Compute (Bridgend) Ltd

Product Suite New Features

Release 3.40

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

CBL Ref: z:\hst\dist\all\snf340\SELCOPY_Product_Suite_3.40_New_Features.pdf

Contents

Contents

SELCOPY Product Suite 3.40 New Features

Documentation Notes

Information in this New Feature List details changes introduced to CBL software since SELCOPY Product Suite 3.30.

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
- SELCOPY C++ (SLC) Language Reference
- CBLVCAT User Manual
- SELCOPYi Reference and User Guide
- SELCOPYi Text Editor (CBLe) Manual
- SELCOPYi Structured Data Editor Manual
- SELCOPYi Training Manual

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 3.40 includes updates to the following product components.

SELCOPYi

z/OS Systems

SELCOPYi has been updated extensively for z/OS systems. The significant new features in SELCOPYi 3.40 for z/OS are:

◊ SELCOPY Debugger support for the SLC program.
◊ Internal COBOL parser negates requirement for Enterprise COBOL compiler.
◊ DB2 XML column data edit, view, browse, export and import.
◊ DB2 scrollable cursors for edit/browse of large DB2 result tables.
◊ DB2 LOAD and UNLOAD utility panels for DB2 tables.
◊ DB2 object ALTER and RENAME panels.
◊ Generate JSON format, exportable copy of structured data.
◊ Display/extract SUM, AVERAGE, MAXIMUM and MINIMUM values for columns of structured data.
◊ CREATE STRUCTURE FIELDS FROM syntax to support record mapping from existing record-type definition.
◊ CREATE STRUCTURE STARTFIELD syntax to assist with ASG SmartFile XREF conversion.
◊ FSU and COMPLIB support for &DDname representing a concatenated library search path.

The remainder of this document contains details of these features and other changes introduced in SELCOPYi 3.40.

z/VM and z/VSE Systems

Improvements have been made to the zVM/CMS and z/VSE install procedures and accompanying install documentation. However, please be aware that features of SELCOPYi 3.40 detailed in this document are specific to **z/OS systems only**.

SELCOPY

The SELCOPY 3.40 (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 RI33001

Applicable to z/OS systems only, correct display of the DB2 version number for DB2 V10 and later in message CBLS000I of the SELCOPY DB2 message log (CBLSQLOG) output.

Previously displayed as 9.9.9.

Ref: APAR RI33002 (s330z01)

Applicable to z/OS systems only, correct possible GETMAIN abend that can occur when repeated OPEN/CLOSE is performed on a PDS or PDSE library, in order to re-read the library using READ DIRDATA (directory and member data input).

This same problem also existed in previous releases of SELCOPY.

Ref: APAR RI33002

Applicable to z/OS systems only, SELCOPY DB2 support for additional DB2 data types including:

◊ XML
◊ Large Objects (CLOB, BLOB)
◊ Decimal Floating Point (DECFLOAT)
◊ Timestamp with time zone (TIMESTMP WITH TIME ZONE)

Ref: APAR QI33002 (s330z02)

Applicable to z/VSE and z/OS systems, improved support for GSAM (RECFM=F/FB/V/VB only) input (GN/GU) and output (ISRT) via the SELCOPY/IMS interface.

Note: GSAM dataset I/O via direct READ/WRITE is already supported using standard access methods. Access via the SELCOPY/IMS interface is required if CHKP/XRST is to be performed and IMS is to recover the GSAM dataset location on the checkpoint restart.

SELCOPY/IMS I/O on GSAM is already possible with following caveats:

1. GN/GU sets the LRECL value at fixed length 89.
2. RECFM=V/VB GSAM input record includes the 2-byte length prefix.
3. RECFM=V/VB GSAM output record must first be prefixed by the 2-byte length.

This update addresses these issues so that:

1. LRECL is set to be the data length of the last GSAM input record.
2. The RECFM=V/VB 2-byte length suffix is not included in the record data.
3. The RECFM=V/VB 2-byte length suffix is automatically generated for ISRT to a GSAM RECFM=V/VB dataset using a value dictated by standard SELCOPY output.

Note also that GSAM GU direct input is possible by specifying the RSA location (or literal) as the argument on the SELCOPY/IMS input SSA= parameter.

Ref: APAR QI33003 (s330z03)

Correct an 0C4 program check that occurs on a backwards READ of a VSAM RRDS dataset when no STARTREC is specified.

Ref: APAR QI33004 (s330z04)

Applicable to z/OS systems only, correct an 0C4 program check that occurs when a DDname that has been allocated to a PDS or PDSE library member using SELCOPY dynamic allocation, is later allocated to a physical sequential dataset.

Note that a SELCOPY OPEN or CLOSE operation must be actioned on the DDname before it can be re-allocted to a different dataset name referenced by the DSN= parameter.

```
  READ CARD   INTO 1      WORKLEN=2048   * Raad DSN(+member) into position 1.

==LOOP==                                 * Label LOOP
  READ    INDD   DSN=54 AT 1  INTO 101   * If INDD is closed, dynalloc and open.
  IF  EOF INDD                           * If End-of-File
    THEN CLOSE INDD                      * Close the dataset allocated to INDD.
    ELSE PRINT        FROM 101           * Otherwise print the record...
    THEN GOTO LOOP                       *  and get the next record.

  GOTO GET                     * Get the next card (DSN) input record.
  END                          * End of control statements, start of CARD input.
NBJ.JCL(DISTGEN)
NBJ.CBLI.WINX
```

Ref: APAR QI33005 (s330z05)

Applicable to z/OS systems only, SELCOPY dynamic allocation will fail and ERROR 574 returned if a variable DSN= value is used which references a non-existant dataset. If CBLNAME option SDynallocFail=Empty is set, then a non-existant dataset will be treated as being empty with end-of-file immediately being triggered. This allows processing to continue following the failed dynamic allocation attempt with further, potential dynamic allocation permitted for the same DDname.

However, datasets which were subsequently allocated to a DDname flagged as being empty, were also treated as being empty. This update resets the DDname empty flag when a dataset is closed, so overcoming this problem.

SELCOPY (SLC) C++ Version

The SELCOPY (SLC) 3.40 C++ Version batch program has been updated with fixes to product defects and functions to support interactive debugging via the SELCOPYi SELCOPY IDE/Debug utility.

The 3.40 release of the SELCOPY Product Suite also introduces the first edition of the *SELCOPY C++ (SLC) Language Reference.*

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 3.40 batch program has undergone no changes since version 3.30 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.

Processing Changes

Only very minor changes have been applied to the SELCOPYi text editor in SELCOPYi 3.40. These are as follows:

EDIT Primary Command

The behaviour of text edit primary command EDIT, with no fileid specification, has been changed to perform the same operation as MDINEXT (EDIT) and MDIPREV (EDIT -).

Previously, EDIT and EDIT - (minus) would place focus on the next or previous text edit document view repectively within the Text Editor application window frame. This did not include any non-text edit document views (panels, lists, data edit views, etc.) that were also opened within the Text Editor frame window.

Performing MDINEXT and MDIPREV operations means that EDIT can navigate from a text edit view to a document window of any window class. Note that primary command WINDOW (assigned to F4 by default) or WINDOW - (minus) is the preferred method of navigating between document windows since WINDOW is supported by all classes of application windows.

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, JSON 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 datasets too large to be loaded into available storage and also has special provision for edit of VSAM data sets.

SELCOPYi structure definition objects (SDO) are in a format used only by SELCOPYi SDE applications and administer formatting on record data and DB2 table rows. SDO structures are comprised of one or more record-type definitions that are used to map data records/rows. An SDO and its record-type definitions can be created directly, using the CREATE STRUCTURE primary command, or automatically via EDIT and BROWSE operations involving non-SDO record mapping objects (e.g. COBOL/PL1 copybooks).

SELCOPYi 3.40 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.

Create Structure - COBOL/PL1 Copybook Parsing

For a COBOL or PL1 copybook source, SELCOPYi 3.40 supports its own parsing of the copybook input (COBOL Data Division data description entries and PL1 structure declarations) in order to build record-type definitions in an SDO. This method is an alternative to performing an Enterprise COBOL or PL1 compilation of the copybook source in order to generate SYSADATA output.

Use of the internal parser is optional and is determined by the prevailing value for the COPYBOOKPROC option (default INTERNAL) or by the COPYBOOKPROC option eplicitly specified on a CREATE STRUCTURE operation. The internal parser has the following advantages:

1. There is no requirement to have the IBM Enterprise COBOL Compiler or Enterprise PL1 Compiler installed and available in the load library search path.
2. Processing is quicker. A COBOL or PL1 compile is time consuming and requires a minimum TSO region size for successful operation (e.g. 128M for Enterprise COBOL 5.2).

Create Structure - COBOL Copybook Interpretation

For a BROWSE or EDIT operation that references a COBOL copybook or a CREATE STRUCTURE operation that uses the CopyBook Definition syntax, record-type definitions are generated from each 01 level group data description entry. e.g.

```
EDIT                USER.DATAFILE           USING COBOL USER.COBOL(XXSAMP01)

CREATE STRUCTURE    USER.SELC.SDO(XXSAMP01) FROM  COBOL USER.COBOL(XXSAMP01)
```

However, in previous releases, if each of the 2nd level entries following the level 01 entry are a union (REDEFINES) of each other, then a new record-type definition will be generated for each 2nd level entry instead of the level 01 entry.

This behaviour may now be suppressed using an environment option set by the **Auto Rec-Types** field of the Structured Data Edit Settings panel (=0.4). Setting the value in this field to "OFF" (the default) will ensure that record-type definitions are always generated from 01 levels.

The Auto Rec-Types field value is saved in the user's INI file as the value of SDE.RECTYPEAUTO and is maintained across each startup of SELCOPYi.

Create Structure - Subset Record-Type Definition

The CREATE STRUCTURE primary command syntax has been extended to allow a record-type definition to be generated using a subset of fields belonging to another record-type definition within the same SDO structure.

Some COBOL copybooks have a format where one or more data description entries are used to map common (header) fields located at the beginning of data records with redefined group entries used to map the record data that follows. The redefined group data description entry used to map the record data is selected based on values assigned to the common fields.

The following sample COBOL copybook member ('NBJ.COPYBOOK.COB(EXTP)') illustrates this format. It also uses a group entry (EXTP-TRAIL-DATA) to map trailing fields in the record data that follow the redefined group entries. The data records mapped by this copybook are of fixed length 318.

```
01  EXTP-RECORD.
    03  EXTP-HEAD-DATA.
        05  EXTP-SYSN                        PIC X(4).
        05  EXTP-PRIN                        PIC X(4).
        05  EXTP-AGNT                        PIC X(4).
        05  EXTP-RECN                        PIC X.
            88  CONTROL-REC                      VALUE '0'.
            88  WORK-REC                         VALUE '1'.
            88  COMP-REC                         VALUE '2'.

    03  EXTP-CONTROL-REC.
        05  FILLER                           PIC X(1).
        05  EXTP-XVC-ID                      PIC X(4).
        05  EXTP-RETAIN-TIME                 PIC 9(2).
        05  EXTP-ACTION-TIME                 PIC 9(2).
        05  FILLER                           PIC X(280).

    03  EXTP-WORK-REC    REDEFINES EXTP-CONTROL-REC.
        05  EXTP-WREC-ZTYP                   PIC X.
        05  EXTP-WREC-JENV                   PIC X.
        05  EXTP-WREC-ICNV                   PIC X.
        05  EXTP-WREC-BCNV                   PIC X.
        05  EXTP-WREC-XCNV                   PIC X.
        05  EXTP-WREC-JEN    OCCURS 4        PIC X(16).
        05  EXTP-WREC-ICN    OCCURS 4        PIC X(16).
        05  EXTP-WREC-BCN    OCCURS 4        PIC X(16).
        05  EXTP-WREC-XCN    OCCURS 4        PIC X(16).
        05  FILLER                           PIC X(28).

    03  EXTP-COMP-REC    REDEFINES EXTP-CONTROL-REC.
        05  EXTP-CREC-ZTYP                   PIC X.
        05  EXTP-CREC-FILLER-1               PIC X(2).
        05  EXTP-CREC-COMV                   PIC X.
        05  EXTP-CREC-FILLER-2               PIC X(69).
        05  EXTP-CREC-COMX   OCCURS 4        PIC X(16).
        05  EXTP-CREC-FILLER-3               PIC X(152).

    03  EXTP-TRAIL-DATA.
        05  EXTP-DATE                        PIC S9(9) COMP-3.
        05  EXTP-TIME                        PIC S9(7) COMP-3.
        05  EXTP-OVAL                        PIC X(3).
        05  EXTP-TERM                        PIC X(4).
```

Figure 1. COBOL Copybook NBJ.COPYBOOK.COB(EXTP).

Since the level 03 groups do not all redefine the same area of the record, the SDO structure created automatically by SELCOPYi will comprise a single record-type definition for the level 01 group. When Data Edit displays records using this SDO, all 3 redefined group definitions are included (i.e. the same area in a data record will be mapped 3 times and displayed in the data edit view).

