Tables" panel is displayed listing all the parent and dependent relationships for the current table. From this list a relationship can be chosen (with the RE list prefix command) to supply the parameters to REDIT.

SORT

Syntax:

```

      +- .ZFIRST +- +- .ZLAST +- +-----+ , +-----+
      |         | |         | |         | |         |
>>-- SORT --+--+-----+--+-----+--+-- | Sort Field Spec | --+--+--><
      |         | |         | |         | |         |
      | +- .name1 --+ +- .name2 --+
      |         | |         | |         | |         |
      +-- KEY -----+
      (1)

```

Sort Field Spec:

```

      +- Ascending --+
      |               |
|-+- field_col -----+--+-----+--+
      |               | |         | |         |
      |               | +- CCharacter --+ +- Descending --+
      |               | |         | |         |
      +- field_pos -- field_length -----+--+-----+--+
      |               | |         | |         |
      |               | +- Fixed -----+
      |               | |         | |         |
      |               | +- PD -----+
      |               | +- DECimal -----+
      |               | +- PACKed -----+
      |               | |         | |         |
      |               | +- ZD -----+
      |               | +- Zoned -----+
      |               | |         | |         |
      |               | +- HFP -----+
      |               | |         | |         |
      |               | +- BFP -----+
      |               | |         | |         |
      |               | +- DFP -----+

```

Notes:

1. Only for a KSDS data set that has never been loaded.

Description:

In SELCOPYi 3.30, the SORT command has been enhanced to sort in-storage data records or DB2 table rows within the current edit view.

Sort is supported only if **all** edited records or table rows are loaded in available storage, otherwise error ZZSD643E is returned.

Sort fields are identified by a record position, length and data type. Alternatively, if the edited data is mapped by a single record type (includes DB2 table rows), a sort field may be identified by a column field reference. If the edited data is mapped by more than one record type, sort fields must be specified using position, length and data type.

Multiple sort fields may be specified. The order in which sort fields are specified define the hierarchy in which the record or DB2 table row data will be sorted.

When sorting variable length records, it is possible for a specified sort field to reference positions that fall outside the length of a record, but not outside the defined maximum record length. In this case, the sort field is padded with nulls (x'00').

If *field_pos* syntax is used to identify a sort field that exists entirely within the fixed portion of **formatted** records, then the sort field references positions in the unexpanded records. If the sort field overlaps variable areas of the formatted records (i.e. in areas mapped by or following fields of variable length data types) then, the following will occur:

- For edited data mapped by a single record type, the sort field will reference positions in **expanded record** data.
- For edited data mapped by multiple record types, the sort field can only reference positions in the unexpanded record and so the edited record data must be displayed in **character format**. If not, error ZZSD666E will be returned.

SORT KEY was supported in previous releases of SELCOPYi and is applicable only when records are added to an initially empty VSAM KSDS data set.

SORTINDEX

Syntax:

```

      +- .ZFIRST -+ +- .ZLAST -+
>>-- SORTIndex --+-----+-----+-----+----->
      |          |          |          |
      +- .name1 --+ +- .name2 -+       +- index_name -+
                                   +- Prime -----+

```

Description:

New in SELCOPYi 3.30, SORTINDEX is supported for DB2 table edit only and may be used to sort in-storage table rows by key columns/expressions identified by a DB2 Index previously defined for the table.

If neither **PRIME** nor *index_name* are specified, a "DB2 Select Table Index" panel is displayed, allowing selection of one of the indexes defined for the table.

Line Commands

The following Data Editor line commands have been included in SELCOPYi 3.30.

E - Expand SQL Error

New in SELCOPY: 3.30, the **E** line command may be entered against edited DB2 table rows for which an SQL code has been flagged following a SAVE operation. For each line in which E is entered, the "SQL Error" panel is displayed containing the full SQL description of the SQL code.

This command is only applicable to DB2 table edit and has the same functionality as the **ERROR** primary command.

See [Related Table Edit](#) for examples of SQL Error display.

RE - Related DB2 Table Edit

New in SELCOPYi 3.30, the **RE** line command is applicable only to browse or edit of DB2 table data and is equivalent to executing the **REDIT** primary command with no parameters using the row on which RE is entered as the focus row.

SET/QUERY/EXTRACT Options

The following Data Editor options have been included in SELCOPYi 3.30.

AUTOSTRUCTURE

Syntax:

```
>>+-----+----- AUTOSTRUCTure -----+-- ON -----+-----><
    |         |         |         |         |         |
    +- SET -+         |         |         |         |
                                +- SAVE-----+
                                |         |
                                +- APPLY --+
                                |         |
                                +- OFF ----+

>>--- Query ----- AUTOSTRUCTure -----><

>>--- EXTRACT -- / AUTOSTRUCTure / -----><
```

Description:

AUTOSTRUCTURE controls the level at which the **structure to dataset association** facility is implemented. SAVE will automatically save associations performed during a data edit or browse operation, APPLY will automatically apply saved structure associations to data files edited or browsed without reference to a structure, and ON will do both.

COLOUR

Note that the following syntax does not include all keywords but serves to highlight use of new and updated parameters.

Syntax:

```
>>+-----+-----+-----+-----+-----+-----+-----+-----+-----+
    |         |         |         |         |         |         |         |         |
    +- SET -+         |         |         |         |         |         |         |
                                +- CMDline -----+ +- Red -----+ +- BLInk ---+
                                |         |         |         |         |         |         |
                                +- PROMpt -----+ +- Pink -----+ +- REVerse +
                                |         |         |         |         |         |         |
                                +- Green -----+ +- Uscore --+
                                |         |         |         |         |         |         |
                                +- Turquoise -+
                                |         |         |         |         |         |         |
                                +- Yellow ----+
                                |         |         |         |         |         |         |
                                +- White ----+
                                |         |         |         |         |         |         |
                                +- Default ----+

>>--- Query -----+----- COLOr ---+-----><
                        |         |
                        +- COLOur -+

>>--- EXTRACT --- / +- COLOr ---+ / -----><
                        |         |
                        +- COLOur -+
```

Description:

In SELCOPYi 3.30, the COLOUR (COLOR) SET option and QUERY/EXTRACT operations have been enhanced to support additional parameters CMDLINE and PROMPT as supported by the Text Editor.

CMDLINE controls the colour of text entered at the command prompt (default green). PROMPT controls the colour of the command prompt itself (default blue).

either of these commands are used, the SET keyword is required.

SET DFPSCALE takes effect at the File level.

EDITPRIMEKEY

Syntax:

```
>>-+-----+---+ EDITPrimekey -+---+--- ON ---+-----><
|         |         |         |         |         |
+- SET -+ -+ EPK -----+ +--- OFF ---+

>>--- Query ---+ EDITPrimekey -+-----><
|         |         |
+- EPK -----+

>>--- EXtract -+ /EDITPrimekey/ +-----><
|         |         |
+ /EPK/ -----+
```

Description:

New in SELCOPYi 3.30, EDITPRIMEKEY applies only to DB2 edit of tables with primary keys.

When EDITPRIMEKEY is ON, the columns of the primary key are enterable, otherwise the columns of the primary key are protected.

SET EDITPRIMEKEY takes effect at the file level. The value is not saved across sessions and may be specified on the EDIT command with the presence or absence of the EDITPRIMEKEY parameter.

NULLCHAR

Syntax:

```
>>-+-----+---+ @ -----+ +--- _ -----+
|         |         |         |         |         |
+- SET -----+ input_indicator -+---+ output_indicator +---><

>>--- Query ----- NULLChar -----><

>>--- EXtract -- /NULLChar/ -----><
```

Description:

New in SELCOPYi 3.30, NULLCHAR applies only to DB2 edit or browse of table columns which are defined as supporting NULL values.

NULLCHAR specifies two special characters. The first identifies the character to be entered when a column's value is to be set to NULL (*input_indicator*). The second identifies the character used by the Data Editor to indicate that a column's value is set to NULL (*output_indicator*).

SET NULLCHAR takes effect at the file level and its setting is saved if SAVEOPTIONS ON is in effect.

NULLIFBLANK

Syntax:

```
>>-+-----+---+ NULLIfblank -+--- ON ---+-----><
|         |         |         |         |         |
+- SET -----+ +--- OFF ---+

>>--- Query ----- NULLIfblank -----><

>>--- EXtract -- /NULLIfblank/ -----><
```


VERSION

Syntax:

```
>>--- Query ----- VERSION -----><

>>--- EXTRACT --- /VERSION/ -----><
```

Description:

QUERY and EXTRACT VERSION report version information for the Data Editor application within the executing SELCOPYi product.

VSHOWEND

Syntax:

```
>>-+-----+ VSHOWend +- ON ---+-----><
  |         |         |         |
+- SET -----+         +- OFF --+

>>--- Query ----- VSHOWend -----><

>>--- EXTRACT -- /VSHOWend/ -----><
```

Description:

New in SELCOPYi 3.30, VSHOWEND applies only to DB2 edit or browse of table columns which are of variable length.

If VSHOWEND is ON, the end of variable length column data is marked with the **VENDCHAR** output indicator character. If VSHOWEND is OFF, the end of the variable length column data is not marked.

SET VSHOWEND takes effect at the file level and its setting is saved if SAVEOPTIONS ON is in effect.

VSTRIP

Syntax:

```
>>-+-----+ VSTRip +- ON ---+-----><
  |         |         |         |
+- SET -----+         +- OFF --+

>>--- Query ----- VSTRip -----><

>>--- EXTRACT -- /VSTRip/ -----><
```

Description:

New in SELCOPYi 3.30, VSTRIP applies only to DB2 edit of table columns which are of variable length.

If VSTRIP is ON, any trailing blanks in a variable length column value are stripped and the length of the column data adjusted accordingly. If VSTRIP is OFF, there is no special treatment of trailing blanks.

SET VSTRIP takes effect at the file level and its setting is saved if SAVEOPTIONS ON is in effect.

Functions

The following Data Editor function has been included in SELCOPYi 3.30 for use in expressions.

SQLERROR()

Syntax:

```
>>-- SQLERROR() -----><
```

Description:

New in SELCOPYi 3.30, SQLERROR() or BIFSQLERROR() has no arguments and applies to DB2 table rows only. If the focus DB2 table row experienced an SQL error on SAVE, this function returns the SQL error code, otherwise zero is returned.

Because SQLERROR() has no arguments, it may be expressed simply as SQLERROR. Since SDE expressions treat any non-zero value as a true result for Boolean operations, SQLERROR may be used for a *line_flag* search on the LOCATE and WHERE (ALL) primary commands. e.g. LOCATE SQLERROR will locate the next occurrence of a row for which an SQL code is flagged.

Panels

The following Data Editor Panels have been updated or introduced in SELCOPYi 3.30.

Create Structure from Copybook(s)

The "Create STRUCTURE from COBOL/PL1 copybook(s)" utility sub-panel view for Record-type manipulation (ZZSGSDOR) has been updated to support a member source language (Lang) of ASM. This allows creation of a structure (SDO) with one or more record type mappings derived from an Assembler DSECT.

Structured Data Browse/Edit

The "Structured Data Browse/Edit" utility panel (ZZSGSDE0) has been updated to support a Structure/Copybook overlay type of ASM to allow creation of a temporary structure (SDO) from an Assembler DSECT member.