Although this is not an error, it is desirable to only display the appropriate group mapping based on the value of the common EXTP-RECN header field. Using the new subset feature, an SDO can be created to achieve this. e.g.

```
CREATE STRUCTURE      NBJ.SELCOPYI.SDO(EXTP)     REPLACE
 RECORD(
        NAME          EXTP-RECORD
        SOURCE COBOL  NBJ.COPYBOOK.COB(EXTP)
        DEFAULT
       )
 RECORD(
        NAME          EXTP-CONTROL
        SOURCE FIELDS EXTP-HEAD-DATA(*),EXTP-CONTROL-REC(*),EXTP-TRAIL-DATA(*)
               FROM   EXTP-RECORD
        USE WHEN      CONTROL-REC           /* EXTP-RECN = 0 */
        PRIMARY
       )
 RECORD(
        NAME          EXTP-WORK
        SOURCE FIELDS EXTP-HEAD-DATA(*),EXTP-WORK-REC(*),EXTP-TRAIL-DATA(*)
               FROM   EXTP-RECORD
        USE WHEN      WORK-REC              /* EXTP-RECN = 1 */
        PRIMARY
       )
 RECORD(
        NAME          EXTP-COMP
        SOURCE FIELDS EXTP-HEAD-DATA(*),EXTP-COMP-REC(*),EXTP-TRAIL-DATA(*)
               FROM   EXTP-RECORD
        USE WHEN      COMP-REC              /* EXTP-RECN = 2 */
        PRIMARY
       )
```

See new RecordType definition parameter FIELDS under the description of the CREATE STRUCTURE primary command below.

Create Structure - Start Field Group Selection

The CREATE STRUCTURE primary command syntax has been extended to support specification of a starting field name. The group in which this field is defined identifies the start of the SDO record-type definition and the offset of the field within that group identifies the numerical offset value to be applied when mapping record data.

The start field specification is used in place of an explicit group name and offset value in the CREATE STRUCTURE operation. It may be used only within a RecordType definition and only when creating record-type definitions directly from a COBOL or PL1 copybook or a HLASM DSECT.

Using the same COBOL copybook source mamber ('NBJ.COPYBOOK.COB(EXTP)') as described in Figure 1. above, the following record-type definitions are equivalent:

```
RECORD(
       NAME           EXTP-RECORD             /* Group Name. */
       OFFSET         -13                     /* Mapping starts 13 bytes before record data. */
       SOURCE COBOL   NBJ.COPYBOOK.COB(EXTP)
       DEFAULT
      )

RECORD(
       STARTFIELD     EXTP-CONTROL-REC        /* Field at offset 13 in group EXTP-RECORD. */
       SOURCE COBOL   NBJ.COPYBOOK.COB(EXTP)
       DEFAULT
      )
```

By default, the group data name selected for the start field will be that belonging to the group with the highest level number in which the start field is defined. This may be overridden using the LEVEL paramter. e.g. The following will create a record-type definition for group EXTP-RECORD at level 01 instead of the default group EXTP-CONTROL-REC at level 03.

```
RECORD(
       STARTFIELD     EXTP-RETAIN-TIME        /* 05 level field. */
       LEVEL          01                      /* Select containing group at 01 level. */
       SOURCE COBOL   NBJ.COPYBOOK.COB(EXTP)
       DEFAULT
      )
```

See new RecordType definition parameter STARTFIELD under the description of the CREATE STRUCTURE primary command below.

Create Structure - PL1 Compiler Options Override

If required, SELCOPYi can now override installation default PL1 compiler options when creating SDO record-type definitions from a PL1 copybook source.

The Data Edit PL/1 Compiler Options panel (=0.4.2) contains a new input field in which to specify additional compiler options. Multiple, blank delimitted compile options may be entered in this field from which SELCOPYi will build a %PROCESS directive. Note that no ";" (semi-colon) should be specified at the end of the options as one will be inserted automatically by SELCOPYi. e.g. `MARGINS(1,72,0) LIMITS(FIXEDDEC(31))`

Create Structure - Datatypes VARHEX and HEXVAR

Data types VARHEX and HEXVAR are now supported by the Direct definition format of the CREATE STRUCTURE operation. Direct definition format allows creation of a record-type using individual field definition syntax and without the pre-requisite of an external copybook.

Like HEX, both VARHEX and HEXVAR data types display data as a printable hexadecimal character string, where each pair of hexadecimal digits that constitute a byte is represented as 2 printable characters. e.g. F1C1C4 would be displayed for EBCDIC characters "1AD".

Record data defined as being data type VARHEX is interpreted as a variable length binary string which is immediately preceded by an integer binary field containing a length value. Parenthesised sub-script parameters define the maximum data length, length of the integer binary length field and whether the length field is included or excluded in the data length value. e.g. `VARHEX(24,4,INCUSIVE)`

Record data defined as being data type HEXVAR is interpreted as being a variable length binary string padded with blanks to occupy an area of fixed length equal to the defined maximum possible length of the binary string. The binary string data is immediately preceded by a 2-byte integer binary field defining the length of the variable binary string data. Therefore, the actual length of data mapped by a HEXVAR field is always the defined maximum length plus 2. A parenthesised sub-script parameter defines the maximum binary string data length. e.g. `HEXVAR(64)`

VARHEX is analogous to the DB2 VARBIN data type. VARHEX or HEXVAR should be used to display data that is of variable length but may not include printable EBCDIC character text.

See new Direct definition parameters VARHEX and HEXVAR under the description of the CREATE STRUCTURE primary command below.

Create Structure - Browse and Edit Options

When used to display formatted record data for BROWSE or EDIT in a Data Edit view, an SDO structure may be created so that, in addition to the data formatting, it also specifies certain display options. Specifically these are as follow:

Colour Highlighting

When displaying formatted record data in a Data Edit view, it may be useful to have SELCOPYi automatically colour specific columns or lines where a value or combination of values in a line satisfies an expression. For example, you may wish to colour highlight entries in an accounts balance column that have a negative (debit) value.

This functionality is possible via execution of the SET CCOLOUR and SET RCOLOUR options once the Data Edit view is opened. However, it is now also possible to define colour highlighting behaviour via the CREATE STRUCTURE primary command.

Column Widths

By default, DB2 table BROWSE will display columns of data using the shortest width possible without truncating column values. However, EDIT and non-DB2 BROWSE columns are displayed using their maximum defined width.

Previously, these defaults could only be overridden using the SET COLWIDTH option once the display was opened. SELCOPYi 3.40 extends support so that display column widths may also be established via the CREATE STRUCTURE primary command.

In addition to these display options, an SDO structure can specify one or more SDE primary commands that will be executed for data in the Data Edit view even before the window view is displayed. Furthemore, these commands will be executed before any SDE commands passed via an INITCMD specification on the BROWSE or EDIT primary command.

To support these features, the CREATE STRUCTURE primary command syntax now includes the following parameters:

- **OPTION** which may be repeated to specify any number of parenthesised CCOLOUR, RCOLOUR or COLWIDTH specifications.
- **INITCMD** followed by a parenthesised list of semi-colon separated SDE commands.

See new parameters OPTION and INITCMD under the description of the CREATE STRUCTURE primary command below.

Performance and Storage Use

Changes have have been implemented that significantly improve Data Editor performance and storage use in the following areas of operation:

1. Calculation of available storage and the algorythms used to determine whether a Full or In-Place data edit will attempt to load all selected records of a dataset into storage (i.e. in-storage edit).
2. Vertical scrolling through the display of a dataset containing a large number of records and for which a large number of line exclusions have occurred. Note that scrolling is also performed internally by SELCOPYi in order to determine limits on a range of lines affected by an operation. e.g. `DELETE 1000 NX`

   For edit of large datasets and browse operations, groups of records are loaded into storage by SELCOPYi and unloaded as necessary. This process has been enhanced so that groups of excluded records are never loaded unnecessarily.
3. Execution of EXCLUDE operations that exclude lines to the end of the display of records.

DB2 Table Edit/Browse Processing Enhancements

The SELCOPYi DB2 table edit/browse feature has received major new enhancements in SELCOPYi 3.40.

XML Column Data Processing

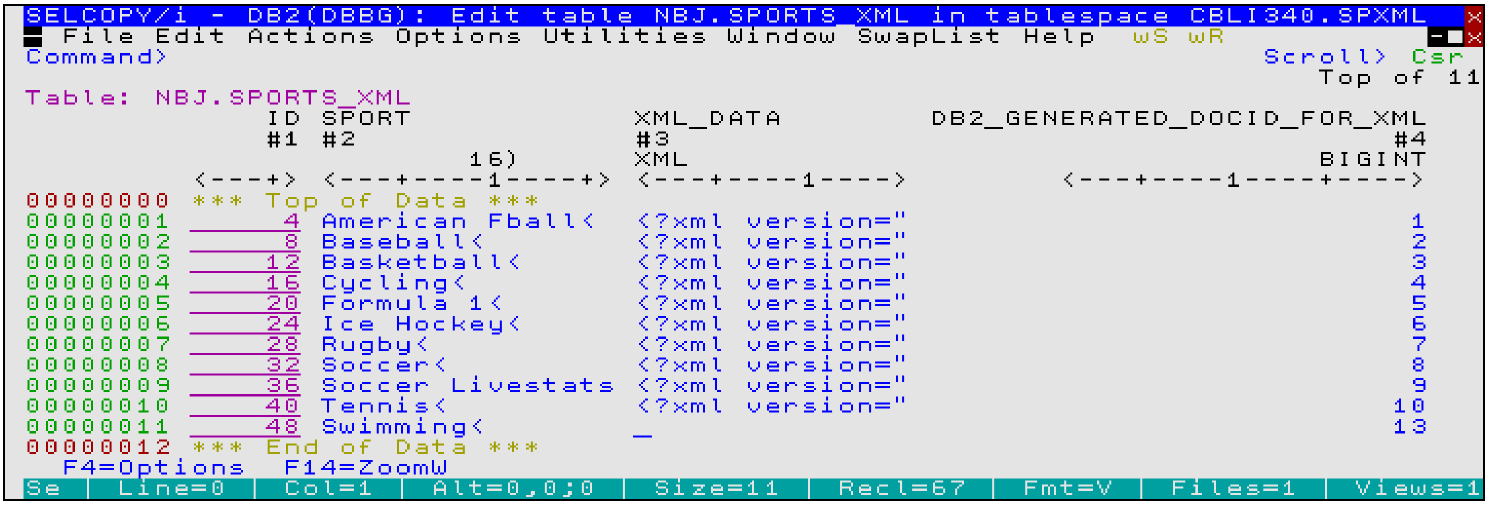
SELCOPYi Data Edit of a DB2 result table now correctly supports XML table columns.

Previous releases of SELCOPYi treat XML columns as datatype VARCHAR with maximum length 32000. When the length of XML document text exceeded this maximum length, the following DB2 error was returned:

```
DSNT408I SQLCODE = -433, ERROR:  VALUE <XML-Value> IS TOO LONG
```

Browse and Edit of a DB2 table will display a fixed number of characters from the start of the XML document as specified by the BROWSE or EDIT XMLLOBWIDTH parameter (default is 0 characters). For example, the following command will display only the first 15 characters of each XML document entry in column XML_DATA.

```
SDATA   EDIT  DB2(DBBG)   NBJ.SPORTS_XML    XMLLOBWIDTH 15
```

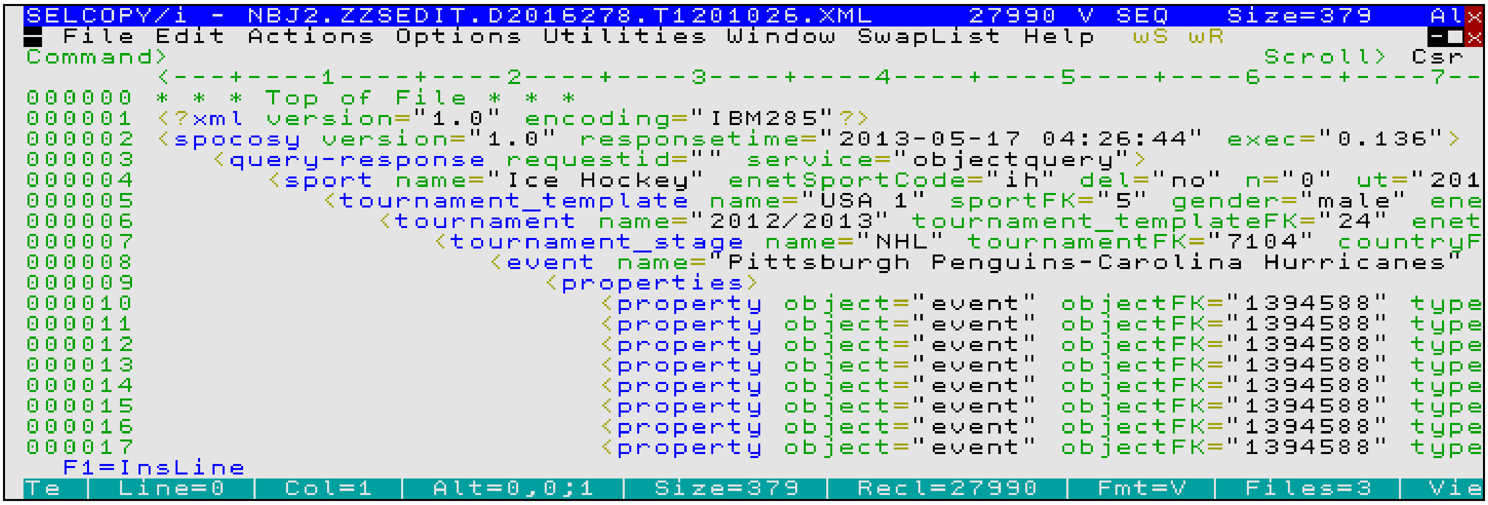


```
SELCOPY/i - DB2(DBBG): Edit table NBJ.SPORTS_XML in tablespace CBLI340.SPXML
 File Edit Actions Options Utilities Window SwapList Help  wS wR
Command>                                                           Scroll> Csr
                                                                      Top of 11
Table: NBJ.SPORTS_XML
             ID SPORT            XML_DATA        DB2_GENERATED_DOCID_FOR_XML
             #1 #2               #3                                       #4
                        16)      XML                                  BIGINT
         <---+> <---+----1----+> <---+----1---->        <---+----1----+---->
00000000 *** Top of Data ***
00000001      4 American Fball<  <?xml version="                           1
00000002      8 Baseball<        <?xml version="                           2
00000003     12 Basketball<      <?xml version="                           3
00000004     16 Cycling<         <?xml version="                           4
00000005     20 Formula 1<       <?xml version="                           5
00000006     24 Ice Hockey<      <?xml version="                           6
00000007     28 Rugby<           <?xml version="                           7
00000008     32 Soccer<          <?xml version="                           8
00000009     36 Soccer Livestats <?xml version="                           9
00000010     40 Tennis<          <?xml version="                          10
00000011     48 Swimming<        _                                        13
00000012 *** End of Data ***
  F4=Options  F14=ZoomW
Se | Line=0 | Col=1 | Alt=0,0;0 | Size=11 | Recl=67 | Fmt=V | Files=1 | Views=1
```

Figure 2. DB2: Table Edit with XML column.

Thereafter, the complete XML text belonging to an individual XML column entry may be browsed, viewed or edited using new primary commands XMLBROWSE, XMLVIEW or XMLEDIT respectively. For edit and view, the text is displayed in a Text Edit view. For the NBJ.SPORTS_XML table above, the following will open a text edit view to edit the XML document in the XML_DATA column (referenced as #3) in row 6 (the "Ice Hockey" row entry).

```
XMLEDIT  #3  6
```



```
SELCOPY/i - NBJ2.ZZSEDIT.D2016278.T1201026.XML     27990 V SEQ   Size=379   Al
 File Edit Actions Options Utilities Window SwapList Help  wS wR
Command>                                                           Scroll> Csr
       <---+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000000 * * * Top of File * * *
000001 <?xml version="1.0" encoding="IBM285"?>
000002 <spocosy version="1.0" responsetime="2013-05-17 04:26:44" exec="0.136">
000003    <query-response requestid="" service="objectquery">
000004       <sport name="Ice Hockey" enetSportCode="ih" del="no" n="0" ut="201
000005          <tournament_template name="USA 1" sportFK="5" gender="male" ene
000006             <tournament name="2012/2013" tournament_templateFK="24" enet
000007                <tournament_stage name="NHL" tournamentFK="7104" countryF
000008                   <event name="Pittsburgh Penguins-Carolina Hurricanes"
000009                      <properties>
000010                         <property object="event" objectFK="1394588" type
000011                         <property object="event" objectFK="1394588" type
000012                         <property object="event" objectFK="1394588" type
000013                         <property object="event" objectFK="1394588" type
000014                         <property object="event" objectFK="1394588" type
000015                         <property object="event" objectFK="1394588" type
000016                         <property object="event" objectFK="1394588" type
000017                         <property object="event" objectFK="1394588" type
  F1=InsLine
Te | Line=0 | Col=1 | Alt=0,0;1 | Size=379 | Recl=27990 | Fmt=V | Files=3 | Vie
```

Figure 3. DB2: XML column text edit.

For XMLEDIT and XMLVIEW only, the text is formatted and displayed in the text edit view so that a new line is started after each tag element not followed by text data, and after each end tag. Nested tag elements are also indented to illustrate the tag hierarchy.

The text editor view containing the XML data is assigned a temporary dataset name. When edited updates to the XML document are saved, SELCOPYi does not attempt to write the data to DASD using the assigned DSN as it would for standard text edit. Instead, the count of alterations made since the last save is reset to zero and, when the text edit view is subsequently closed, the corresponding DB2 row in the data edit view is flagged as changed. When save is executed in the DB2 table edit view, a DB2 SQL UPDATE statement is executed which parses the updated XML document and verifies that it is well-formed before the text is saved to the DB2 data base.

See descriptions of new Data Edit primary commands XMLBROWSE, XMLEDIT and XMLVIEW below.

General utility commands GETXML and PUTXML have also been introduced to unload and load an XML document. Using these utilities XML document text can be transferred between a DB2 XML column entry and a dataset, library member or HFS file.

See descriptions of new SELCOPYi DB2 utility primary commands GETXML and PUTXML below.

DB2 Scrollable Cursors

By default, SELCOPYi DB2 table browse and edit will attempt to load all eligible rows of a DB2 result table into virtual storage at the start of the edit or browse session.

Therefore, attempting to browse or edit a very large DB2 result table may result in storage error ZZSD016E and failure to display the table contents. To overcome this, the number of rows loaded into virtual storage must be limited by restricting the size of the result table. This can be done using one or both of the following:

1. Using the **FROM** and **FOR** parameters. For example:

   ```
   BROWSE  DB2(CBLA)   SYSIBM.SYSTABLES     FROM 21    FOR 100
   ```

   will open a Data Editor browse view and load 100 rows of the table **SYSIBM.SYSTABLES** starting at the 21st row.

2. Using the **WHERE** parameter which specifies a standard SQL **WHERE** clause in parentheses. For example:

   ```
   BROWSE  DB2(CBLA)   SYSIBM.SYSTABLES    WHERE( CREATEDBY='NBJ' )
   ```

   will open a Data Editor browse view and load all rows of the table **SYSIBM.SYSTABLES** for which the **CREATEDBY** column is equal to the string **NBJ**.

   Note that the argument to the **WHERE** parameter can by given as **?** which will invoke the **DB2 row selection dialog** which can be used to generate a **WHERE** clause without having to type the column names. So for example:

   ```
   BROWSE  DB2(CBLA)   SYSIBM.SYSTABLES    WHERE(?)
   ```

   will first run the row selection dialog so that a **WHERE** clause can be generated to limit the rows loaded before starting the browse operation.

SELCOPYi 3.40 introduces support for DB2 scrollable cursors which may be used as an alternative to the above methods when handling a large result table. Use of scrollable cursors mean that a relatively small number of rows is kept in storage at any time.

Note that, when a result table is opened using a static scrollable cursor, DB2 makes a materialised (temporary) copy of the table in the work file database. This means that, for large tables, the process of opening the cursor can consume much resource and cause a noticeable delay in the time taken to display the data edit view.

Before DB2 scrollable cursors can be used, the feature must be enabled. To do this an authorised SELCOPYi administrator must update the SELCOPYi system INI dataset so that the value for option variable **DB2.SCROLL** is **YES**. Thereafter, use of a scrollabe cursor is triggered by specifying the **SCROLL** keyword parameter on the EDIT or BROWSE command.

SELCOPYi uses a scrollabe **SENSITIVE STATIC** cursor for EDIT and a scrollable **INSENSITIVE** cursor for BROWSE. With either of these types of cursor the results table has a fixed size determined when the cursor is opened.

Once the cursor has been opened the results table functions rather like a fixed sized VSAM RRDS. It contains a fixed number of rows each of which may be directly referenced by row number. The edit mode is **update-in-place**, so rows cannot be inserted or moved. Deletes are allowed, although the deleted row is not removed from the display, becoming instead a form of shadow line.

The fact that rows can be retrieved directly by row number means that direct access to the middle or end of the table is relatively fast because intermediate rows do not have to be read.

The size of the results table is not known until the last row has been read and the table size in the status bar is prefixed by a greater than sign to indicate this. Scrolling to the end of the table (with command **DOWN MAX** for example) will establish the actual table size.

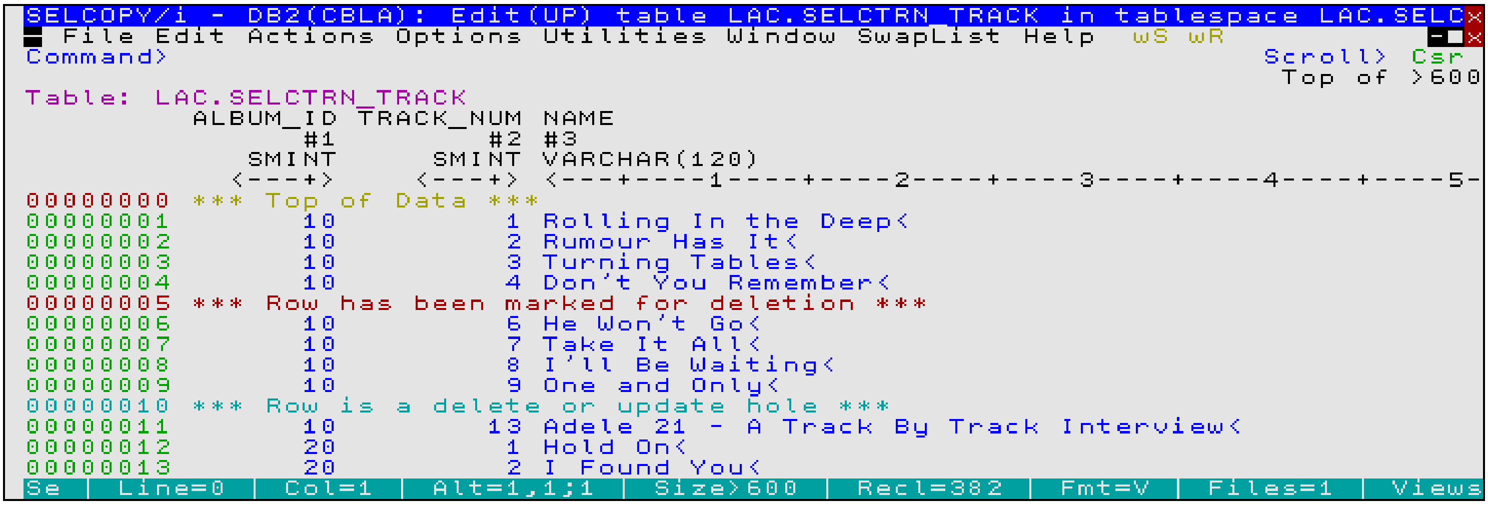
As for **non-scrollable** cursors, a WHERE clause can be used to limit the size of the results table. However, FOR and FROM parameter keywords cannot be used with SCROLL.

For browse the use of an INSENSITIVE cursor means that the results table does not change even if other processes are deleting or updating rows in the base table. For edit the use of a SENSITIVE STATIC cursor means that when another process either deletes a row or updates a row so that it no longer satisfies the WHERE clause then such a row becomes a **cursor hole** and can no longer be fetched. Cursor holes are represented by a form of shadow line.

In the following example db2 table **LAC.SELCTRN_TRACK** is being edited with a scrollable, sensitive static cursor.

- **Edit(UP)** in the panel title bar indicates that the edit mode is **update-in-place** so that rows cannot be moved or inserted.
- **Top of >600** at the top right of the panel indicates that 600 rows of the table have been fetched but the end of the table has not yet been reached.
- Row 5 has been deleted (using the **D** prefix command) and the *** **Row has been marked for deletion** *** message indicates the change has not yet been saved.
- Row 10 has been deleted or updated by another process and having been re-fetched was found to be a cursor hole as indicated by the *** **Row is a delete or update hole** *** message.

  The row was re-fetched because only a subset of table rows are kept in storage and the user has scrolled away from the top of the table and then scrolled back causing the first part of the table to have been removed from storage and then reloaded.



```
SELCOPY/i - DB2(CBLA): Edit(UP) table LAC.SELCTRN_TRACK in tablespace LAC.SELC
  File Edit Actions Options Utilities Window SwapList Help   wS wR
Command>                                                                Scroll> Csr
                                                                         Top of >600
Table: LAC.SELCTRN_TRACK
          ALBUM_ID TRACK_NUM NAME
                #1        #2 #3
             SMINT     SMINT VARCHAR(120)
            <---+>    <---+> <---+----1----+----2----+----3----+----4----+----5-
00000000 *** Top of Data ***
00000001        10         1 Rolling In the Deep<
00000002        10         2 Rumour Has It<
00000003        10         3 Turning Tables<
00000004        10         4 Don't You Remember<
00000005 *** Row has been marked for deletion ***
00000006        10         6 He Won't Go<
00000007        10         7 Take It All<
00000008        10         8 I'll Be Waiting<
00000009        10         9 One and Only<
00000010 *** Row is a delete or update hole ***
00000011        10        13 Adele 21 - A Track By Track Interview<
00000012        20         1 Hold On<
00000013        20         2 I Found You<
Se | Line=0 | Col=1 | Alt=1,1;1 | Size>600 | Recl=382 | Fmt=V | Files=1 | Views
```

Figure 4. DB2: Scrollable cursor edit.

DB2 Version 11 - TIMESTAMP Datatypes

Support for extensions to the TIMESTAMP datatype introduced in DB2 v11 has been included in SELCOPYi 3.40. Specifically, this involves interpretation of the following:

TIMESTAMP(*p*)

A timestamp where *p* is an integer specifying the optional precision attribute in the range from 0 to 12 and denotes the number of fractional second digits that are included in the timestamp.
The default is 6.

TIMESTAMP(*p*) **WITH TIME ZONE**

A UTC timestamp, with optional precision as described above, and a time zone adjustment in the format +hh:mm or -hh:mm (hh => hour displacement from UTC, mm => minute displacement.)

Data Editor - Panels

The following Data Editor specific panels have been included in SELCOPYi 3.40.