Select Display Fields

New to SELCOPYi 3.30, the "Select Display Fields for a Record Type" panel (ZZSGSELD) is opened on executing either of the following:

1. Primary command **SELECT** with no parameters. In this case the default record type is used.
2. Line (prefix) command **SEL** in the prefix area of a record/table row of the required record type.

This panel is used to update column width, column selection and column sequence for a particular record type data mapping in the focus SDE view. The changes made in this panel may be temporary or saved as a permanent update to the structure (SDO).

Note that the dialog is a *modal* panel. While it is open no other window can be made the focus. The dialog must be terminated (using **END** or **CANCEL**) before any other window can become active.

If opened for a record type which has array (OCCURS DEPENDING) entries, then the individual array element entries, represented as field names suffixed with a parenthesised element number subscript, are excluded by default. These entries may be displayed and excluded using primary commands +ELEMENT (+E) and -ELEMENT (-E) respectively.

SELCOPY/i - Select Display Fields for a Record Type

File Edit Actions Options Utilities Window SwapList Help wS wR

Command> ZZSGSELD

Structure Name: USER123.SELCTRN.SDO(ZZST2)

Record Type: TRACK

Perm/Temp: PERM

Show unselected fields at the end: N Y/N

Select record type fields 35 Rows

S	H	Seq	Width	Name	Type	Start	Length	Picture	Data type	Precision
				TRACK	AN	1	268		STRUCTURE	0 001
				RT	AN	1	1	X(001)	CHARACTER	0 002
				PERSISTENT-ID	AN	2	16	X(016)	CHARACTER	0 003
				TRACK-NUM	ZD	18	3	9(003)	ZONED	3 004
				TRACK-ID	ZD	21	4	9(004)	ZONED	4 005
				NAME	AN	25	120	X(120)	CHARACTER	0 006
				TOTAL-TIME	FB	145	4	9(007)	FIXED	0 007
				FILE-SIZE	FB	149	4	9(009)	FIXED	0 008
				BIT-RATE	FB	153	2	9(004)	FIXED	5 009
				SAMPLE-RATE	PD	155	3	9(005)	DECIMAL	5 010
				YEAR	ZD	158	4	9(004)	ZONED	4 011
				NORMALIZATION	PD	162	3	9(005)	DECIMAL	5 012
				DISC-NUMBER	ZD	165	3	9(003)	ZONED	3 013
				ALBUM-ARTIST	AN	168	41	X(041)	CHARACTER	0 014

Figure 6. SDE Select Columns Panel.

DB2 Row Selection

New to SELCOPYi 3.30, the "DB2 Select table rows by column value" panel (ZZS2DRSD) is used to generate a basic SQL WHERE clause for use in a DB2 table edit or browse operation, or in creating a SELCOPYi structure (SDO).

See topic on [Row Selection](#) in this document for details.

DB2 Edit Object / DB2 Browse Object

The "DB2 Browse Object" panel (ZZS2BROW) and "DB2 Edit Object" panel (ZZS2EDIT) have been updated so that they include a Location input field allowing specification of a DB2 table or view belonging to a remote DB2 server.

See topic on [Remote Location Table Edit/Browse](#) in this document for details.

SELCOPY/i - DB2(CBLA): Browse Object

File Run Command Help wS wR

Command> ZZS2BROW

DB2 Object:

Location> DALLAS9 (optional)

Owner> CBL (optional)

Name> APIFUNC

Row Selection Options:

From> Start row within the result set.

For> Number of Rows to be browsed.

Select> FUNCMOD, FUNCNAME, FUNCTITLE

Where> FUNCMOD#REF=0

Order By> FUNCMOD

Initial Display Format:

/ Table _ Single

Profile Macro:

/ Use Default.

_ Use Specified Macro. Macro Name> SDEPROF

_ Do not use a Profile Macro.

Figure 7. DB2 Browse Object Panel.

DB2 Select Table Index

New to SELCOPYi 3.30, the "DB2 Select Table Index" panel (ZZS2SINX) is displayed when primary command **SORTINDEX** is executed with no parameter arguments.

The panel displays a list of indexes defined for the DB2 table edit/browse view. If more than one key column is defined for an index, then repeated list rows exist for that index with one row for each index key column. Selection of any of these rows will select the equivalent index and sort the rows by that index's key columns.

SELCOPY/i - DB2: Select Table Index

View Refresh Back Forward FDB Text Help

Command> ZZS2SINX

Select Index for table:

---Index---	-Cols-	-Seq--	--Column---	Order	Unique	Data Type	Schema
APIFUNC#IX0	2	1	APILIB	A	P		CBL
		2	FUNCREF	A	P		CBL
APIFUNC#IX1	2	1	APILIB	A	R		CBL
		2	FUNCNAME	A	R		CBL
APIFUNC#IX2	2	1	FUNCMOD	A	C		CBL
		2	FUNCMOD#REF	A	C		CBL
APIFUNC#IX3	1	1	FUNCNAME	A	D		CBL

Line 1 of 7 | Col 1 of 68 | Views 1 | select * sort Index,Cols

Figure 8. DB2 Select Table Index Panel.

SELCOPYi Utilities

SELCOPYi supports a number of general utilities (e.g. File Erase), utilities that require SDE Data Editor operations (e.g. File Search/Update/Copy/Remap) and DB2 based utilities (e.g. DB2 Create Table).

SELCOPYi 3.30 includes a number of new utilities and enhancements to existing utilities.

CSV Generation

SELCOPYi 3.30 introduces the CSV generation utility to generate Comma Separated Variable (CSV) output from formatted (structured) data. Data may be any data that is formatted by a structure (SDO) either explicitly, via a nominated SDO dataset, or implicitly via a nominated COBOL, PL1, Assembler copybook/DSECT or DB2 result table.

Fields, defined by a structure record/table row mapping, identify columns of data that define the content and sequence order of variables in the CSV output. The CSV output may be restricted to a subset of the input records (or table rows) and fields. Furthermore, fields may be resequenced before the CSV output is generated.

Where the input structure contains multiple record-type (record mapping) definitions, the user is prompted to select one of the record-types. All records associated with other record-type mappings are not processed by the utility.

If executed from a Data Editor browse or edit view, CSV generation may be performed on the currently displayed data, operating only on non-excluded lines and using only columns selected for display.

CSVGEN Primary Command

Syntax:

```
>>- CSVgen +-----+><
|
|+- | Input dataset specification | +-+ | Common Options | -+
|
|   +- .ZFIRST -- .ZLAST +-+
|   |
| (1) |
|+- * +-----+
|
|   +- .name1 +-+
|   |
|       +- .name2 -+
|
```

Input dataset specification:

```
| - INDSN ( input_dataset_specification ) -----|
```

Common Options:

```
|----->
|+-| Output dataset specification |-----+
>----->
|+-| Character conversion option |-----+
>----->
|
|+- SEPARATOR , -----+ +- QUOTE CHARACTER +-+
|
|+- SEPARATOR sep_char +-+ +- QUOTE ALL -----+
|
|       +- QUOTE REQUIRED --+
|
|
|
|+- HEADER +-+ +- BROWSE +-+
|
|+- RECORD +-+ record_type +-+ +- NOHEADER +-+ +- EDIT +-+
|   +- Type +-+
|
|       +- NOVIEW +-+
```


Output dataset specification:

```

      +- EOL NL -----+
      |                   |
>- OUTDSN ( output_dataset ) +-+-----+-----+-----+-----+-----+-----+-----+
      |                   |                   |                   |                   |
      | +- MODify +-+    +-+ NL -----+    |                   |
      | +- APPend +-+    +-+ EOL -----+    |                   |
      |                   |                   |                   |
      |                   |                   | +- CR -----+
      |                   |                   | +- LF -----+
      |                   |                   | +- CRLF -----+
      |                   |                   |

```

Character conversion option:

```

      +- terminal_ccsid +-+
      |                   |
>-+- CONVert +-+-----+-----+-----+-----+-----+-----+-----+
      |                   |                   |                   |                   |
      |                   | +- TO +-+        +-+ FRom +-+ +-+ from-ccsid -----+
      |                   |                   |                   |                   |
      |                   |                   |                   |                   |
      | +- ASCII -----+
      |                   |
      | +- UNICODE -----+

```

Notes:

1. Asterisk (*) is optional if any other parameter is specified.

Description:

New to SELCOPYi 3.30, CSVGEN is used to produce an exportable copy of structured data as Comma Separated Variable text. The structured data may belong to a specific structured data set or the current SDE view.

If asterisk (*), or any other parameter other than INDSN, is specified, then data in the current SDE edit or browse view will be used as input for CSV generation. If executed with no parameters from a Data Editor browse or edit view, the "SDE CSV Generation" panel is opened to process data from the focus view, otherwise the general "CSV Generation" utility panel is opened.

The CSVGEN command may be executed in the foreground or via SDEIN input to program SDEAMAIN for batch processing.

During foreground execution a progress window is displayed showing input and output record counts, updated every second, which allows the user to interrupt processing before completion using the attention key.

CSV Generation Panels

The CSV Generation utility supports two panel formats as follow:

CSV Generation Utility Panel

The **CSV Generation** utility panel may be displayed by selecting option 17 from the Utilities menu (=8.17) or by entering the primary command CSVGEN (or simply CSV) with no parameters from any window that is not a Data Editor view.

The **SELECT** primary command (assigned to F5 by default) may be issued in order to restrict and re-sequence the fields defined by the selected record-type mapping.

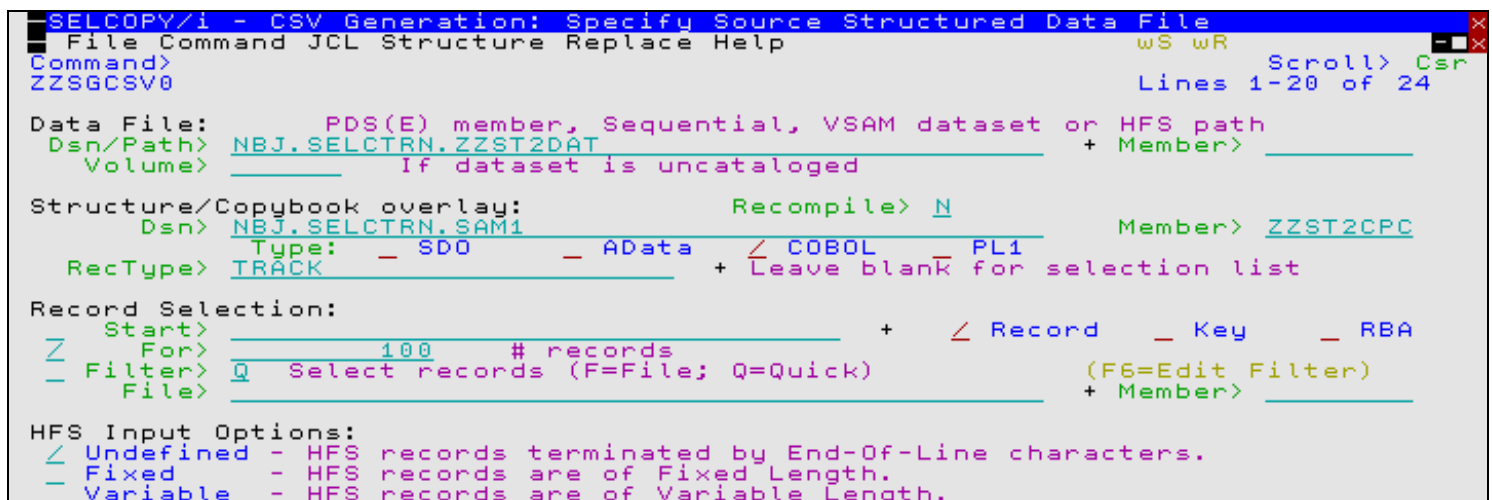


Figure 9. CSV Generation Utility Panel.

SDE CSV Generation Panel

The **SDE CSV Generation** panel may be displayed by entering the primary command CSVGEN (or simply CSV) with no parameters from any Data Editor view.

The CSV generation will process only fields in the sequence they are displayed in the browse/edit view, as customised by the **SELECT** primary command (or "Select Display Fields" panel). Furthermore, it will bypass all excluded records, records not within a selected range of lines (defined by line labels) and records not associated with the selected record-type. Note that, when this panel is opened, the setting of the Record Type panel input field is primed to be the record type associated with the focus record when CSVGEN was issued.

The CSVGEN process will be executed in the foreground.

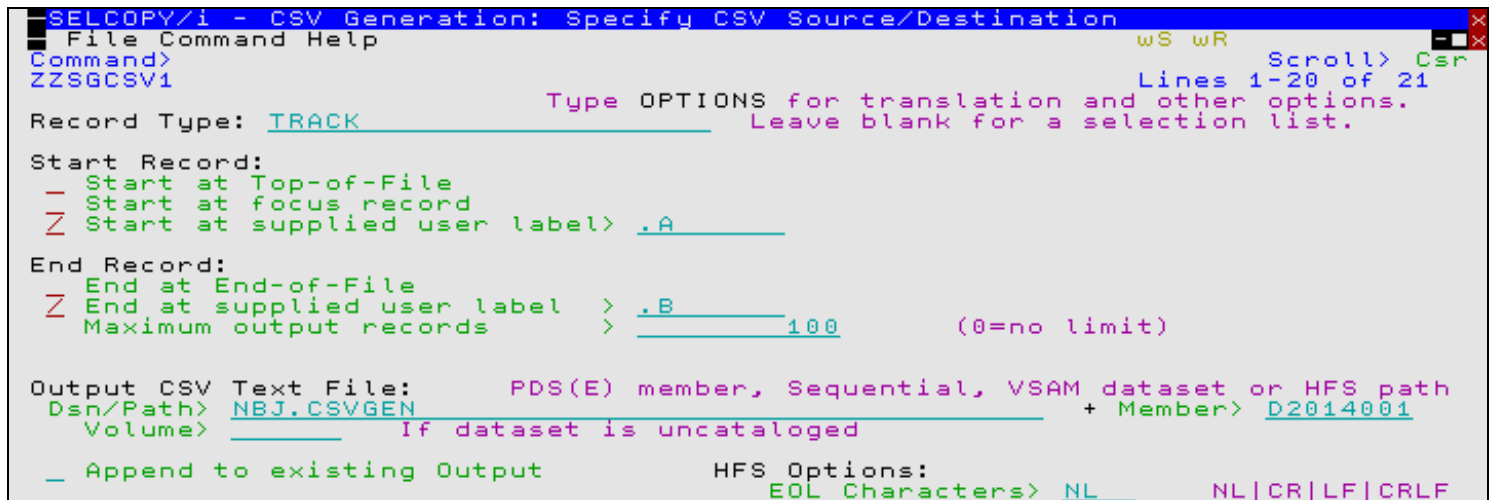


Figure 10. SDE CSV Generation Panel.

Data Set Information (DSI)

Detailed information on all types of file objects may be displayed using the Data Set Information (DSI) utility. A series of panel views or an HTML report output are generated to display information for a selected file. Information panels have different contents depending on the type and organisation of the file object.

The Data Set Information (DSI) utility has been enhanced in SELCOPYi 3.30 to support the following.

DSNTYPE=LARGE Datasets

Physical sequential (DSORG=PS) datasets may be allocated as DSNTYPE=LARGE to support a number of tracks per volume greater than 65,535. Information panels for this type of data set now display "PS LARGE" in the Organisation field.

DB2 Table Information

DB2 table information output is supported in HTML report format and provides a useful reference for anyone working with DB2 databases.

The information report for a DB2 table comprises general table information, detailed column attributes, indexes defined on the table, parent tables and dependent (child) tables. Cross links exist throughout the report to link to associate information. e.g. Links to named column information from index key column name references.

DB2 Object Creation Panels

Support for DB2 related functionality continues to grow and, in SELCOPYi 3.30, this includes support for sequences of panel views that intelligently assist with creation of DB2 indexes and clone tables, and general enhancements to existing DB2 object creation panels.

Object generation panels culminate with generation of valid SQL statements that may be executed immediately, saved to a data set for execution with EXEC SQL or inserted into a generated JCL job stream.

Create Index

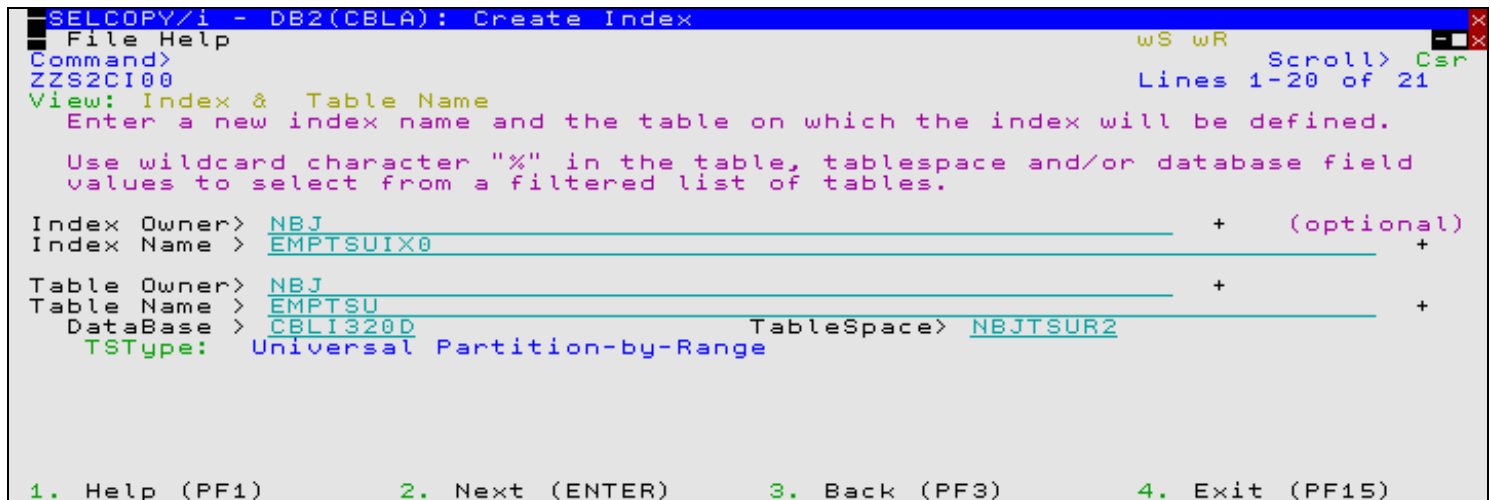
The DB2 **Create Index** series of panel views (ZZS2CI00) generate an SQL CREATE INDEX statement which may be used to create a new DB2 index, XML index or Auxiliary index in the current DB2 subsystem.

The DB2 Create Index panel may be started by selecting Index (option 8) in the SELCOPYi DB2 Create Objects option menu (DB2 5.8) or following execution of prefix command "X" against an entry in a DB2 Tables list.

By default, field entries are populated with arguments and options that were entered the last time the panels were used.

The panel views are displayed in a sequence based on choices made within each panel view and also on the type of table space for which the index will be created. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL CREATE INDEX syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceeding forwards again.



```

SELCOPY/i - DB2(CBLA): Create Index
File Help
Command>
ZZS2CI00
View: Index & Table Name
Enter a new index name and the table on which the index will be defined.
Use wildcard character "%" in the table, tablespace and/or database field
values to select from a filtered list of tables.

Index Owner> NBJ
Index Name > EMPTSUIX0
Table Owner> NBJ
Table Name > EMPTSU
DataBase > CBLI320D
TableSpace> NBJTSUR2
TSType: Universal Partition-by-Range

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

Figure 11. DB2: Create Index - Index & Table Name.

Create Clone Table

The "DB2 Create Clone Table" sequence of panel views (ZZS2CCT0) generate an SQL ALTER TABLE ADD CLONE statement to create a clone table for the selected table within the current DB2 subsystem.

The DB2 Create Clone Table panel views may be started by selecting Clone Table (option 16) in the SELCOPYi DB2 Create Objects option menu (DB2 5.16).

The panel views are displayed in a sequence based on choices made within each panel view. Field values and options may be selected in the focus panel before progressing to the next panel view in the sequence (NEXT). Progressing forwards from the last panel view in the sequence will generate the SQL ALTER TABLE ADD CLONE syntax.

Options and field entries specified in panel views that have been visited may be changed simply by navigating backwards along the view sequence (BACK) before proceeding forwards again.

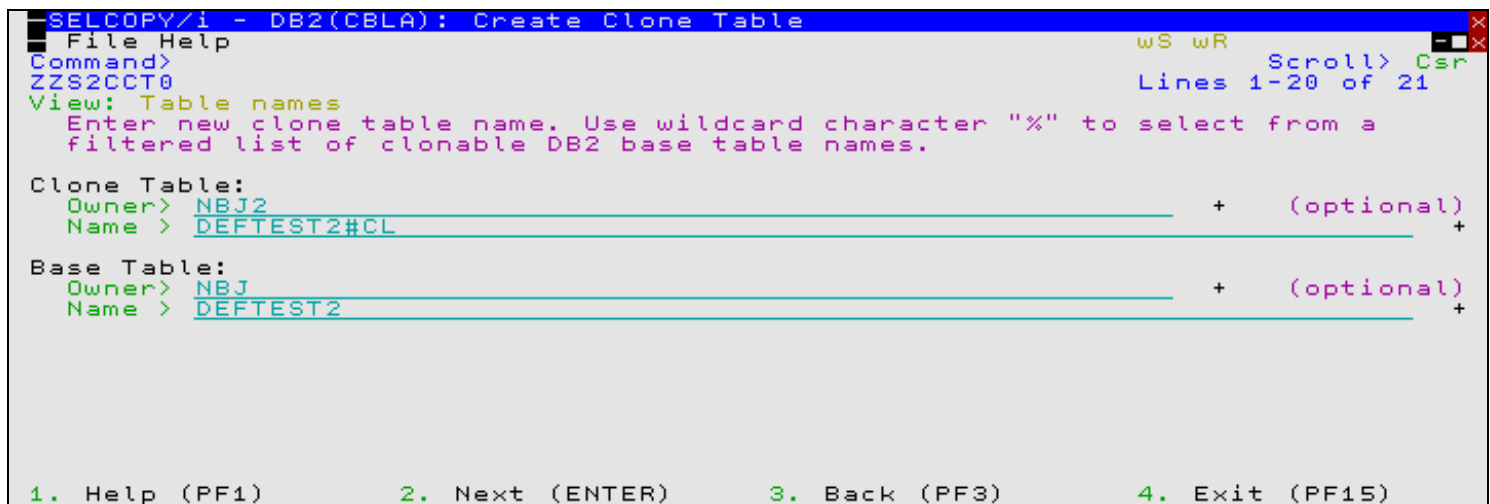


Figure 12. DB2: Create Clone Table.

General Enhancements

Existing DB2 object creation panels have been enhanced as follow:

1. Include a standard SQL Generation panel view as the last view in the sequence of panels. Values entered in the "SQL Output File" attribute fields included in this view are common to all object creation panels. This is so generated SQL statements may be appended to SQL statements already generated by other object creation panels. All statements may then be executed by a single execution of the EXECSQL utility.
2. Where relevant, include support for object creation modelled on an existing object of the same type. Although the object to be created will exist on the local DB2 server, the model object may exist on a remote DB2 server as identified by a Location input field.
3. Additional validation of the new object name has been introduced and is performed before progressing to subsequent panel views in the sequence. If the name entered for the new object would cause a name conflict with an object that already exists at the local server, then error ZZSP116E is returned which prevents progress to the next panel view.

In addition to changes to object creation panels, a Training Material item has been added allowing any user with DB2 access to create personal DB2 test tables and data in a named database. These test tables correspond to tables used in tutorials found in the SELCOPY Product Suite 3.30, "*SELCOPYi Training Manual*" publication.

DB2 SQL Execution (EXECSQL)

In previous versions of SELCOPYi, DB2 SQL execution could only be performed via the SQL or DSQL (DB2 SQL) utilities. Both utilities support input of an SQL statement via a panel input field and display SQL output in the display area of the panel. The difference between the two utilities is the DB2 PLAN used.

The SQL utility uses the same DB2 plan used by the SELCOPY batch component for DB2 SQL calls. This has the benefit of allowing a SELCOPY programmer to analyse the effect of an SQL statement with SELCOPYi (e.g. as part of the SELCOPY Debug utility) before being issued by a SELCOPY batch job. The DSQL utility uses the same DB2 plan used by SELCOPYi for all its native DB2 operations, i.e. (Table Edit, Object Management, etc.)

SELCOPYi 3.30 introduces an additional DB2 SQL Execution utility (EXECSQL) which, like DSQL, uses the SELCOPYi DB2 plan, but has the advantage that SQL input and output may be directed to and from specified datasets. Unlike DSQL, it may also be executed using the SDEAMAIN (SELCOPYi batch) utility.

EXECSQL Primary Command

Syntax:

```
>>- EXECSQL  ---+-----+-----+-----+-----+----->
                |         |         |         |
                +- SSN ssn_name -+ +- LIMIT n_rows -+

>+-----+-----+-----+-----+-----+----->
    |         |         |         |         |         |
    +- OUTDSN outfile -+

```

```
+-- NOFORMat -- | Unformatted Options | +--+
>+-----+-----+-----+-----+----->
|      FORMat -----+-----+-----+-----+
|                      | OUTSDO struct_name -----+
|                               | INDSN infile ----+
|                               | 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. If EXECSQL is executed with no parameters, the EXECSQL utility panel is opened.

EXECSQL has all the features of the IBM SPUFI utility but with additional support for the following:

- SQL source input file and report output file may be a sequential data set or a PDS or PDSE library member.
- 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 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.

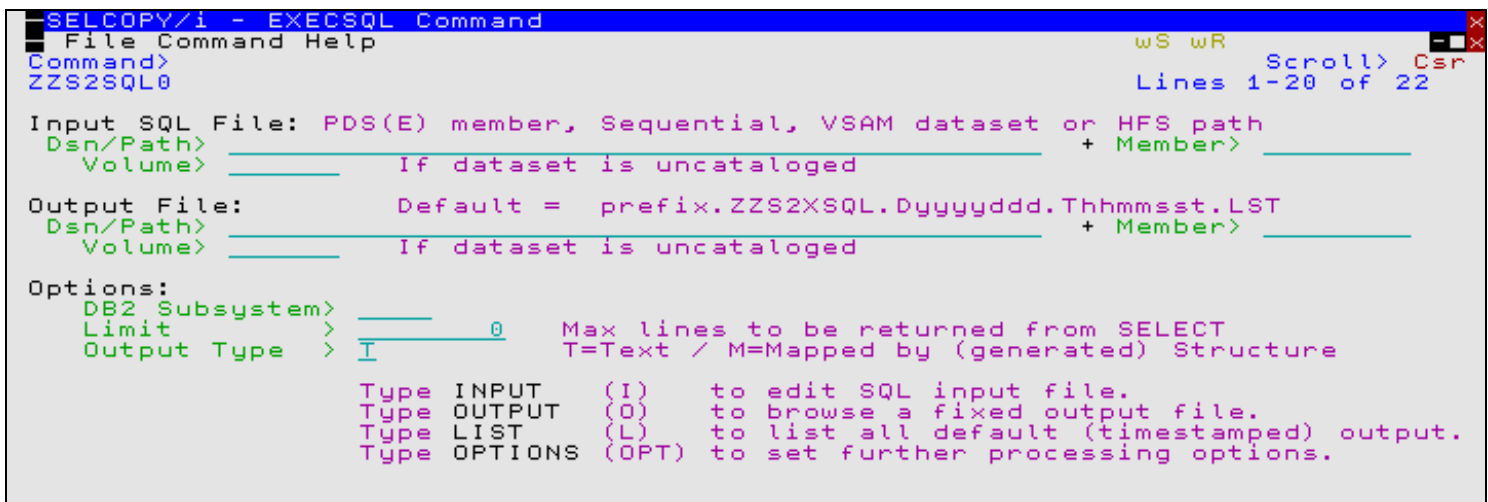
If EXECSQL is executed with no parameters, SQL is sourced from text in the focus text or data edit window.

When run interactively, the output report file is browsed automatically on completion.

EXECSQL Panels

The Execute SQL statements panel (ZZS2SQL0) is opened on selection of option 1 in the Execute SQL options menu off the DB2 primary options panel (DB2 2.1) or on execution of the EXECSQL command.

This panel provides facility to execute SQL statements from a data set or library member in a specified DB2 subsystem and direct the SQL output to a data set as either plain text or formatted output. The user must have the required level of authority in order to successfully execute an SQL statement.



```

SELCOPY/i - EXECSQL Command
File Command Help
Command>
ZZS2SQL0
WS WR
Scroll> Csr
Lines 1-20 of 22

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

Output File: Default = prefix.ZZS2XSQL.Dyyyyddd.Thhmsst.LST
Dsn/Path> _____ + Member> _____
Volume> _____ If dataset is uncataloged

Options:
DB2 Subsystem> _____
Limit > _____ 0 Max lines to be returned from SELECT
Output Type > I T=Text / M=Mapped by (generated) Structure

Type INPUT (I) to edit SQL input file.
Type OUTPUT (O) to browse a fixed output file.
Type LIST (L) to list all default (timestamped) output.
Type OPTIONS (OPT) to set further processing options.

```

Figure 13. ExecSQL Command panel.

DB2 Table Compare

SELCOPYi 3.30 introduces the DB2 Table Compare utility, an enhancement to the existing File Compare utility.

COMPTABLE Primary Command

Syntax:

```
>>- COMPTABLE -----><
```

Description:

New in SELCOPYi 3.30, COMPTABLE displays the DB2 Compare Base/Result Tables panel.

COMPFILE Primary Command

Note that the following syntax does not include all keywords but serves to highlight use of new and updated parameters.

Syntax:

```

>>- COMPFile -----><
      +- | Unformatted Compare Opts | +- | Common Opts | +-
      | | |
      +- | Formatted Compare Opts | ---+
      | | |
      +- | DB2 Table Compare Opts | ---+

```

DB2 Table Compare Opts:

```

|-- DB2 -+-----+-----+-- new_table -----+----->
      | | |
      +- (ssn) +- | new_view -----+-----+
                  |
                  +- SQL ( new_select_statement ) -----+

>-- DB2 -+-----+-----+-- old_table -----+----->
      | | |
      +- (ssn) +- | old_view -----+-----+
                  |
                  +- SQL ( old_select_statement ) -----+

```

```
> +-----+ +-----+ | | +-----+
|          V            | | |             |
+- KEY -----+ columnname +-+--+ | +- SORT ---- (order_by_clause) -+
| KEYINDEX -- indexname -----+ | +- SORTINDEX - indexname -----+

> +-----+ +-----+ | | +-----+
|          V            | | |             |
+- Select +-+ columnname +-+--+ | +- WHERE (where_clause) -+
```

Existing primary command COMPFILE has been enhanced to support DB2 Table Compare options.

Compare Base/Result Table Panels

The Compare DB2 Base/Results Tables panel (ZZS2CFT0) is opened on selection of option 9 in the DB2 primary options panel (DB2 9) or on execution of the COMPTABLE command.

This panel provides facility to compare two DB2 result tables identified via a table/view name or an SQL query statement. The result tables may be local to the DB2 server specified by the sub-system name or may exist on a remote DB2 server identified by the Location input field.

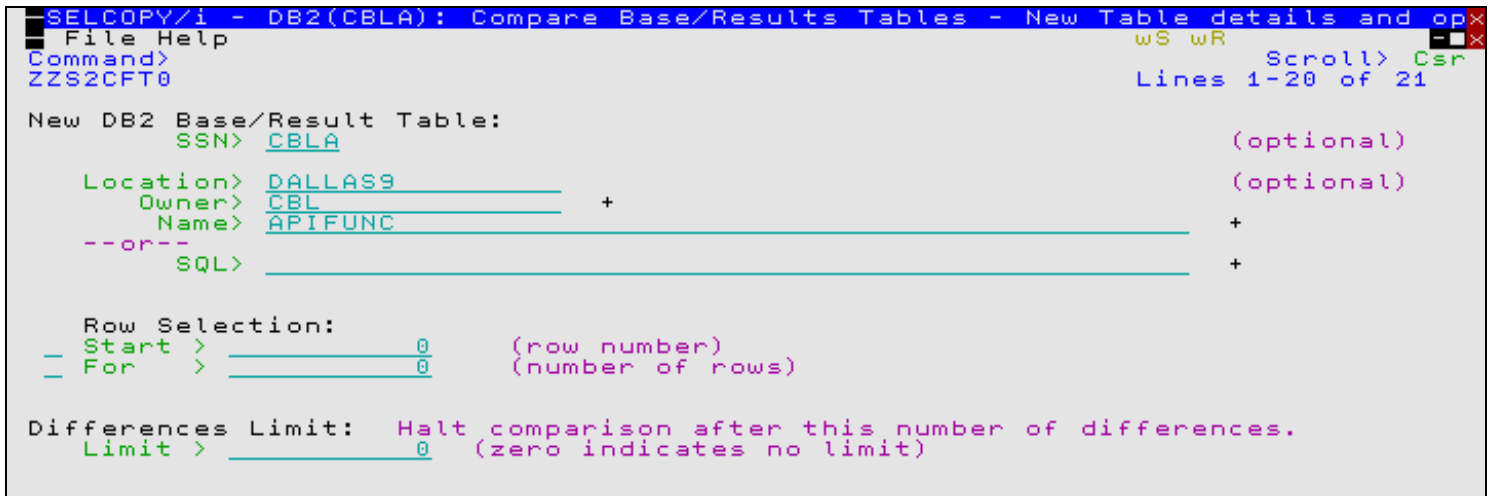


Figure 14. SELCOPYi - Compare DB2 Base/Results Tables - New table details and options.