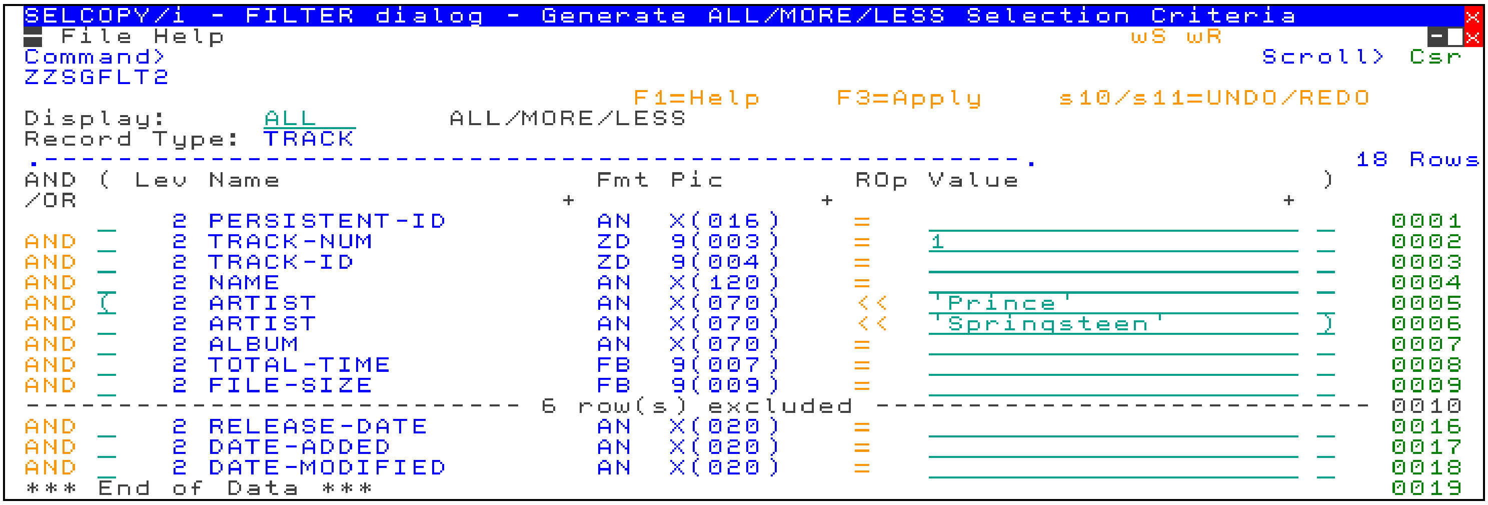
SDE FILTER Records Panel

SELCOPYi 3.40 introduces the SDE FILTER Records panel to filter the display of loaded records (table rows) that are assigned the default record type.

The panel will dynamically generate or execute an SDE WHERE, MORE or LESS primary command that specifies record selection criteria using an SDE expression.

The SDE Filter Records panel (ZZSGFLT2) is an interactive panel window, opened on executing either of the following:

1. Primary command **FILTER** with no parameters (in which case the default record type is used).
2. Line (prefix) command *FLT* in the prefix area of a record of the required record type.



```
SELCOPY/i - FILTER dialog - Generate ALL/MORE/LESS Selection Criteria
  File Help                                                 wS wR
Command>                                                           Scroll> Csr
ZZSGFLT2
                                 F1=Help    F3=Apply    s10/s11=UNDO/REDO
Display:     ALL       ALL/MORE/LESS
Record Type: TRACK
.-----------------------------------------------------.                 18 Rows
AND ( Lev Name                 Fmt Pic       ROp Value                 )
/OR                          +             +                        +
    _   2 PERSISTENT-ID        AN  X(016)    =                         _   0001
AND _   2 TRACK-NUM            ZD  9(003)    =   1                     _   0002
AND _   2 TRACK-ID             ZD  9(004)    =                         _   0003
AND _   2 NAME                 AN  X(120)    =                         _   0004
AND (   2 ARTIST               AN  X(070)    <<  'Prince'              _   0005
AND _   2 ARTIST               AN  X(070)    <<  'Springsteen'         )   0006
AND _   2 ALBUM                AN  X(070)    =                         _   0007
AND _   2 TOTAL-TIME           FB  9(007)    =                         _   0008
AND _   2 FILE-SIZE            FB  9(009)    =                         _   0009
--------------------------- 6 row(s) excluded ---------------------------- 0010
AND _   2 RELEASE-DATE         AN  X(020)    =                         _   0016
AND _   2 DATE-ADDED           AN  X(020)    =                         _   0017
AND _   2 DATE-MODIFIED        AN  X(020)    =                         _   0018
*** End of Data ***                                                        0019
```

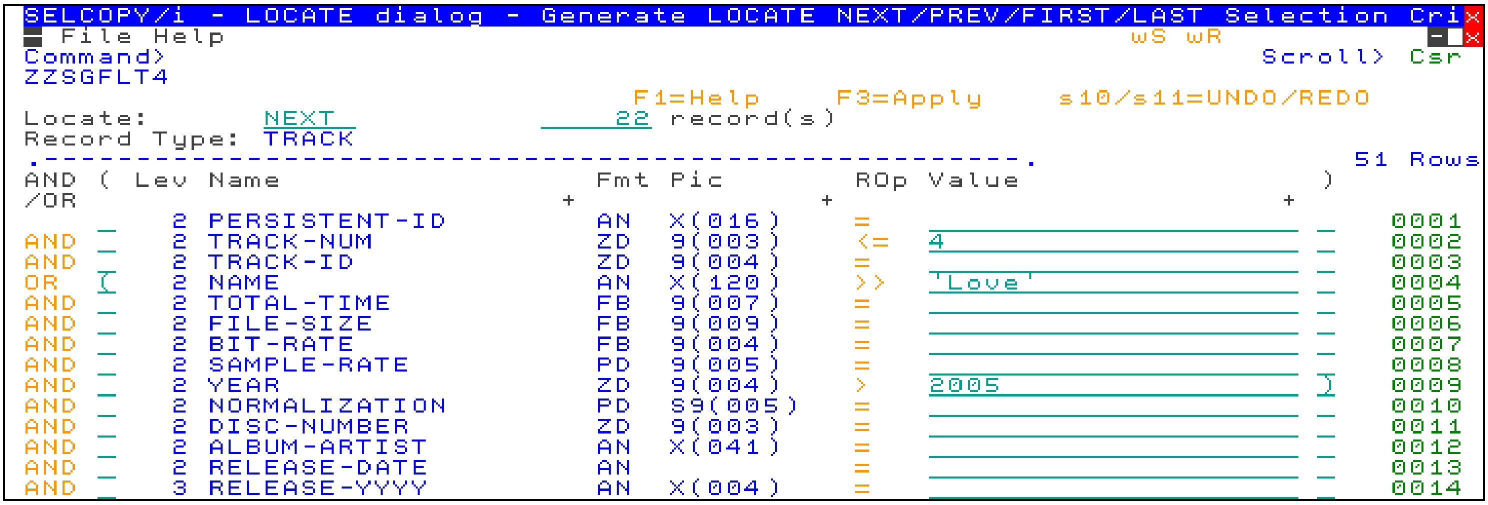
Figure xx. SDE Filter Records Panel.

SDE LOCATE Records Panel

SELCOPYi 3.40 introduces the SDE LOCATE Records panel to locate and, if excluded, display loaded record(s) that are assigned the default record type and match the specified selection criteria.

The panel will dynamically generate or execute an SDE LOCATE primary command that specifies record selection criteria using an SDE expression.

The SDE Locate Records panel (ZZSGFLT4) is an interactive panel window, opened when primary command **LOCATE** is executed with no parameters. The LOCATE operation will operate on the default record type.



```
SELCOPY/i - LOCATE dialog - Generate LOCATE NEXT/PREV/FIRST/LAST Selection Cri
  File Help                                                 wS wR
Command>                                                           Scroll> Csr
ZZSGFLT4
                                 F1=Help    F3=Apply    s10/s11=UNDO/REDO
Locate:      NEXT                  22 record(s)
Record Type: TRACK
.-----------------------------------------------------.                 51 Rows
AND ( Lev Name                 Fmt Pic       ROp Value                 )
/OR                          +             +                        +
    _   2 PERSISTENT-ID        AN  X(016)    =                         _   0001
AND _   2 TRACK-NUM            ZD  9(003)    <=  4                     _   0002
AND _   2 TRACK-ID             ZD  9(004)    =                         _   0003
OR  (   2 NAME                 AN  X(120)    >>  'Love'                _   0004
AND _   2 TOTAL-TIME           FB  9(007)    =                         _   0005
AND _   2 FILE-SIZE            FB  9(009)    =                         _   0006
AND _   2 BIT-RATE             FB  9(004)    =                         _   0007
AND _   2 SAMPLE-RATE          PD  9(005)    =                         _   0008
AND _   2 YEAR                 ZD  9(004)    >   2005                  )   0009
AND _   2 NORMALIZATION        PD  S9(005)   =                         _   0010
AND _   2 DISC-NUMBER          ZD  9(003)    =                         _   0011
AND _   2 ALBUM-ARTIST         AN  X(041)    =                         _   0012
AND _   2 RELEASE-DATE         AN            =                         _   0013
AND _   3 RELEASE-YYYY         AN  X(004)    =                         _   0014
```

Figure xx. SDE Locate Records Panel.

Data Editor - Primary Commands

The following Data Editor primary commands have been updated or included in SELCOPYi 3.40.

AVERAGE

Syntax:

```
                 +------------------------- , --------------------------+
                 v                                                      |
>>-+- AVerage -+-+- field_col ------------------------------------------+->
   |           | |                                                      |
   +- AVG -----+ |                                      +- CHaracter -+ |
                 |                                      |             | |
                 +- field_pos -- , - field_length -- , -+-------------+-+
                                                        |             |
                                                        +- Fixed -----+
                                                        +- Binary ----+
                                                        +- DECimal ---+
                                                        +- Zoned -----+
                                                        +- HFP -------+
                                                        +- BFP -------+
                                                        +- DFP -------+
                               +- .ZFIRST -- .ZLAST -+
                               |                     |
 >--+- WHere -- expression -+--+---------------------+--+---------------+->
    |                       |  |                     |  |               |
    |                       |  +- .name1 -+----------+  +- DIgits int --+
    | +- NX -----------+    |             |          |
    | +- NOTEXcluded --+    |             +- .name2 -+
    | |                |    |
    +-+----------------+----+
      |                |
      +- EXcluded -----+
      +- X ------------+
      |                |
      +- ALL ----------+

 >--+--------------------------------------------+--+----------------+----><
    |                                            |  |                |
    +-------+- RECord -+--------+- record_type --+  +- STEM stemvar -+
    |       |          |        |
    +- FOR -+          +- TYPE -+
```

Description:

New in SELCOPYi 3.40, AVERAGE will display or extract the average value for values belonging to columns of data in unformatted records or, if formatting is applied, specific columns in records of a specific record type.

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) -+
DB2 Opts:

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

 >--+-----------------------------------------------------------------+--->
    |                                                                 |
    +-- SCROLL -------------------------------------------------------+
    |                                                                 |
    +------------------------------+--+-------------------------------+
    |                              |  |                               |
    +-- FROM --+----------+- n ----+  +-- FOR -- n_rows --+-----------+
               +- ROW ----+                               +- ROWS ----+

                                          +-- XMLLOBWidth -- 0 --------+
                                          |                            |
 >--+---------------------------------+---+----------------------------+--|
    |                                 |   |                            |
    +---- SELect (select_syntax) -----+   +-- XMLLOBWidth -- n_bytes --+
```

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

- New **SCROLL** keyword parameter to use a DB2 scrollable INSENSITIVE cursor to fetch DB2 result table rows.
- New **XMLLOBWIDTH** keyword parameter to specify the number of bytes (*n_bytes*) of text, at the start of an XML column, to be displayed for all XML columns in the DB2 table view.

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 | --------+

 >--+--------------------------------------------------------------+------>
    |                                                              |
    |                  +- NOTHING ---------------------------+     |
    |                  |                                     |     |
    +- REPLACING - ( - +-------------------------------------+- ) -+
                       |          +------------------------+ |
                       |          v                        | |
                       +--------+-+- text_1 - BY - text_2 -+-+
                       |        |
                       +- ONLY -+

 >--+--------------------------------------------+------------------------>
    |                                            |
    +- SYSADATA - ( - sysadata_dataset_name - ) -+

 >--+---------------------------------------+----------------------------->
    |                                       |
    +- COPYBOOKProc - ( -+- COMPILER -+- ) -+
                         |            |
                         +- INTERNAL -+

 >--+----------------------------------------------------+---------------->
    |                                                    |
    | +--------------------------------------------+     |
    | v                                            |     |
    +-+- OPTion - ( -+- | COLWIDTH command | -+- )-+-----+
                     |                        |
                     +- | CCOLOUR  command | -+
                     |                        |
                     +- | RCOLOUR  command | -+

 >--+--------------------------------------+------------------------------><
    |                                      |
    |             +------------------+     |
    |             v                  |     |
    +- INITCMD ( -+- | sd_command | -+- ) -+
```

```
XRef Definition:

 |----+--------------------------------------------------+---------------->
      |                +--------------------+            |
      |                v                    |            |
      +- LIBrary -- (---- copybook_library -+- ) --------+

    +------------------------------------------------------+
    v                                                      |
 >------ RECord  -- ( -- | RecordType Definition | -- ) ---+--------------|
```

RecordType Definition:

```
|-+- NAME ------ record_type --+------------------------------------+--+->
  |                            |                                    |  |
  |                            |          +- + (plus) --+           |  |
  |                            |          |             |           |  |
  |                            +- OFFset -+-------------+- n_bytes -+  |
  |                                       |             |              |
  |                                       +- - (minus) -+              |
  |                                                                    |
  +- STARTField -- data_element ---------------------------------------+
    (1)
```

```
              +- COBol -+                       +- LEVEL 1 ---------+
              |         |                       |                   |
>-- SOURCE -+-+---------+--- source_fileid -----+-------------------+--+-|
            | |         |                       |                   |  |
            | +- PL1 ---+                       +- LEVel level_num -+  |
            |                                                          |
            +--- HLAsm ----- source_fileid ----------------------------+
            |                                                          |
            +--- ADAta ------ adata_fileid ----------------------------+
            |                                                          |
            +--- SDO ---------- sdo_fileid ----------------------------+
            |                                                          |
            +--- STRUCTure (-- | Record Definition | --) --------------+
            |                                                          |
            |                +------------------+                      |
            |                v                  |                      |
            +--- SYMNAMes  (-+- sym_fileid -----+-) -------------------+
            |                |                  |                      |
            |                +- sym_namedef ----+                      |
            |                                                          |
            |                +------ , ---------+                      |
            |                v                  |                      |
            +--- FIELDS -----+- data_element ---+-- FROM record_type --+
```

Direct Definition:

```
      +---------------------------- , -----------------------------+
      v                                                            |
|- ( -+--| Record Definition |---+------------------------------+--+- ) -|
                                 |                              |
                                 +-+-------+- WHEN expression --+
                                   |       |
                                   +- USE -+
```

Record Definition:

```
                                +--------- , ----------+      + EBCdic +
                                v                      |      |        |
|- record_type -- STRUCTure - ( --| Field Definition |-+- ) --+--------+-|
                                                              |        |
                                                              + ASCii -+
```

Field Definition:

```
|--+--------------+---| Data Definition |--------------------------------|
   |              |
   +- field_name -+
```

Data Definition:

```
|-+- BINTeger -+--------------------------------------------------+---+--|
  |            |                                                  |   |
  |            +- ( n_bits ) -------------------------------------+   |
  |                                                                   |
  +- ... -------------------------------------------------------------+
  |                                                                   |
  +- HEXVAR ---+------------------------------------------------------+
  |            |                                                  |   |
  |            +- ( n_bytes ) ------------------------------------+   |
  |                                                                   |
  +- ... -------------------------------------------------------------+
  |                                                                   |
  +- VARHEX ---+------------------------------------------------------+
               |                                                      |
               |               +- ,2 ---------+  +- ,EXCLUSIVE -+     |
               |               |              |  |              |     |
               +- ( max_bytes -+--------------+--+--------------+- ) -+
                               |              |  |              |
                               +- ,len_bytes -+  +- ,INCLUSIVE -+
```

Notes:

1. STARTFIELD may be used only with SOURCE COBOL, PL1 or HLASM.

Existing primary command CREATE STRUCTURE has been enhanced to support the following:

- New **REPLACING** keyword parameter and arguments that specify text replacing options to be performed when processing COBOL copybooks.