File Compare

In addition to support for **DB2 Table Compare**, the following enhancements have been made to the Compare File utility report output. Note that the report output records are formatted by record-type field mappings defined by a generated structure and are displayed via a Data Editor browse view.

Field Subscripts

A compare of DB2 tables or records formatted using a structure is defined as being a **formatted compare**. Report output from a formatted compare may include report records mapped by the "Field" record-type. These identify the individual field names belonging to the previous record pair for which a mismatch has occurred.

SELCOPYi 3.30 has been enhanced to support display of a field name subscript (e.g. DIM(2)) to identify the element of an array (OCCURS DEPENDING) field for which the mismatch occurred. In previous releases, only the field name representing the entire array was displayed.

The CFDIFF operation (assigned to F18 by default) opens a new view of the data to display only mismatched fields belonging to the focus record pair. This has been enhanced so that, where a mismatch has occurred within an array, only those elements for which a mismatch has occurred are included in the new window view.

Contextual Matching Records

SELCOPYi 3.30 introduces support of the COMPFILE primary command parameter, CONTEXT, and an equivalent panel input field. A context value indicates that the number of (possibly matching) records that immediately precede and follow a mismatching record pair are to be displayed in the report. This provides a context for the mismatching record pair within the file data without having to include all matching records which, for large files, is likely to be prohibitive.

The File Copy utility (FCOPY) and copy/remap operations of the File Search/Update/Copy/Remap (FSU) utility support copying records from multiple library members to a single HFS file or physical sequential or VSAM dataset.

The value 0, 1 and 2 identify the format of the delimiter record to be inserted as defined below.

Additionally, if executing the File Copy utility using the FCOPY primary command in the foreground (i.e. not via the SDEMAIN batch program), then the **Allocate NonVSAM** dialog window will open automatically if the output *to_fileid* dataset does not already exist and option **NEW** is not specified (for library copy). Note that this feature already exists when executing File Copy using the utility panel.

File Erase may be performed via the ERASE primary command or via the file list prefix commands "D" or "K". SELCOPYi 3.30 enhances this functionality as follows:

Erase PDS (E) member(s) :

Suppress Successful Erase Message

However, this option does not suppress messages relating to failure to erase the file.

Primary command ERASE now supports specification of a number of blank and/or comma delimited library member name masks, having wildcard characters "*" and/or "%" as described for Library Member lists. Previous releases of SELCOPYi supported erase of a single library member only.

This new feature applies equally to library member lists where multiple members are selected for delete using the prefix commands "D", "K" or their blocked formats, "//D //" and "//K //". The feature dramatically improves performance and efficiency when multiple library members are selected for erase.

When the ERASE command or "D" list prefix command identifies multiple members for erase then, prior to erasing the members, a message window is displayed with the number of members selected. At this point, the user has the option to cancel or proceed with the member erase.

The ERASE command -P option has also been introduced, specifically for library member erase, to prompt the user for confirmation prior to erasing a member. This is the default when more than one member is selected for erase or when the "D" list prefix command is used. For each selected member, the user may confirm or skip erase, set prompt for erase confirmation off or cancel this and all subsequent member erase operations.

File Merge

SELCOPYi 3.30 introduces the File Merge utility which may be used to merge records from a number of existing input datasets into a single output dataset. The records in all of the input datasets must have already been sorted by a common "key" field in ascending order of sequence.

The utility generates SELCOPY batch control statements that may be executed immediately in the foreground or inserted in a generated JCL job stream.

MERGE Primary Command

Syntax:

```
>>- MERGE  ---+-----+-----><
           |               |
           +-- | Merge Options | ---+

Merge Options:

|---+ Foreground ---+ KEYLEN keylen -- KEYPOS keypos ----->
|               |
+- Batch -----+

                                     +-----+
                                     v         |
>--- OUTFILE out_fileid ----- INFILES  +-- in_fileid --+-----|
```

Description:

Generate SELCOPY batch control statements to merge multiple datasets, each sorted by a specified key position and length, into a single output file. *out_fileid* and *in_fileid* may be a library DSN and member name or a sequential or VSAM DSN.

Merge Datasets Panel (=8.18)

The **Merge Datasets** panel may be displayed by selecting option 18 from the Utilities menu (=8.18) or by entering the **MERGE** primary command with no parameters.


```

(2)
+- OLD ----+
|
>+-----+-----+-----+-----+
|
+- SHr ----+ +- PAGEWidth --+ n_cols --+ +- PAGEDepth --+ n_lines +
|
+- SHare --+ +- PW -----+
|
+- MODify --+
+- APPEND --+

```

Notes:

1. Asterisk (*) is optional if any other parameter is specified.
2. Default disposition is SHR if OUTDSN is a PDS/PDSE library member.

Description:

PRINT is used to open the general Print File panel or to immediately print the data belonging to a specific data set or data displayed in the current SDE Data Editor view.

SDE Print File Panel

The SDE **PRINT** panel may be displayed executing primary command PRINT with no parameters at the command prompt of a Data Editor browse or edit view.

The panel allows selection of an output file and a range of lines to be printed. The print output will reflect the display of data in the browse/edit view at the time the print operation was executed. i.e. Only records and columns that have been selected for display are printed in the sequence and format (e.g. VFMT, MAP, etc.) in which they are displayed.

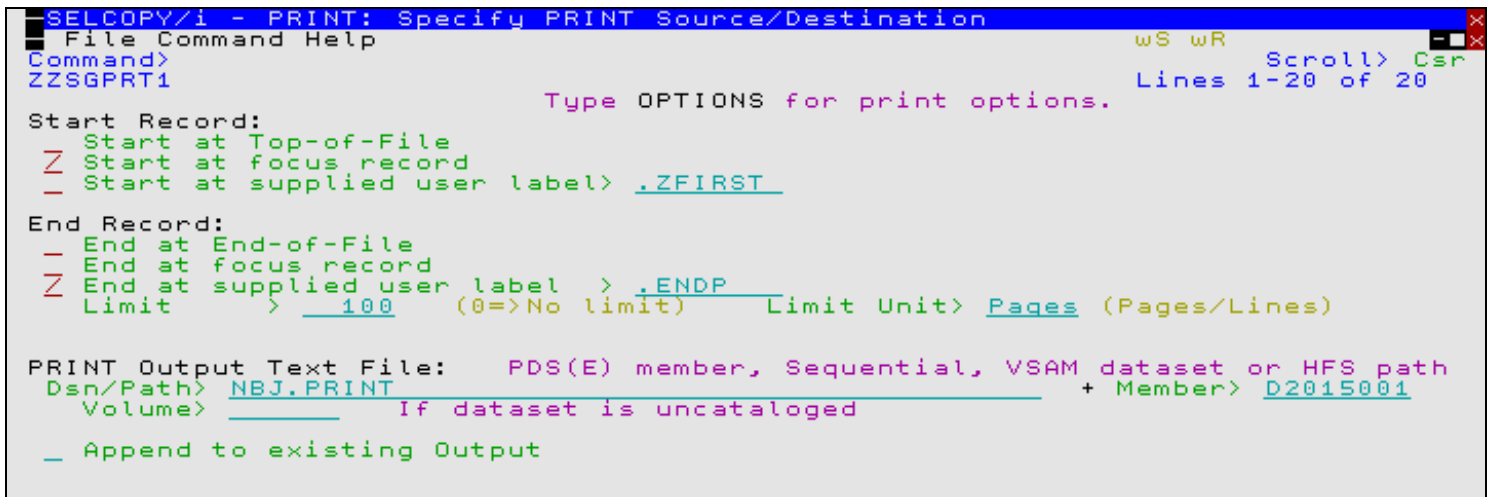


Figure 16. SELCOPYi - SDE PRINT: Specify PRINT Source/Destination

SDEMAIN (SELCOPYi SDE batch)

SDEMAIN is the name of the program used to execute SELCOPYi SDE Data Editor operations and utilities in a batch environment (SELCOPYi SDE batch).

SDE Rexx Macro Support

Macros written in the Rexx programming language may be executed in any interactive Data Editor browse or edit window view. The named environment, CBLSDATA, may be addressed in order to execute Data Editor primary commands, including commands that require the SDE environment (e.g. FCOPY, FSU, COMPILE, XMLGEN).

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

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

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

    address CBLSDATA;          'extract /region/'

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

    return rv
%%
```

Figure 17. Sample SDEAMAIN - SELCOPYi SDE batch job stream.

SELCOPY Debugger

The SELCOPY Debugger utility is used to assist in trouble shooting SELCOPY batch component control statements. It provides the ability to step through control statements, track address pointers and to display different areas of storage owned by the application.

JCL Job Stream Input

In previous releases of SELCOPYi, the SELCOPY debugger utility could only accept SELCOPY control statement (SYSIN) input. In order to support SELCOPY SYSIN embedded in JCL job streams, the JCLCMX macro was provided as a rudimentary tool to split job steps into separate groups of dataset allocation and task execution commands. These commands could then be executed in the foreground (TSO) to establish the I/O datasets required by the SELCOPY job step to be debugged.

SELCOPYi 3.30 advances this JCLCMX facility so that it is integrated into the SELCOPY debugger. The SELCOPY Debug utility panel now accepts JCL library member input which is then analysed and displayed as a job-step selection list.

Job steps may be selected for execution or ignored as applicable and, furthermore, SELCOPY job steps may be selected for execution with or without use of the SELCOPY Debugger. Allocation and subsequent freeing of dataset DDnames, referenced by selected job steps, is performed silently by the utility.

Sel	Debug	Step	PGM	PROC	Status	RetCode	Info
		STEP010	IDCAMS		Pending		
		STEP050	SELCOPY		Pending		
		DEL01	IEFBR14		Pending		
		SELCOP1	SELCOPY		Pending		
		DELETE	IEFBR14		Pending		
		SORT001	SORT		Pending		
		REPRO	IDCAMS		Pending		

Figure 18. SELCOPY/debug - Job-Step Selection List.

XML Generation

In the previous release of SELCOPYi, the XML generation utility was introduced via the XMLGEN primary command which applied only to data in the current Data Editor browse or edit view.

SELCOPYi 3.30 extends this utility to support an XMLGEN primary command that is not dependent on an open Data Editor view and also to support both a general and SDE Dator Editor specific XML Generation utility panel.

XML output may be generated from any data that is formatted by a structure (SDO) either explicitly, via a nominated SDO dataset, or implicitly via a nominated COBOL, PL1, Assembler copybook/DSECT or DB2 result table.

Fields, defined by a structure record/table row mapping, identify the XML tag names and attributes of variables in the XML output. The XML output may be restricted to a subset of the input records (or table rows) and fields. Furthermore, fields may be resequenced before the XML output is generated.

If executed from a Data Editor browse or edit view, XML generation may be performed on the currently displayed data, operating only on non-excluded lines and using only columns selected for display.

XMLGEN Primary Command Enhancements

Note that the following syntax does not include all keywords but serves to highlight use of new and updated parameters.

Syntax:

```
>>- XMLgen +-----+><
|         | Input dataset specification | --+ | Common Options | -+
|         |         +- .ZFIRST -- .ZLAST --+ |
| (1)     |         |         |         |
|         |         +- * -----+
|         |         +- .name1 -----+
|         |         |         |         |
|         |         +- .name2 -----+
```

Input dataset specification:

```
| - INDSN ( input_dataset_specification ) -----|
```

Common Options:

```
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
|         |         |         |         |         |         |         |         |
|         |         +- 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 +-
```

```
+- GROUP -----+ +- INDent 1 -----+         +- BROWSE +-
|         |         |         |         |         |         |         |         |
>+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|         |         |         |         |         |         |         |         |
+- ELEmentary +- +- INDent n_cols +- +- LIMit n_lines +- +- EDIT ---+
|         |         |         |         |         |         |         |         |
|         |         |         |         |         |         |         |         |
+- NOVIEW +-
```

Notes:

1. Asterisk (*) is optional if any other parameter is specified.

Existing primary command XMLGEN has been enhanced to support the following:

- The input dataset specification is in the form of a Data Editor BROWSE command except that the BROWSE command verb itself is not specified.
BROWSE keyword options such as **FROM**, **FOR**, **FILTER** and **VIEW** may be specified to limit records for which XML is generated. Similarly, DB2 table browse parameters (e.g. **SQL**) may be used to specify DB2 result table input.
- For execution on data in the current Data Editor view only, the start/end line labels (.name1/.name2) may be used to select the first and last lines of a range of displayed lines for which XML will be generated.
- Mutually exclusive **GROUP** and **ELEMENTARY** options that determine whether parent group fields are omitted from (ELEMENTARY) or included in (GROUP) the XML output. Inclusion of the group fields creates a hierarchy of XML tags.
- Option **LIMIT** restricts the number of input records, record segments or DB2 table rows that may be processed by the utility.

XML Generation Panels

The XML Generation utility supports two panel formats as follow:

XML Generation Utility Panel

The **XML Generation** utility panel may be displayed by selecting option 16 from the Utilities menu (=8.16) or by entering the primary command XMLGEN (or simply XML) with no parameters from any window that is not a Data Editor view.

This panel allows the user to nominate the input data source and structure used to produce an exportable copy of a structured dataset as extended markup language (XML) text.

Where the input structure maps more than one record-type, the **VIEW** (F22) primary command may be issued in order to restrict which record-types are processed. Similarly, the **SELECT** primary command (assigned to F5 by default) may be issued in order to restrict and re-sequence the fields defined by the selected record-type mapping.

The panel may perform the XML generation in the foreground or create a JCL job stream that may be submitted to batch.

```

SELCOPY/i - XML Generation: Specify Source Structured Data File
File Command JCL Structure Replace Help          wS wR
Command>                                     Lines 2-21 of 23
ZZSGXML0
Data File: PDS(E) member, Sequential, VSAM dataset or HFS path
Dsn/Path> NBJ.SELCTRN.ZZST2DAT + Member>
Volume> If dataset is uncataloged

Structure/Copybook overlay: Recompile> N
Dsn> NBJ.SELCTRN.SAM1 Member> ZZST2CPC
Type: / SDO - AData - Cobol - PL1

Record Selection:
- Start> + / Record - Key - RBA
- For> 0 # records
Z Filter> F Select records (F=File; Q=Quick) (F6=Edit Filter)
File> CBL.RRD.FILTER + Member> TDTMM1

HFS Input 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)
  
```

Figure 19. XML Generation Utility Panel.

SDE XML Generation Panel

The **SDE XML Generation** panel may be displayed by entering the primary command XMLGEN (or simply XML) with no parameters from any Data Editor view.

The XML generation will process only fields in the sequence they are displayed in the browse/edit view, as customised by the **SELECT** primary command (or "Select Display Fields" panel). Furthermore, it will bypass all excluded records and records not within a selected range of lines (defined by line labels). Unlike the general utility panel, the XML generation is executed in the foreground.

SELCOPY/i - XML Generation: Specify XML Source/Destination

File Command Help

Command> ZZSGXML1

Type OPTIONS for translation and other options.

Start Record:

- Start at Top-of-File
- Start at focus record
- Z Start at supplied user label> .XMLB

End Record:

- End at End-of-File
- Z End at supplied user label > .XMLE
- Maximum output records > 0 (0=no limit)

Output XML Text File: PDS(E) member, Sequential, VSAM dataset or HFS path

Dsn/Path> NBJ.XMLGEN + Member> D2015001

Volume> _____ If dataset is uncataloged

_ Append to existing Output

HFS Output Options:

EOL Characters> NL NL|CR|LF|CRLF

Figure 20. SDE XML Generation Panel.

SELCOPYi Lists

SELCOPYi List windows are a specific class of windows which display information as a table of rows and columns. List windows have a common set of features and are often used as a springboard to the editors and utilities.

SELCOPYi 3.30 includes enhancements to existing lists and introduces a number of new list window types.

General List Enhancements

PFKey Assignment

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

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

SMS Storage Groups (=3.14)

SELCOPYi 3.30 introduces support for listing SMS Storage Group attributes.

LSG Primary Command

Syntax:

```
>>--+ LSG -----+-----+-----+-----><
      |              |      |      |
      +- LISTSTORAGEGROUPS -+ +- SGName ----+
```

Description:

Use the LSG command to open an SMS Storage Group list window and optionally specify a storage group name mask.

SMS Storage Groups List

The Storage Groups List window displays the defined System Managed Storage (SMS) Storage Groups. It may be opened via the following:

- Select option 14 'StorGrps' from the List Menu. (=3.14)
- Select 'SMS Storage Groups' from the Utilities/List menu in the CBL main window menu bar.
- Enter command LSG on the command line of any window.

The Storage Groups List window displays the name, description and other information relating to the Storage Group definition.


```

SELCOPY/i - SMS Storage Groups 2014/12/08 0
View Refresh Back Forward FDB Text Help wS wR Scroll> Csr
Command>
SMS Storage Group name pattern>
- SGName- --LUU--- ---LUD--- --LUT---
CBLDB2 NBJ 2014/10/11 06:39 CBL STORAGE GROUP - CBL DB2 VOLU
CBLEAV NBJ 2014/11/19 11:13 CBL EAV VOLUMES STORAGE GROUP
CBLSMS NBJ 2014/11/16 12:41 CBL STORAGE GROUP - CBL DATASET
CBLVIO NBJ 2014/10/11 06:42 CBL STORAGE GROUP - VIO UNIT MAX
DBCLASS IBMUSER 2013/04/24 21:35
DBCLASSB IBMUSER 2013/11/20 02:15 STORAGE GROUP FOR DB2 V11 DIRECT

```

Line 1 of 6 | Col 1 of 770 | Views 1 | select *

Figure 21. Storage Groups List window.

SMS Storage Group Volumes (=3.15)

In conjunction with support for SMS Storage Group lists, SELCOPYi 3.30 also introduces support for listing SMS Storage Group volumes.

LSGV Primary Command

Syntax:

```

>>--+ LSGV -----+-----+-----+-----><
      |             |             |             |
      +- LISTSTORAGEGROUPVOLS -+ +--- SGName -+-----+
                                   |             |
                                   +--- volmask --+

```

Description:

Use the LSGV command to open an SMS Storage Group Volumes list window and optionally specify storage group name and volume id mask. The list displays volumes belonging to an SMS pool storage group.

SMS Storage Group Volumes List

The Storage Group Volumes List window displays the volumes assigned to a specific System Managed Storage (SMS) Storage Group. It may be opened via the following:

- Select option 15 'StorGrps Vols' from the List Menu. (=3.15)
- Select 'SMS StorGrp Vols' from the Utilities/List menu in the CBL main window menu bar.
- Enter command LSGV on the command line of any window.
- Enter list prefix command "V" against an entry in the List Storage Groups window.

The Storage Group Volumes List window displays the attributes of DASD volumes defined to a Pool type SMS storage group.

Vol	SGName	TotalMB	FreeMB	LExtentMB	TTotalMB	TFreeMB
CBLM01	CBLSMS	2707	404	61	2707	
CBLM02	CBLSMS	2707	423	73	2707	
CBLM03	CBLSMS	2707	417	94	2707	
CBLM04	CBLSMS	2707	270	65	2707	
CBLM05	CBLSMS	2707	419	117	2707	
CBLM06	CBLSMS	2707	189	36	2707	
CBLM07	CBLSMS	8120	600	72	8120	
CBLM08	CBLSMS	8120	1221	585	8120	1
CBLM09	CBLSMS	2707	273	45	2707	
CBLM10	CBLSMS	2707	407	81	2707	
CBLM11	CBLSMS	2707	414	112	2707	
CBLM12	CBLSMS	2707	479	73	2707	
CBLM13	CBLSMS	8120	277	23	8120	
CBLM14	CBLSMS	8120	1380	531	8120	1
CBLM15	CBLSMS	8120	1150	541	8120	1

Figure 22. Storage Group Volumes List window.

Library Member Move/Copy

SELCOPYi 3.30 introduces efficiencies in the use of Library Members List prefix commands "M" and "C" (or their block formats "/M" and "/C") to move or copy multiple members to another library.

In previous releases, selecting multiple members in this way would treat each move, move block, copy or copy block as a separate transaction and a separate utility window displayed for each.

In most cases, it is intended that selection of multiple member entries using prefix commands "M" or "C" should move or copy all selected members to the same target library. As such, SELCOPYi has been updated so that multiple member entries selected for move or copy are actioned in a single transaction.

To support this, SELCOPYi 3.30 introduces the "Library Member Move/Copy" panel to allow specification of the target library. Note that, if only a single member has been selected, the panel allows copy or move to a new member name.

The contents of the Library Member Move/Copy panel depends on whether it has been started with multiple or single members selected for move or copy from the library list. For a single member, the input member field is displayed. For multiple members, a comment is displayed indicating that the selection of members may be displayed and updated using the SELECT primary command or menu option.

On pressing the <Enter> key or selecting "Command" or "JCL" menu items, a check is made to determine whether the target library exists. If not, the "Allocate NonVSAM" dialog is started to allocate a new library. Default allocation attributes are modelled on the source library.

SELCOPY/i - Library Member Move

File Help JCL Command Select

Command> ZZSGMOV0

Input PDS/PDSE Library and Member mask:

Library DSN: NBJ.CBLI.CBLE.FSU

Member: DBAPAR (Move single member)

Output PDS/PDSE Library:

Library DSN> _____

Member> _____ Replace existing members.

1. Help (F1) 2. Execute (ENTER) 3. Back (F3) 4. Exit (F15)

Figure 23. Library Member Move - Single member.

DB2 Objects

SELCOPYi 3.30 includes a number of new DB2 object lists and enhancements to existing list features.

Index Keys

The List Indexes panel (ZZS2LINX) may be used to list indexes defined at the local or remote DB2 server. It may be started via the following:

- Select option 7 Indexes from the List DB2 Object options menu. (DB2 7.7)
- Execute the command **LDINDEX** with or without parameters from the command line of any window.
- Execute list prefix command "XL" against an entry in the DB2 List Tables panel.

Panel input fields support standard DB2 pattern-expression wild cards ('%' and '_') and may be amended to apply a filter before rows are fetched from the relevant catalog tables.