  By default, all pseudo text replacements that have been defined in the SELCOPYi system INI dataset and in the user's COBOL Compiler Options panel (=0.4.1) are applied to the COBOL copybook source. Additional pseudo text replacement values may be specified on this REPLACING parameter.

  Alternatively, the following keyword arguments may be specified:

  - **NOTHING** suppresses all COBOL pseudo-text replacement.
  - **ONLY** restricts COBOL pseudo-text replacement to that specified on the REPLACING parameter. i.e. System INI and user defined replacements are ignored.

- New **SYSADATA** keyword parameter that specifies the name of the SYSADATA dataset to be created when a COBOL, PL1 or HLASM compile/assembly is to be performed to create SDO record-type definitions.

  By default, the SYSADATA dataset is a dynamically allocated temporary dataset which is deleted when the CREATE STRUCTURE command has been processed.

  There may be a requirement to examine or save this SYSADATA file (for example for problem diagnosis). This option allows the specification of a user allocated permanent dataset or PDS(E) member which will be used instead of the temporary dataset.

  The SYSADATA dataset or PDS(E) member must have record format variable blocked (RECFM=VB) and logical record length greater than 1000 (LRECL>1000).

- New **COPYBOOKPROC** keyword parameter and keyword arguments **COMPILER** or **INTERNAL** specify whether the COBOL language compiler or SELCOPYi's internal code is used to process copybooks for this command.

  By default, the current setting of the **COPYBOOKPROC** option determines the copybook processor.

- New **OPTION** keyword parameter may be repeated to specify separate edit and browse display options that can be saved in the SDO structure. These options are normally set interactively using a data edit primary command but can be pre-defined when the structure is created using the OPTION keyword.

  The OPTION argument is one instance of a parenthesised SET COLWIDTH, CCOLOUR or RCOLOUR primary command that will set a column width, colour a column entry or colour a row of data respectively.

- New **INITCMD** keyword parameter specifies one or more data edit primary commands that are to be executed whenever a browse or edit session is started using this structure. These commands are executed before any INITCMD specifications on the BROWSE or EDIT operations.

  Any number of commands may be listed in parentheses following the INITCMD keyword, with each command enclosed in either apostrophes or quotation marks.

- New **STARTFIELD** keyword parameter, supported as part of RecordType Definition syntax, specifies the name of a field data element within the source COBOL, PL1 or HLASM copybook that belongs to the group (structure) element used to define the record-type. STARTFIELD also implies an offset into the owning group at which record data will be mapped.

  STARTFIELD may be used as alternative to NAME and OFFSET parameters. The group element used is, by default, the group with the highest level value in which the start field is defined. The LEVEL parameter may be used to override this default.

- New **FIELDS** keyword parameter, comma separated data elements argument list and FROM record-type arguments is now supported as part of RecordType Definition syntax. This syntax defines a new record-type that is a clone of another record-type already defined within the same structure, but includes only those data elements (fields) that have been selected.

  In the list of selected fields (data elements), all fields within a group (structure) may be selected by specifying a "(*)" suffix on the field data element name. For example, XREF_KEY(*) selects all fields in the group field XREF_KEY.

- New **HEXVAR** data type keyword parameter with a parenthesised length argument (*n_bytes*) is supported as part of the Data Definition syntax.

  HEXVAR defines a field to be interpreted as a variable-length printable hexadecimal string padded with blanks to occupy a fixed *n_bytes*, where *n_bytes* defaults to 1. The actual length of the string is defined by a 2-byte binary integer at the start of the field so that the overall length of the field is 2+*n_bytes*.

- New **VARHEX** data type keyword parameter with parenthesised length and keyword arguments (*max_bytes*, *len_bytes*, INCLUSIVE or EXCLUSIVE) are supported as part of the Data Definition syntax.

  VARHEX defines a field to be interpreted as a variable-length printable hexadecimal string. The maximum possible length of the string is defined as integer value *max_bytes* (default 0). The actual length of the string is defined by a binary integer field of length *len_bytes* (default 2) at the start of the field. Option specification of INCLUSIVE or EXCLUSIVE (default) indicates whether *len_bytes* is included or excluded respecively in the the value of *max_bytes*.

EDIT

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

Syntax:

```
>>- Edit -+-------------- fileid ---------------------| SDE Opts |--+-----><
          |                                                         |
          +-- DB2 --+---------+--+-- table_name --+---| DB2 Opts |--+
                    |         |  |                |
                    +- (ssn) -+  +-- view_name ---+

DB2 Opts:

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

 >--+-----------------------------------------------------------------+--->
    |                                                                 |
    +-- SCROLL -------------------------------------------------------+
    |                                                                 |
    +------------------------------+--+-------------------------------+
    |                              |  |                               |
    +-- FROM --+----------+- n ----+  +-- FOR -- n_rows --+-----------+
               |          |                               |           |
               +- ROW ----+                               +- ROWS ----+

                                          +-- XMLLOBWidth -- 0 --------+
                                          |                            |
 >--+---------------------------------+---+----------------------------+--|
    |                                 |   |                            |
    +---- SELect (select_syntax) -----+   +-- XMLLOBWidth -- n_bytes --+
```

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

- New **SCROLL** keyword parameter to use a DB2 scrollable SENSITIVE STATIC cursor to fetch DB2 result table rows.
- New **XMLLOBWIDTH** keyword parameter to specify the number of bytes (*n_bytes*) of text, at the start of an XML column, to be displayed for all XML columns in the DB2 table edit view.

FILTER

Syntax:

```
>>- FILTer ---------------------------------------------------------------><
```

Description:

New in SELCOPYi 3.40, FILTER will open the SDE FILTER Records panel.

LIST

Syntax:

```
>>-- LIst - listtype -- /listparms/ --+---------------------+-+---------+->
                                      |                     | |         |
                                      +- STEM rexx_stemvar -+ +- STRIP -+
                                      |                     |
                                      +- FILE filename -----+

    +- Lines ---+
    |           |
 >--+-----------+--+--------------------------+---------------------------><
    |           |  |                          |
    +- Columns -+  +- SUBset /select_clause/ -+
```

Description:

New in SELCOPYi 3.40, LIST is supported, as it is for the Text Editor, to extract rows of data returned from various SELCOPYi LIST type commands and either place the output in a temporary edit view or assign the fields to REXX stem variables for use in a Data Edit REXX macro.

Support in the Data Editor means that the LIST functionality (particularly STEM *rexx_stemvar*) can be used in Data Edit macros run in a batch environment (SDEAMAIN).

LOCATE

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

Syntax:

```
   +- Locate --+
   |           |
>>-+-----------+-------+---------------------------------------+----------><
                       |                                       |
                       +--| Record Locate Options |------------+
                       |                                       |
                       +--| Formatted Column Locate Options |--+
```

```
Record Locate Options:

 |---+----------------------------- .name ------------------------------+-|
     :                                                                  :
     |                       +- NEXT -----------  1 ----+               |
     |                       |                          |               |
     +-+-- expression ---+---+--------------------------+-----+------+--+
       |                 |   |                          |     |      |
       +-- line_flag ----+   +- NEXT----+   +---- 1 ----+     +- EX -+
                             |          |   |           |     |      |
                             +- FIRST --+---+-----------+     +- NX -+
                             |          |   |           |     |      |
                             +- LAST ---+   +- n_times -+     +- X --+
                             |          |               |
                             +- PREV ---+               |
                             |                          |
                             +- ALL  -------------------+
```

Existing primary command LOCATE has been enhanced to support argument *n_times* following NEXT, FIRST, LAST and PREV parameter keywords. If specified, the LOCATE command will be repeated so that it is executed *n_times* in total.

This is particulaly useful when displaying all records from a very large dataset and the first (or last) *n_lines* that satisfy a LOCATE condition are to be displayed. e.g.

```
  X ALL
  LOCATE NEXT 20   Track_Num < 5
```

Furthermore, LOCATE with no parameters will open the SDE LOCATE Records panel.

MAXIMUM

Syntax:

```
                 +------------------------- , --------------------------+
                 v                                                      |
>>-- MAXimum ----+- field_col ------------------------------------------+->
                 |                                                      |
                 |                                      +- CHaracter -+ |
                 |                                      |             | |
                 +- field_pos -- , - field_length -- , -+-------------+-+
                                                        |             |
                                                        +- Fixed -----+
                                                        +- Binary ----+
                                                        +- DECimal ---+
                                                        +- Zoned -----+
                                                        +- HFP -------+
                                                        +- BFP -------+
                               +- .ZFIRST -- .ZLAST -+  +- DFP -------+
                               |                     |
 >--+- WHere -- expression --+-+---------------------+-------------------->
    |                        | |                     |
    |                        | +- .name1 -+----------+
    | +- NX -----------+     |            |          |
    | +- NOTEXcluded --+     |            +- .name2 -+
    | |                |     |
    +-+----------------+-----+
      |                |
      +- EXcluded -----+
      +- X ------------+
      |                |
      +- ALL ----------+

 >--+--------------------------------------------+--+----------------+----><
    |                                            |  |                |
    +-------+- RECord -+--------+- record_type --+  +- STEM stemvar -+
    +- FOR -+          +- TYPE -+
```

Description:

New in SELCOPYi 3.40, MAXIMUM will display or extract the maximum value for values belonging to columns of data in unformatted records or, if formatting is applied, specific columns in records of a specific record type.

MINIMUM

Syntax:

```
                 +------------------------- , --------------------------+
                 v                                                      |
>>-- MINimum ----+- field_col ------------------------------------------+->
                 |                                                      |
                 |                                      +- CHaracter -+ |
                 |                                      |             | |
                 +- field_pos -- , - field_length -- , -+-------------+-+
                                                        |             |
                                                        +- Fixed -----+
                                                        +- Binary ----+
                                                        +- DECimal ---+
                                                        +- Zoned -----+
                                                        +- HFP -------+
                                                        +- BFP -------+
                               +- .ZFIRST -- .ZLAST -+  +- DFP -------+
                               |                     |
 >--+- WHere -- expression --+-+---------------------+-------------------->
    |                        | |                     |
    |                        | +- .name1 -+----------+
    | +- NX -----------+     |            |          |
    | +- NOTEXcluded --+     |            +- .name2 -+
    | |                |     |
    +-+----------------+-----+
      |                |
      +- EXcluded -----+
      +- X ------------+
      |                |
      +- ALL ----------+

 >--+--------------------------------------------+--+----------------+----><
    |                                            |  |                |
    +-------+- RECord -+--------+- record_type --+  +- STEM stemvar -+
    +- FOR -+          +- TYPE -+
```

Description:

New in SELCOPYi 3.40, MINIMUM will display or extract the minimum value for values belonging to columns of data in unformatted records or, if formatting is applied, specific columns in records of a specific record type.

REDIT

Syntax:

```
>>-- REdit ---+---------------------------------------------------+-------><
              |                                                   |
              +--- Parent ------+--+---------------------------+--+
              |                 |  |                           |
              +--- Dependent ---+  +- referential_constraint --+
              +--- Child -------+
```

Existing primary command REDIT is used to edit or browse a DB2 table that has a relationship with the DB2 rows displayed in the focus Data Editor view.

In SELCOPYi 3.40, REDIT has been enhanced so that specification of *referential_constraint* is not mandatory when optional parameter PARENT or CHILD (synonym DEPENDENT) is specified. If not specified, the default *referential_constraint* used will be the first (or only) PARENT or CHILD constraint found for the table in the SYSIBM.SYSRELS catalog table.

RFIND

Syntax:

```
             +---- 1 ----+
             |           |
>>-- RFInd --+-----------+-------------------------------------------------><
             |           |
             +- n_times -+
```

Existing primary command RFIND has been enhanced to support argument *n_times*. If specified, the RFIND command is repeated so that it is executed *n_times* in total.