```

SELCOPY/i - DB2(CBLA): List Indexes
View Refresh Back Forward FDB Text Help
Command>
ZZS2LINX
DB2 Index list:

Location>
Index Owner> %
Index Name > %
+ Table Owner> NBJ
+ Table Name > %

-----NAME----- CREATOR -----TBNAME----- TBCREATOR UNIQURULE COLCOUNT
EMP1IX0 NBJ EMP1 NBJ U 2
EMP1IX1 NBJ EMP1 NBJ U 3
ISAMPTDOCU_I812BJ1 NBJ SAMPTDOCU_I812NXCZ NBJ U 2
I_DOCIDXML01 NBJ XML01 NBJ X 1
I_NODEIDXXML01 NBJ XXML01 NBJ N 2
NBJDEFT1 NBJ DEFTEST NBJ U 2
SELCTRN_#_CGE NBJ SELCTRN_TRACK NBJ P 1
SELCTRN_#_LCB NBJ SELCTRN_ALBUM NBJ P 1
SELCTRN_#_PCV NBJ SELCTRN_ARTIST NBJ P 1
TESTAB01#IX0 NBJ TESTAB01 NBJ U 3
TESTAB01_#_76R NBJ TESTAB01 NBJ P 3
Line 1 of 13 | Col 1 of 695 | Views 1 | select *

```

Figure 24. DB2 List Indexes.

Packages

The List Packages panel (ZZS2LPKG) may be used to list packages defined at the local or remote DB2 server. It may be started via the following:

- Select option 18 Packages from the List DB2 Object options menu. (DB2 7.18)

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.

```

SELCOPY/i - DB2(CBLA): List Packages
View Refresh Back Forward FDB Text Help
Command>
ZZS2LPKG
DB2 Package list:

Location>
Name> %
Collection> CBL%

-COLLID- --NAME-- ---CONTOKENX--- OWNER CREATOR -----TIMESTAMP-----
CBLPACK1 SELCOPQX 1982C4E21FF9D341 NBJ NBJ 2013-10-22-12.56.35.335305
CBLPACK1 SQXFCHK0 197D86601308FC81 NBJ NBJ 2010-10-27-16.27.18.983980
CBLPACK1 SQXFCNV0 197D866109FE8681 NBJ NBJ 2010-10-27-16.27.18.906988
CBLPACK1 SQXFDBA0 197D86611A5C0941 NBJ NBJ 2013-04-15-11.21.34.521244
CBLPACK1 SQXFFKR0 19B3EF690EA3D7B1 NBJ NBJ 2010-10-27-16.27.19.159738
CBLPACK1 SQXFEXX0 19B3EF691C97E216 NBJ NBJ 2010-10-27-16.27.19.291210
CBLPACK1 SQXFOBJ0 19B3EF6A14DED16D NBJ NBJ 2014-08-20-17.36.38.080588
CBLPACK1 SQXFSGA0 197D9183021917C1 NBJ NBJ 2013-09-06-11.36.04.934703
CBLPACK1 SQXFSQL0 19B3EF6B07F23AD0 NBJ NBJ 2010-10-27-16.27.19.434563
CBLPACK1 SQXFSQL1 197D86840467AB01 NBJ NBJ 2010-10-27-16.27.19.605944
CBLPACK1 SQXFSTR0 19B48A2500289499 NBJ NBJ 2014-08-20-17.36.38.955058
CBLPACK1 SQXFSTAB1 19C6F0B60A82651A NBJ NBJ 2010-10-27-16.27.19.769457
Line 1 of 14 | Col 1 of 501 | Views 1 | select *

```

Figure 25. DB2 List Packages.

Package Dependencies

The List Package Dependencies panel (ZZS2LPKD) may be used to list dependencies for the selected package defined at the local or remote DB2 server. It may be started by executing list prefix command 'DEP' against an entry in the List DB2 Packages 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.

```

SELCOPY/i - DB2(CBLA): List Package Dependencies
View Refresh Back Forward FDB Text Help
Command>
ZZS2LPKD
DB2 Package Dependencies list:

Location>
Name> SQXFCHK0
Collection> CBLPACK1

-DNAME-- BQUALIFIER ---BNAME--- BTYPE DLOCATION DCOLLID- ---DCONTOKENX--- IBM
- SQXFCHK0 SYSIBM DSNDX01 I CBLPACK1 197D86601308FC81 N
- SQXFCHK0 SYSIBM DSNSDX01 I CBLPACK1 197D86601308FC81 N
- SQXFCHK0 SYSIBM DSNSCX01 I CBLPACK1 197D86601308FC81 N
- SQXFCHK0 DSNDDB06 SYSDBASE R CBLPACK1 197D86601308FC81 N
- SQXFCHK0 DSNDDB06 SYSSTR R CBLPACK1 197D86601308FC81 N
- SQXFCHK0 SYSIBM SYSCHECKS T CBLPACK1 197D86601308FC81 N
- SQXFCHK0 SYSIBM SYSCHECKDEP T CBLPACK1 197D86601308FC81 N
- SQXFCHK0 SYSIBM SYSCOLUMNS T CBLPACK1 197D86601308FC81 N

Line 1 of 8 | Col 1 of 107 | Views 1 | select *

```

Figure 26. DB2 List Package Dependencies.

Package Privileges

The List Package Privileges panel (ZZS2LPKP) may be used to list privileges for the selected package defined at the local or remote DB2 server. It may be started by executing list prefix command "P" against an entry in the List DB2 Packages 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.

```

SELCOPY/i - DB2(CBLA): List Package Privileges
View Refresh Back Forward FDB Text Help
Command>
ZZS2LPKP
DB2 Package Privileges list:

Location>
Name> SQXFCHK0
Collection> CBLPACK1

GRANTOR GRANTEE LOCATION -COLLID- --NAME-- -----TIMESTAMP----- GRANTEE
- NBJ JGE CBLPACK1 SQXFCHK0 2014-08-26-15.27.00.435372
- NBJ LAC CBLPACK1 SQXFCHK0 2013-11-13-18.25.35.308903
- NBJ JGE2 CBLPACK1 SQXFCHK0 2012-06-25-14.18.24.023495
- NBJ NBJ CBLPACK1 SQXFCHK0 2010-10-27-16.27.18.983980

Line 1 of 4 | Col 1 of 132 | Views 1 | select *

```

Figure 27. DB2 List Package Privileges.

Related Tables

The List Related Tables panel (ZZS2LREL) may be used to list DB2 tables that are involved in a referential constraint (parent/foreign key) relationship as the parent or dependent of the selected DB2 table. It may be started via the following:

- Execute the prefix command "RL" against a DB2 table entry in a List DB2 Tables panel or another List Related Tables panel.
- From a DB2 table data edit view, execute the prefix command "RE" or primary command REDIT against a row for which related table rows will be displayed. See [Related Table Edit](#) DB2 feature of the Data Editor.

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.

```

SELCOPY/i - DB2(CBLA): List Related Tables
File Edit Actions Options Utilities Window SwapList Help  wS wR
Command>
ZZS2LREL
List tables related to:

Location> CBLAZOS
Owner> CBL
Name> ZZSFUNC

Related Tables
Sel  Related  Related  Relationship  Relationship  Foreign  Delete  Enforced
     Table    Table    Name          Type          Key      Rule    by DB2
     Owner    Name
-----
<--> <->      <---->      <-----1->  <----->    <---->  <---->  <->
*** Top of Data ***
CBL  ZZSFUSE  ZZSFUSE_REF2  DEPENDENT    2  Restrict  Yes
CBL  ZZSPARM  ZZSPARM_REF2  DEPENDENT    2  Restrict  Yes
CBL  ZZSRETV  ZZSRETV_REF2  DEPENDENT    2  Restrict  Yes
CBL  ZZSLIBS  ZZSFUNC_REF1  PARENT       1  Restrict  Yes
CBL  ZZSSMOD  ZZSFUNC_REF2  PARENT       1  Restrict  Yes
*** End of Data ***
  
```

Figure 28. DB2 Related Tables.

List DB2 Object Enhancements

The following general enhancements have been applied to DB2 object list windows.

- All existing DB2 object list panels have been updated to support a Location input field. This allows lists of objects belonging to remote DB2 servers to be displayed without having to manually connect to the required sub-system.

A selectable list of recognised DB2 server locations is displayed if a wildcard entry is entered in the Location field.

The selected server location is passed to any object list panels displayed as a result of a prefix command executed against an entry in the current list.

- Where applicable, support for the following list prefix area commands has been included.

RL	Display the DB2 Related Table object list.
X	Display the DB2 Create Index panel views.

SELCOPYi Panels

Many of SELCOPYi's applications and utilities support a front end dialog window of the interactive panel class. Examples of these utility panels may be found under **SELCOPYi Utilities** in this document.

SELCOPYi 3.30 includes a number of general enhancements to windows assigned the interactive panel window class.

Processing Enhancements

Dataset Selection

Panel views may include file identification input field groups (e.g. Volume, DSN and Member fields) that identify the source and/or target files used by a utility.

Specification of a wildcard character asterisk (*) or percent (%) in any of these fields will display a list of fileids (datasets, member names or HFS paths) that match the specified fileid mask. For DSN masks, this selection list is a list of cataloged dataset names displayed within the "Select a Dataset" panel (ZZSGSLD0).

SELCOPYi 3.30 reduces the time taken to generate this selection list by generating a Catalog list instead of a Dataset list by default. Associated dataset information is less detailed in a Catalog list since, unlike a Dataset list, VTOC information is not retrieved.

If required, the more detailed Dataset list output may still be generated on selecting the Extended Details option field in this panel.

Embedded Table Primary Commands & Options

Panels may contain embedded tables. The table display may be manipulated and data within an embedded table may be updated, resequenced, duplicated and deleted as required using a set of table edit primary and line commands.

The following additional table primary commands have been introduced in SELCOPYi 3.30.

CCOLOUR SET/QUERY/EXTRACT

Syntax:

```
>>-----+-----+----- CColour --- column_name ----->
|         |         |         |         |         |
+- SET -+-|         |         +- ccolour_id -+
|         |         |         |         |         |
|         +--- NONE -----+
>--+--+ Blue -----+-----+----- When --+ expression ----->
|    |   Red -----+   BLInk -----+   WHERE +-
|    +- Pink -----+   REVvideo --+
|    |             |
+- Green -----+   Uscore ----+
|    | Turquoise -+
|    | Yellow ----+
|    +- White -----+
|    | Default ---+
|-- OFF -----+

```

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

```

```
>>---- EXtract --- /CColour +-----+----- / ----->
|         |         |
+- column name -+

```

Description:

Conditionally set the colour display of individual values within a column based on a true result from a supplied SDE expression.

REFRESH

Syntax:

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

Description:

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

REFRESH will discard any table editing and row selections that have been performed. Furthermore, if the table is based on list information, SQL tables or other panel tables that have been updated since the table was created, then the refreshed table data will reflect these changes.

e.g. A panel table based on the contents of a PDS library will reflect the members at the point the table is refreshed. Members that have been deleted will not appear in the refreshed table whereas members that have been renamed or added since the table was first created will be displayed accordingly.

SELECTALL

Syntax:

```
>>-- SELECTall ---+-----+-----><
                |         |
                +-- Deselect --+
```

Description:

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

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

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

VLMAX

Syntax:

```
>>-- VLMAX -----><
```

Description:

Sets the display width of all variable length columns to the maximum width.

VLMIN

Syntax:

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

Description:

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

SELCOPYi Miscellaneous Enhancements

Additional miscellaneous enhancements included in SELCOPYi 3.30 are documented here.

3390 Extended Address Volumes (EAV)

SELCOPYi now fully supports 3390 Extended Address Volumes (EAV).

These types of volume may comprise between 65521 and 262668 cylinders and introduce the concept of cylinder (as opposed to track) managed space. To accommodate this, Volume Table of Contents (VTOC) datasets on EAV volumes contain additional DSCBs.

DASD Volume and Dataset Lists

Information displayed by DASD Volume lists (LVOL) now includes the following **additional** columns:

Name	Description
CYLMS	Cylinder managed storage
EXATTR	Extended attribute DSCBs
STARTCYL	Cylinder managed space start
MINCYL	Minimum cylinder allocation unit
ALTCYL	Number of alternate cylinders
STORGRP	SMS storage group
TMFREEEXTN	Track managed total free extents
TMFREECYL	Track managed total free cylinders
TMFREETRK	Track managed total free tracks
TMMAXCYL	Track managed largest free extent (cylinders)
TMMAXTRK	Track managed largest free extent (tracks)
TMFRAGINDX	Track managed fragmentation index
VOLTRACKS	Total tracks on volume
TMTRACKS	Track managed total tracks on volume

Information displayed by Dataset lists (LD) now includes the following **additional** columns:

Name	Description
Stripes	SMS - Stripe Count for striped datasets
KRQual	VS - VSAM Key range qualifier
VolPrime	Vol - Primary volume
VolCandi	Vol - Candidate volume
VolOFlow	Vol - Overflow keyrange volume
VolConv	Vol - Converted VSAM dataset volume
VolNVSAM	Vol - NonVSAM
VolKyRng	Vol - Keyrange qualifier exists.
VolVPCI	Vol - Primary VVR
VolSSQWD	Vol - Sequence set with data
Flag1	DS - Dataset flags (DS1FLAG1) (hex)
LastTrack	DS - Last used track
LastBlock	DS - Last used block on last used track
LastFree	DS - Space remaining on last used track
Compressable	DS - Compressable extended format
Checkpointed	DS - Checkpointed
Large	DS - More than 64K tracks
EAttrOpt	DS - Optional extended attributes specified
EAttrNo	DS - No extended attributes specified

Large Format Datasets

Support for EAV volumes includes support for physical sequential datasets allocated as DSNTYPE=LARGE. See support for [Large Format Datasets](#) in the SELCOPYi Data Editor for further information.

Large physical sequential datasets may now be allocated via the SELCOPYi "Allocate Non-VSAM" dialog which includes specification of Type LARGE. This panel also supports specification of Extended attributes as NO or OPT (corresponding to allocation parameter EATTR) which determines whether the dataset can reside in the extended (cylinder managed) EAV volume address space.

By default, on using the Data Editor to edit a non-KSDS dataset which is too large to be loaded into available storage, Auxiliary Edit is performed. This involves first taking an auxiliary copy of the dataset before loading a subset of records belonging to the file.

Unless explicitly defined by values in the "Data Edit Auxiliary Dataset Settings" panel (=0.4.4), the auxiliary data set is allocated with the same Unit and SMS classes as the edited data set. This now includes DSNTYPE=LARGE and EATTR attributes if applicable.

FIRSTUSE utility

Execution of the FIRSTUSE utility is no longer required by regular users as it is executed automatically on startup. However, following an install, the Systems Programmer must execute FIRSTUSE to establish new or updated System INI file options which are applicable to all users.

The utility has been enhanced in SELCOPYi 3.30 as follows.

Automatic Execution

Following install of a new SELCOPYi version, the Systems Programmer must start SELCOPYi in order to run the FIRSTUSE utility for system-wide customisation.

On install of SELCOPYi 3.30 and subsequent versions, FIRSTUSE will execute automatically and so prevent accidental omission of this step in the post-install customisation process.

INI Dataset Update

Since FIRSTUSE writes system-wide customisation options to the System INI dataset, a problem arises if the Systems Programmer customising SELCOPYi does not have ACF UPDATE authority for this dataset. Note that the DSN of the System INI dataset is identified by the CBLNAME module option, INamDSN.

The FIRSTUSE utility has now been updated to verify read/write access to the INI dataset to be updated prior to displaying customisation prompts. If the user does not have the required authority, the utility ends with an explanatory error message.

Window Manipulation

The following general enhancements have been made to SELCOPYi window display and window manipulation.

Initial Windows State

At almost every mainframe installation, 3270 terminals are no longer physical devices but are, in general, emulated by PC or Unix software packages. This software, together with VTAM support of dynamic terminal sizes, means the number of rows by columns may exceed that provided by the traditional Model 2/3/4/5 physical terminals.

Because 3270 terminal displays are character based as opposed to pixel based, the SELCOPYi windowing environment operates best with terminals that have a large number of defined rows and columns. i.e. Display of the window frame characters, etc. does not unduly affect display of the window content.

To provide the optimum working environment, the initial state of SELCOPYi windows is either maximised or non-maximised depending upon the number of rows and columns provided by the 3270 terminal. This behaviour is governed by INI variable System.InitialStateAuto which is set to YES by default.

If SELCOPYi 3.30 is started in a 3270 terminal with number of rows less than 32 or number of columns less than 132 then all windows are maximised by default. In previous versions, maximised state occurred only if the number of columns was less than 80 with no check on the number of rows.

MOVEWINDOW/SIZEWINDOW Primary Commands

SELCOPYi environment primary commands MOVEWINDOW and SIZEWINDOW accept a window name parameter identifying the window to be re-positioned or re-sized respectively.

Since the window names are not displayed by default (requires execution of WINDOWNAMES primary command), SELCOPYi 3.30 introduces support of window title as an alternative parameter on MOVEWINDOW and SIZEWINDOW. Like window name, window title identifies the window and may be specified as a substring so long as it uniquely identifies a window's title.

PFKey Definitions

The following enhancements have been made to SELCOPYi keylist display.

PFSHOW Primary Command

Syntax:

```
>>--+- PFSHOW +------+-----+-----+----->>
      |          |         |         |
      +- FKA ----+        +- ON ---+
                        |         |
                        +- OFF --+
                        |         |
                        +- ALL ---+
                        |         |
                        +- LONG --+
                        |         |
                        +- SHORT -+
```

Description:

PFSHOW controls the function key display area at the bottom of the screen. SELCOPYi 3.30 introduces support for parameter ALL which displays the same PFKey definitions as for operand LONG but includes empty slots where PFKey definitions are not format LONG.

On a 160 width screen, this means that Shift-F1 always appears below F1, Shift-F2 below F2, etc.

PFSHOWSTYLE Primary Command

Syntax:

```
>>--+- PFSshowstyle --+-----+-----+----->>
      |               |         |         |
      +- PFS -----+   +- STD ---+
                        |         |
                        +- SHIFT -+
```

Description:

Introduced in SELCOPYi 3.30, PFSHOWSTYLE controls the display of Function Keys F13-F24, which are normally assigned to your PC keyboard as **Shift**-F1 to **Shift**-F12. PFS without parameters toggles between the two modes, SHIFT and STD.

If PFSHOWSTYLE SHIFT is used, these PFKeys are displayed as s1-s24, otherwise they are displayed as F13-F24.

TITLEBAR and BORDER PFKey Display

In addition to being assigned to the display area of different utility windows (panels, lists, etc.) common key list definitions are assigned to the title bar and borders of every window.

SELCOPYi has been enhanced so that, if the cursor is positioned in a window title bar or window border when an AID key is pressed (i.e. <Enter> or any PFKey), then the PFShow display area gets updated to display TITLEBAR or BORDER PFKeys respectively.

Likewise, execution of the KEYS command to view and update PFKey assignments, is now also sensitive to the cursor being positioned in a window title bar or window border.

Regular Expression Enhancement

Regular expressions are supported by the Text Editor, Data Editor and the File Search/Update/Copy/Remap utility to search data using complex pattern matching.

SELCOPYi regular expression support follows the specification as defined by Mansfield Software's KEDIT product. For completeness, SELCOPYi 3.30 includes a minor enhancement to allow specification of octal (base 8) escape sequences to represent character values.

e.g. \301 (octal) = 193 (decimal) = x'C1' (hexadecimal) = c'A' (EBCDIC)

SELCOPYi Session Count

In order to track SELCOPYi product usage, a record of the number of times SELCOPYi has been started by a user is maintained in the user's personal INI dataset as variable System.Session.

This value is reported as session number in the "About SELCOPYi" window, opened using primary command ABOUT or on selecting Help->About SELCOPYi from the CBL main window menu bar.

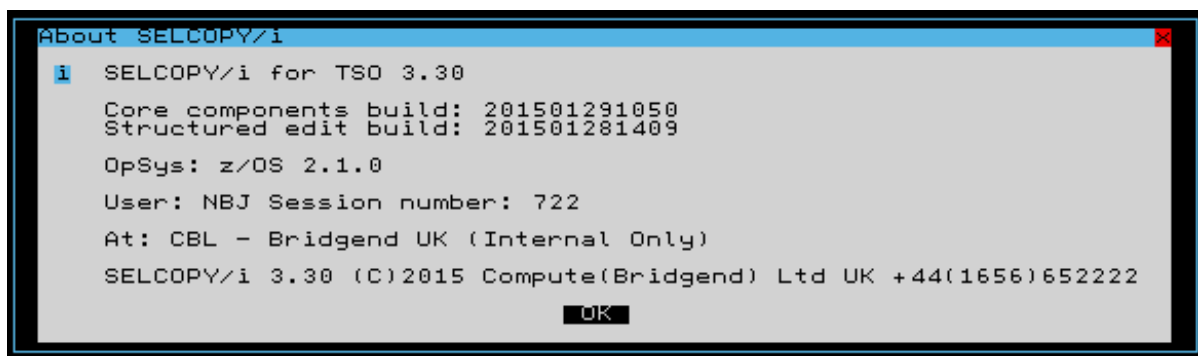


Figure 29. About SELCOPYi window.

Maintenance Applied

The following table identifies maintenance to SELCOPY Product Suite 3.20 that has been applied at source to SELCOPY Product Suite 3.30.

Details of each SYSMOD may be found at the CBL web page entitled "[SELCOPY Product Suite 3.20 Maintenance Summary](#)". Individual descriptions are referenced by links to this web page in the following table.

Service Package Id	SYSMOD
X2013317	RS32001
X2014057	RS32002
X2014063	RS32003
X2014105	RS32004
X2014132	RS32005
X2014241	RS32006
X2014323	RS32007
X2014351	RS32008