Since entries at the command prompt are passed as parameters to commands executed via a PFkey, the *n_times* value may be typed on the command line before executing RFIND (assigned to F5 by default). e.g.

```
  X ALL             <-- Press Enter to exclude all records of the default record type.
  FIND  C'ABC' #5   <-- Press Enter to find the next occurrence of ABC in column #5.
  19                <-- Press F5 (RFIND) to find the 19th occurrence that follows.
```

RUNSELCOPY

Syntax:

```
                    +-- -PGM SELCOPY ---+
                    |                   |
>>-- RUNSELCopy ----+-------------------+--+-------------------+----------><
                    |                   |  |                   |
                    +-- -PGM progname --+  +-- parm_string ----+
```

Description:

New in SELCOPYi 3.40, RUNSELCOPY is supported, as it is for the Text Editor, to execute the program SELCOPY using the potentially unsaved control statements in the focus data edit view as the SYSIN input. SYSPRINT output is captured and displayed in another temporary edit view (RECFM=VBA, LRECL=1024).

RUNSELCOPY in both the Text and Data Editors now also supports specification of a SELCOPY load module name, *progname* (default SELCOPY), and a parameter string, *parm_string*. Any specified parameter string is passed unchanged to the SELCOPY program.

RUNSLC

Syntax:

```
                    +-- -PGM SLC -------+
                    |                   |
>>-- RUNSLC --------+-------------------+--+-------------------+----------><
                    |                   |  |                   |
                    +-- -PGM progname --+  +-- parm_string ----+
```

Description:

New in SELCOPYi 3.40, RUNSLC is supported, as it is for the Text Editor, to execute the program SLC using the potentially unsaved control statements in the focus data edit view as the SYSIN input. SYSPRINT output is captured and displayed in another temporary edit view (RECFM=VBA, LRECL=1024).

RUNSLC in both the Text and Data Editors now also supports specification of a SLC load module name, *progname* (default SLC), and a parameter string, *parm_string*. Any specified parameter string is passed unchanged to the SLC program.

SHIFT

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

Syntax:

```
>>-- SHIft --+- Left -+-- n_bytes --+-----------------------------------+-><
             |        |             |                                   |
             +- Right-+             +- BOUNds +- left_col -+ right_col -+
                                    |         |            |            |
                                    +- BNDs --+            +- * --------+
```

Existing primary command SHIFT has been enhanced to support a right boundary column specification of "*" (asterisk) which represents the prevailing right column boundary value.

SUM

Syntax:

```
                 +------------------------- , --------------------------+
                 v                                                      |
>>-- SUM --------+- field_col ------------------------------------------+->
                 |                                                      |
                 |                                      +- CHaracter -+ |
                 |                                      |             | |
                 +- field_pos -- , - field_length -- , -+-------------+-+
                                                        |             |
                                                        +- Fixed -----+
                                                        +- Binary ----+
                                                        +- DECimal ---+
                                                        +- Zoned -----+
                                                        +- HFP -------+
                                                        +- BFP -------+
                               +- .ZFIRST -- .ZLAST -+  +- DFP -------+
                               |                     |
 >--+- WHere -- expression --+-+---------------------+-------------------->
    |                        | |                     |
    |                        | +- .name1 -+----------+
    | +- NX -----------+     |            |          |
    | +- NOTEXcluded --+     |            +- .name2 -+
    | |                |     |
    +-+----------------+-----+
      |                |
      +- EXcluded -----+
      +- X ------------+
      |                |
      +- ALL ----------+

 >--+--------------------------------------------+--+----------------+----><
    |                                            |  |                |
    +-------+- RECord -+--------+- record_type --+  +- STEM stemvar -+
    +- FOR -+          +- TYPE -+
```

Description:

New in SELCOPYi 3.40, SUM will display or extract the sum of values belonging to columns of data in unformatted records or, if formatting is applied, specific columns in records of a specific record type.

TOTALS

Syntax:

```
                 +------------------------- , --------------------------+
                 v                                                      |
>>-- TOTals -----+- field_col ------------------------------------------+->
                 |                                                      |
                 |                                      +- CHaracter -+ |
                 |                                      |             | |
                 +- field_pos -- , - field_length -- , -+-------------+-+
                                                        |             |
                                                        +- Fixed -----+
                                                        +- Binary ----+
                                                        +- DECimal ---+
                                                        +- Zoned -----+
                                                        +- HFP -------+
                                                        +- BFP -------+
                               +- .ZFIRST -- .ZLAST -+  +- DFP -------+
                               |                     |
 >--+- WHere -- expression --+-+---------------------+--+---------------+->
    |                        | |                     |  |               |
    |                        | +- .name1 -+----------+  +- DIgits int --+
    | +- NX -----------+     |            |          |
    | +- NOTEXcluded --+     |            +- .name2 -+
    | |                |     |
    +-+----------------+-----+
      |                |
      +- EXcluded -----+
      +- X ------------+
      |                |
      +- ALL ----------+

 >--+--------------------------------------------+--+----------------+----><
    |                                            |  |                |
    +-------+- RECord -+--------+- record_type --+  +- STEM stemvar -+
    +- FOR -+          +- TYPE -+
```

Description:

New in SELCOPYi 3.40, TOTALS performs all the aggregate operations (AVERAGE, MAXIMUM, MINIMUM and SUM) for specific columns within unformatted records or, if formatting is applied, within records of a specific record type.

XMLBROWSE

Syntax:

```
>>--+- XMLBrowse -+---+-----------------+--+-----------------+------------><
    |             |   |                 |  |                 |
    +- XMLBrws ---+   +-- row_number ---+  +-- #column_ref --+
                      |                                      |
                      +-- #column_ref -----+-----------------+
                                           |                 |
                                           +-- row_number ---+
```

Description:

New in SELCOPYi 3.40 and applicable only to to DB2 table browse and edit, XMLBROWSE opens a new Data Editor view in order to display the XML document text located within a specific row and XML column of the current DB2 data edit view.

Unlike XMLEDIT and XMLVIEW, the display of the XML text is not formatted.

XMLBROWSE is sensitive to the cursor position. By default, if a *row_number* value is omitted and a multi-record data edit view is current, then the row on which the cursor is positioned will be used. Alternatively, for a single record data edit view, the focus table row will be used. If *#column_ref* is omitted, the DB2 table column on which the cursor is positioned will be used.

XMLEDIT

Syntax:

```
>>-- XMLEdit --+-----------------+--+-----------------+-------------------><
               |                 |  |                 |
               +-- row_number ---+  +-- #column_ref --+
               |                                      |
               +-- #column_ref -----+-----------------+
                                    |                 |
                                    +-- row_number ---+
```

Description:

New in SELCOPYi 3.40 and applicable to DB2 table edit only, XMLEDIT opens a new Text Editor view in order to edit and update XML document text located within a specific row and XML column of the current DB2 data editor view.

The text is formatted and displayed in the edit view so that a new line is started after each tag element not followed by text data and after each end tag. Nested tag elements are also indented to illustrate the tag hierarchy.

The text editor view containing the XML data is assigned a temporary dataset name. When edit updates to the XML document are saved, SELCOPYi does not attempt to write the data to DASD using the assigned DSN as it would for standard text edit. Instead, the count of alterations made since the last save is reset to zero and, when the text edit view is subsequently closed, the corresponding DB2 row in the data edit view is flagged as changed. When save is executed in the DB2 table edit view, a DB2 SQL UPDATE statement is executed which parses the updated XML document and verifies that it is well-formed before being saved to the DB2 table.

Like XMLBROWSE, XMLEDIT is sensitive to the cursor position.

XMLLENGTH

Syntax:

```
>>-- XMLLength --+-----------------+--+-----------------+-----------------><
                 |                 |  |                 |
                 +-- row_number ---+  +-- #column_ref --+
                 |                                      |
                 +-- #column_ref -----+-----------------+
                                      |                 |
                                      +-- row_number ---+
```

Description:

New in SELCOPYi 3.40 and applicable to DB2 table edit only, XMLLENGTH displays the length of text belonging to an XML document located within a specific row and XML column of the current DB2 data editor view.

The message format is:

```
  ZZSD697I XML document length of column "col_name" in row row_number
             (DocId=document_id) is length.
```

Where:

- *col_name* is the name of the DB2 XML column.
- *row_number* is the row number within the data edit display.

- *document_id* is the integer value found in the implicit DOCID column which uniquely identifies the row containing the XML document.
- *length* is the length of the XML document text.

Like XMLBROWSE, XMLLENGTH is sensitive to the cursor position.

XMLVIEW

Syntax:

```
>>-- XMLView --+-----------------+--+-----------------+------------------><
               |                 |  |                 |
               +-- row_number ---+  +-- #column_ref --+
               |                                      |
               +-- #column_ref -----+-----------------+
                                    |                 |
                                    +-- row_number ---+
```

Description:

New in SELCOPYi 3.40 and applicable to DB2 table edit only, XMLVIEW opens a new Text Editor view in order to perform a read-only edit of the XML document text located within a specific row and XML column of the current DB2 data editor view.

The text is formatted and displayed in the edit view so that a new line is started after each tag element not followed by text data and after each end tag. Nested tag elements are also indented to illustrate the tag hierarchy.

The text editor view containing the XML data is assigned a temporary dataset name and, although alterations to the text are possible, its contents cannot be used to update the XML column data in the DB2 table browse/edit view. However, alterations may be saved to a dataset, library member or HFS/ZFS file using the text edit primary command SAVE or SSAVE with parameter *fileid*.

Like XMLBROWSE, XMLVIEW is sensitive to the cursor position.

SET/QUERY/EXTRACT Options

The following Data Editor options have been included or updated in SELCOPYi 3.40.

COLWIDTHAUTO

Syntax:

```
>>-+-----------+-- COLWIDTHAuto ----+-- ON ---+---+--------------+--------><
   |           |                    |         |   |              |
   +- SET -----+                    +-- OFF --+   +-- Browse ----+
                                                  |              |
                                                  +-- Edit     --+
                                                  |              |
                                                  +-- All      --+


>>--- Query ------ COLWIDTHAuto ------------------------------------------><


>>--- EXTract --- /COLWIDTHAuto/ -----------------------------------------><
```

Description:

New in SELCOPYi 3.40 and applicable only to DB2 table browse and edit, COLWIDTHAUTO controls whether the displayed width of variable length columns is reduced to be equal to the maximum length of the data occupying a field in the column. i.e. for DB2 variable length columns COLWIDTH is automatically set to length of the longest value encountered on initial load of the DB2 result table.

This action may be set ON for either BROWSE only (the default), EDIT only or ALL (EDIT and BROWSE).

COPYBOOKPROC

Syntax:

```
>>-+-----------+- COPYBOOKProc ---+- COBOL ----+-+- INTERNAL --+----------><
   |           |                  |            | |             |
   +- SET -----+                  +- PL1 ------+ +- COMPILER --+


>>--- Query ----- COPYBOOKProc -------------------------------------------><


>>--- EXTract -- /COPYBOOKProc/ ------------------------------------------><
```

Description:

New in SELCOPYi 3.40 COPYBOOKPROC controls how COBOL and PL1 copybooks are to be processed by the CREATE STRUCTURE command if the COPYBOOKPROC parameter keyword is not specified.

By default, where available, SELCOPYi will use internal code to process copybooks. (Not currently supported for PL1).

FIDCHANGED

Syntax:

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


>>--- Query ------ FIDCHanged --------------------------------------------><


>>--- EXTract --- /FIDCHanged/ -------------------------------------------><
```

Description:

New in SELCOPYi 3.40, option FIDCHANGED is supported, as it is for the Text Editor, to SET, QUERY and EXTRACT the internal flag that indicates whether or not the fileid of the file in the focus data edit view has been changed by the user. (i.e. via the SET options DSN, FMODE, FNAME, MBR, FPATH, FTYPE or FILEID.)

USING

Syntax:

```
>>--- Query ------ USING -------------------------------------------------><

>>--- EXTract --- /USING/ ----------------------------------------------><
```

Description:

Used to QUERY or EXTRACT the name of the SDO structure and record-type definition assigned to the default record within the focus data edit view.

The following REXX compound variables have been added in SELCOPYi 3.40:

`using.3`	PERM or TEMP. TEMP indicates the SDO was created internally by SELCOPYi e.g. for a DB2 table or because a COBOL copybook was directly referenced.
`using.4`	The structure source. DIRECT, COBOL, ADATA COBOL, PL1, ADATA PL1, ASM, ADATA ASM, DB2, SEGMENT, RECORD.
`using.5`	EXPLICIT or AUTO. AUTO indicates that the mapping was automatically applied due to a dataset/structure association .

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 3.40 includes a number of new utilities and enhancements to existing utilities.

JSON Generation

SELCOPYi 3.40 introduces the JSON generation utility to generate JavaScript Object Notation (JSON) 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 names and values in the JSON output. The JSON output may be restricted to a subset of the input records (or table rows) and fields. Furthermore, fields may be resequenced before the JSON output is generated.

The utility processes the input records/rows and writes an output text dataset consisting of JSON {name:value} pairs. The JSON names correspond to the field names of the copybook/structure applied to the input dataset and the values to the field values expressed in character format.

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

JSONGEN Primary Command

Syntax:

```
>>- JSOngen -+----------------------------------------------------------+-><
             |                                                          |
             +- | 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 | -----+

   +- NOSPLit + +- NOREDefines -+ +- INDent 1 ------+
   |          | |               | |                 |
 >-+----------+-+---------------+-+-----------------+-+-----------------+->
   |          | |               | |                 | |                 |
   +- SPLit --+ +- REDefines ---+ +- INDent n_cols -+ +- LIMit n_lines -+

   +- BROWSE -+
   |          |
 >-+----------+-----------------------------------------------------------|
   |          |
   +- EDIT ---+
   |          |
   +- NOVIEW -+


Output dataset specification:
                                            +- EOL NL -------------+
                                            |                      |
 |- OUTDSN ( output_dataset ) -+----------+-+----------------------+------|
                               |          | |       +-- NL ------+ |
                               +- MODify -+ |       |            | |
                               |          | +- EOL -+------------+-+
                               +- APPend -+         |            |
                                                    +-- CR ------+
                                                    +-- LF ------+
                                                    +-- CRLF ----+
Character conversion option:
                                              +- terminal_ccsid -+
                                              |                  |
 |-+- CONVert -+------+- to_ccsid -+--------+-+------------------+-+------|
   |           |      |            |        | |                  | |
   |           +- TO -+            +- FRom -+ +- from-ccsid -----+ |
   |                                                               |
   +- ASCII -------------------------------------------------------+
   |                                                               |
   +- UNICODE -----------------------------------------------------+
```

Notes:

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

Description:

New to SELCOPYi 3.40, JSONGEN is used to produce an exportable copy of structured data as JavaScript Object Notation text. The structured data may belong to a specific structured dataset or the current SDE view.

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

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

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

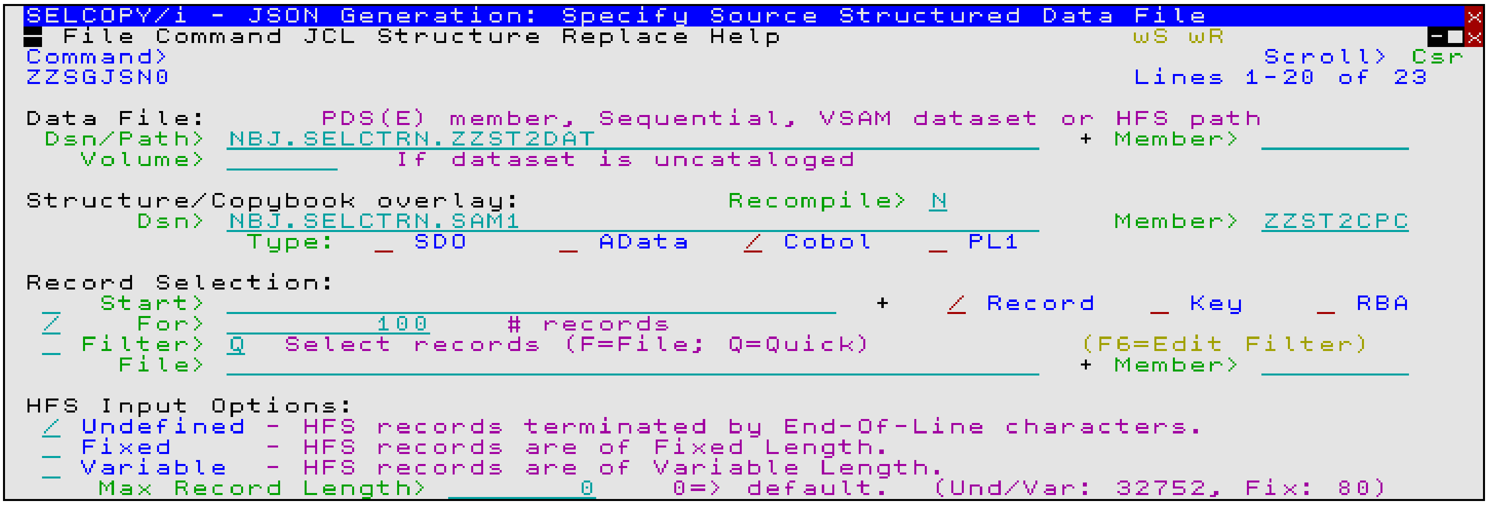
JSON Generation Panels

The JSON Generation utility supports two panel formats as follow:

JSON Generation Utility Panel

The **JSON Generation** utility panel may be displayed by selecting option 18 from the Utilities menu (=8.18) or by entering the primary command JSONGEN (or simply JSON) 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.



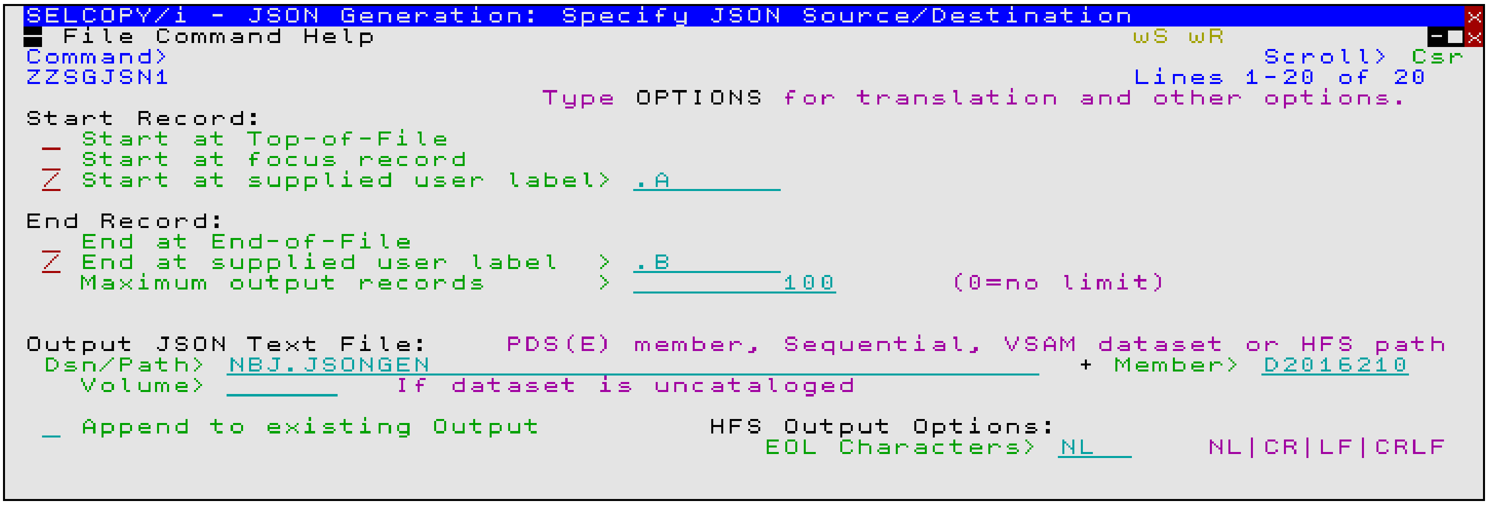
```
SELCOPY/i - JSON Generation: Specify Source Structured Data File
  File Command JCL Structure Replace Help                       wS wR
Command>                                                          Scroll> Csr
ZZSGJSN0                                                     Lines 1-20 of 23

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>        100     # records
 _  Filter> Q  Select records (F=File; Q=Quick)              (F6=Edit Filter)
      File>                                                     + Member>

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)
```

Figure 5. JSON Generation Utility Panel.

SDE JSON Generation Panel

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

The JSON generation is executed in the foreground and 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).



```
SELCOPY/i - JSON Generation: Specify JSON Source/Destination
  File Command Help                                          wS wR
Command>                                                            Scroll> Csr
ZZSGJSN1                                                     Lines 1-20 of 20
                            Type OPTIONS for translation and other options.
Start Record:
 _  Start at Top-of-File
    Start at focus record
 Z  Start at supplied user label> .A
 
End Record:
    End at End-of-File
 Z  End at supplied user label   > .B
    Maximum output records       >         100      (0=no limit)


Output JSON Text File:     PDS(E) member, Sequential, VSAM dataset or HFS path
 Dsn/Path> NBJ.JSONGEN                                  + Member> D2016210
   Volume>           If dataset is uncataloged

 _ Append to existing Output               HFS Output Options:
                                              EOL Characters> NL       NL|CR|LF|CRLF
```

Figure 6. SDE JSON Generation Panel.

Concatenated Library Directories

A DDname which has been allocated to concatenation of PDS/PDSE libraries may be referenced for input on the panels and primary commands for **COMPLIB** (library compare), **FCOPY** (file copy) and **FSU** (file search and update) utilities. This is supported in releases prior to SELCOPYi 3.40.

By default, FCOPY and FSU utilities would treat a DDname library concatenation as representing **all** occurrences of matching member names in the libraries. If, however, it is intended that the concatenation represents a search path, then an ampersand (&) prefix may now be specified on the DDname (e.g. &SYSEXEC). This notation implies that only the **first** occurrence of a matching member name found within the library concatenation search path will be selected for processing.

For FCOPY and FSU utilities which support processing multiple members that match an input member mask, the members are processed in alphabetical order of member name, not in library search order. The following will copy the first occurrence, not all occurrences, of member OSHELL found in the SYSPROC search path.

```
FCOPY    &SYSPROC(OSHELL)        USER.PROCLIB
```

In the following example, a member name (e.g. DSNTEDIT) may occur in more than one library in the SYSPROC library concatenation. However, only the first occurrence will be searched.

```
FSU     INPUT( &SYSPROC(DSN*) )     FIND( c'EXECIO' )
```

For the COMPLIB utility, a DDname (without an & prefix) identifies a library concatenation and may be specified as the new and/or old library reference. If a member with the same member name exists in more than one of the concatenated libraries, then only the first occurrence of the member within the concatenation will be compared.

FSU & COMPFILE Printed Report

By default, the report output generated for **COMPFILE** (file compare) and **FSU** (file search and update) utilities is presented (and optionally saved) as formatted data records. This type of generated report is suitable for display using an associated SDO structure in a Data Editor window view.

SELCOPYi 3.40 introduces an additional option to the COMPFILE and FSU primary command syntax to allow a print of the formatted report data to be performed on completion.

Unless NOREPORT is specified, PRINTREPORT controls whether or not the generated structured report output is to be printed. Keyword parameters determine the environment(s) in which execution of the FSU or COMPFILE primary command will generate printed output.

BATCH	Batch only using program SDEAMAIN. (Default)
INTERACTIVE	Interactive only (VTAM, TSO or ISPF)
ALL	Either batch or interactive environments.
NONE	Suppress print for all environments.

An optional *limit* argument may be specified as a numeric integer literal indicating the maximum number of pages of print output allowed. A limit value of 0 (zero) indicates that there should be no restriction placed on the number of output pages. This is the default.

If print output is required, the following PRINT utility command is executed at end-of-job.

```
PRINT FILE TRUNC LIMIT <limit>
  INDSN(      <ReportFileid>
        USING <ReportFileid>.SDO
        INITCMD ("SELECT  zFileId,zRecNo,zRecord,*    FROM Hit"
                 "VFMT"
                 "PUSH SAVEOPTIONS"
                 "SAVEOPTIONS OFF"
                 "RECLEN OFF"
                 "REFERENCE OFF"
                 "PREFIX OFF"
                 "POP SAVEOPTIONS"
                )
       )
```

FSU Primary Command

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

Syntax:

```
>>- FSU -+---------------------------------------------------------------+-><
         |                                                               |
         +-+-| Unformatted File Search Opts |--+----| Common Opts |-----+
           |                                   |
           +-| Unformatted File Update Opts |--+
           |                                   |
           +-| Unformatted File Copy Opts |----+
           |                                   |
           +-| Unformatted Library Copy Opts |-+
           |                                   |
           +-| Formatted File Search Opts |----+
           |                                   |
           +-| Formatted File Update Opts |----+
           |                                   |
           +-| Formatted File Copy Opts |------+
           |                                   |
           +-| Formatted File Remap Opts |-----+
           |                                   |
           +-| Formatted Library Copy Opts |---+
           |                                   |
           +-| Formatted Library Remap Opts |--+
```

```
Common Opts:

   +----- PRINTReport --- BATCH --------------- 0 --------+
   |                                                      |
 |-+------------------------------------------------------+---------------|
   |                                                      |
   |                   +- ALL ----------+  +--- 0 -----+  |
   |                   |                |  |           |  |
   +----- PRINTReport -+----------------+--+-----------+--+
                       |                |  |           |
                       +- BATCH --------+  +-- limit --+
                       |                |
                       +- INTERACTIVE --+
                       |                |
                       +- NONE ---------+
```

Existing primary command FSU has been enhanced to support the PRINTREPORT parameter together with ALL, BATCH, INTERACTIVE and NONE keyword arguments and a print number of pages *limit*.

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 | ---+
Common Opts:

   +----- PRINTReport --- BATCH --------------- 0 --------+
   |                                                       |
 |-+-------------------------------------------------------+----------------|
   |                                                       |
   |                   +- ALL ----------+  +--- 0 -----+  |
   |                   |                 |  |           |  |
   +----- PRINTReport -+-----------------+--+-----------+--+
                       |                 |  |           |
                       +- BATCH ---------+  +-- limit --+
                       |                 |
                       +- INTERACTIVE ---+
                       |                 |
                       +- NONE ----------+
```

Existing primary command COMPFILE has been enhanced to support the PRINTREPORT parameter together with ALL, BATCH, INTERACTIVE and NONE keyword arguments and a print number of pages *limit*.

COMPLIB Library Compare

The COMPLIB library compare utility has been enhanced in SELCOPYi 3.40 to support the following:

Batch Execution

Support of Text Editor options and primary commands in the Data Editor has enabled execution of the COMPLIB primary command in a batch environment.

Each record of the COMPLIB output report contains a short message describing the result of the member compare followed by a complete COMPFILE command which may be executed via the ACTION key (F4). Each generated COMPFILE command executes a compare against members of the same name in the old and new libraries. The COMPLIB report is directed to SYSPRINT.

```
//NBJJOB JOB ,,CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//SELC0001 EXEC PGM=SDEAMAIN,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=CBL.CBLI340.EXE
//ZZSUSERI DD DUMMY
//SDEPRINT DD SYSOUT=*
//SYSPRINT DD DISP=SHR,DSN=CBL.SYSPRINT(COMPLIB)
//ASMNEW   DD DISP=SHR,DSN=CBL.CBLI340.ASM
//ASMOLD   DD DISP=SHR,DSN=CBL.CBLI330.ASM
//SDEIN    DD   *

 COMPLIB    ASMNEW    ASMOLD    /* Compare HLASM source lbraries. */
    STRIP  ' '
    SYNC   READAHEAD(200)
    LIMIT   0
    SHOWMATCHES
    CONTEXT 10

/*
```

Figure 7. COMPLIB Batch Execution.

Uncataloged Libraries

Uncataloged library dataset names may be identified as the old and/or new library specifications in the comparison simply by including the volume id of the DASD volume on which the first extent of the library is saved. The volume id is specified with a trailing ":" (colon) before the library DSN (volser:dsn). e.g.

```
COMPLIB    CBLMCT:CBL.NEW.JCL(SS*)      CBLMCT:CBL.OLD.JCL
```

DB2 XML Text Import/Export

To complement SELCOPYi support for DB2 XML column data browse and edit using the Data Editor, two addional utilities have been introduced to import and export XML document text between a dataset object and a DB2 table XML column entry.

These utilities are supported as primary commands, GETXML and PUTXML.

GETXML Primary Command

Syntax:

```
>>-- GETXML --- INTO --+--------+- fileid -------------------------------->
                       |        |
                       +- FILE -+
 >------ FROM --+---------+- table_name ------+----------+- column_name -->
                |         |                   |          |
                +- TABle -+                   +- COLumn -+

 >--------------+---------+- DOCid -+-----+- document_id -+--------------->
                |         |         |     |               |
                +- WHERE -+         +- = -+               |
                |                                         |
                +- WHERE ( where_clause ) ----------------+

 >--------------+-----------------------+---+---------+-------------------><
                |                       |   |         |
                +- DB2 -+--+-- ssn --+      +- AUDit -+
                |       |  |            |
                +- SSN -+  +- (ssn) -+
```

Description:

New in SELCOPYi 3.40 and applicable to DB2 base tables only, GETXML copies an XML document from an XML column within a specific DB2 table row to a dataset, library member or HFS/ZFS file.

If a connection to the required DB2 sub-system does not already exist, then GETXML will first start a new connection before fetching the XML column data from selected table row and disconnect again afterwards.

Having retrieved the XML document to the specified *fileid*, the XML text is formatted and displayed in a text edit view. Formatting of the XML document starts a new line after each tag element not followed by text data and after each end tag. Nested tag elements are also indented to illustrate the tag hierarchy. A SAVE must be executed if the formatting is to be preserved.

A WHERE filter must be specified which uniquely identifies the table row from which the XML column data will be copied. If the filter identifies more than one table row, then the operation fails with error ZZSD711E.

PUTXML Primary Command

Syntax:

```
>>-- PUTXML --- FROM --+--------+- fileid -------------------------------->
                       |        |
                       +- FILE -+

 >------ INTO --+---------+- table_name ------+----------+- column_name -->
                |         |                   |          |
                +- TABle -+                   +- COLumn -+

 >--------------+---------+- DOCid -+-----+- document_id -+--------------->
                |         |         |     |               |
                +- WHERE -+         +- = -+               |
                |                                         |
                +- WHERE ( where_clause ) ----------------+

 >--------------+-----------------------+---+---------+-------------------><
                |                       |   |         |
                +- DB2 -+--+-- ssn --+      +- AUDit -+
                |       |  |            |
                +- SSN -+  +- (ssn) -+
```

Description:

New in SELCOPYi 3.40 and applicable to DB2 base tables only, PUTXML copies an XML document from a dataset, library member or HFS/ZFS file to an XML column within a specific DB2 table row.

Each execution of PUTXML will execute an immediate SQL UPDATE statement, to update the selected table row, and then COMMIT the change. If a connection to the required DB2 sub-system does not already exist, then PUTXML will start a new connection before attempting the UPDATE, and disconnect again following the COMMIT.

A data edit view of the DB2 table is not necessary. If a data edit view of the table has been opened using a non-zero value for BROWSE/EDIT option XMLLOBWIDTH, then the display of the XML column data is not updated as a result of an execution of PUTXML. However, execution of XMLBROWSE, XMLEDIT or XMLVIEW on the updated XML column entry will display the updated XML document text.

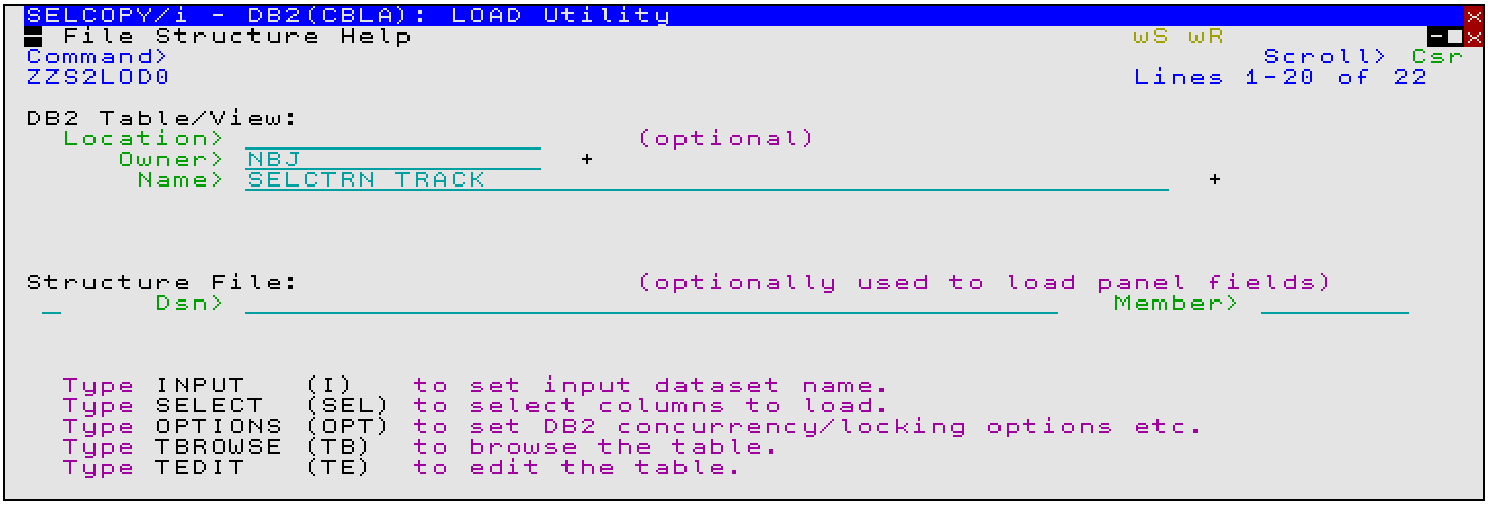
A WHERE filter must be specified which uniquely identifies the table row to which the XML document will be copied. If the filter identifies more than one table row, then the operation fails with error ZZSD711E.

DB2 LOAD Utility Panel

New in SELCOPYi 3.40, The DB2 Load Utility panel (ZZS2LOD0) assists the user in creating a batch job to load data into a DB2 table from a sequential input dataset or PDS/PDSE library member.

Options are provided to:

- Select table columns.
- Input data in delimited format. e.g. Comma separated (CSV).
- Map table columns to input fields defined by a structure/copybook.



```
SELCOPY/i - DB2(CBLA): LOAD Utility
   File Structure Help                                        wS wR
Command>                                                            Scroll> Csr
ZZS2LOD0                                                      Lines 1-20 of 22

DB2 Table/View:
  Location> ________________     (optional)
     Owner> NBJ_____________  +
      Name> SELCTRN_TRACK__________________________________  +



Structure File:                  (optionally used to load panel fields)
 _      Dsn> ____________________________________________  Member> ________


  Type INPUT   (I)   to set input dataset name.
  Type SELECT  (SEL) to select columns to load.
  Type OPTIONS (OPT) to set DB2 concurrency/locking options etc.
  Type TBROWSE (TB)  to browse the table.
  Type TEDIT   (TE)  to edit the table.
```

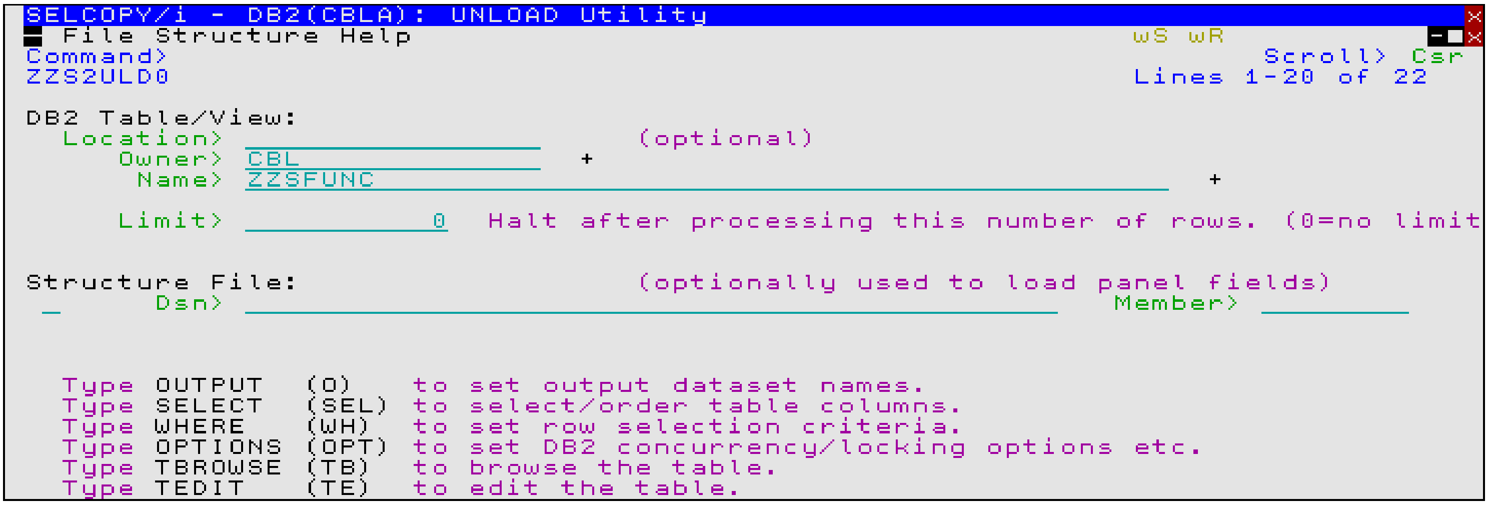
Figure 8. DB2: LOAD Utility Panel.

DB2 UNLOAD Utility Panel

New in SELCOPYi 3.40, the DB2 Unload Utility panel (ZZS2ULD0) assists the user in creating a batch job to unload data from a DB2 table to a sequential output dataset.

Options are provided to:

- Select table columns and set row sort order.
- Set table row selection criteria.
- Set DB2 concurrency and locking options, etc.
- Output data in delimited format. e.g. Comma separated (CSV).
- Map table columns to output fields defined by a structure/copybook.

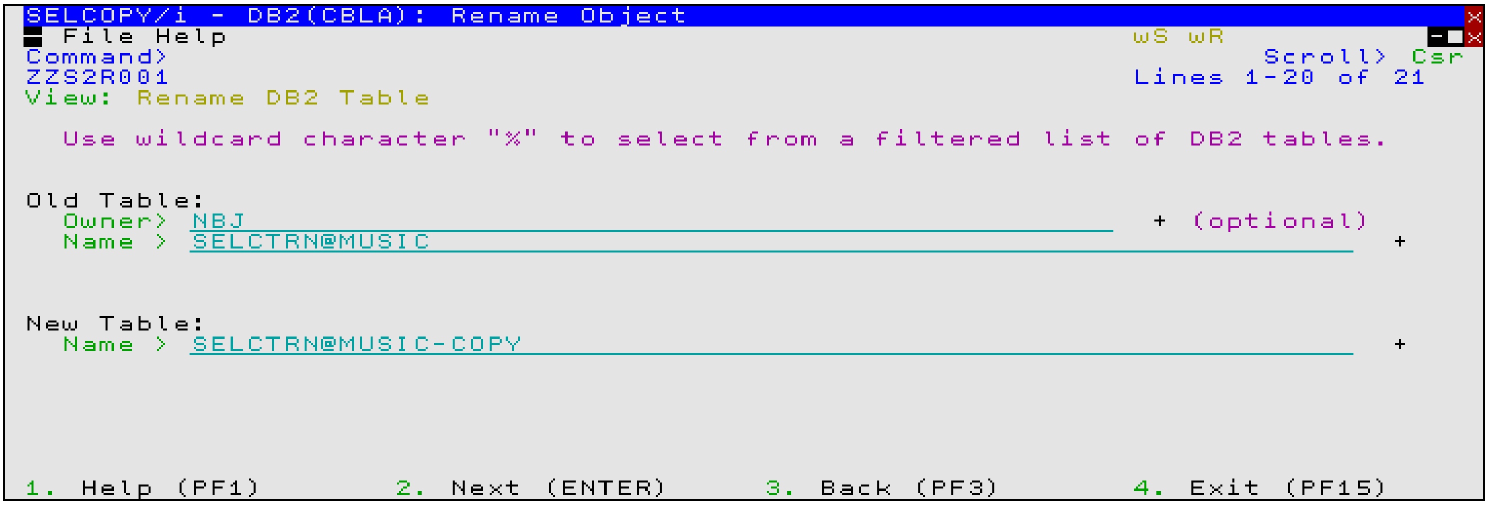


```
SELCOPY/i - DB2(CBLA): UNLOAD Utility
   File Structure Help                                        wS wR
Command>                                                            Scroll> Csr
ZZS2ULD0                                                      Lines 1-20 of 22

DB2 Table/View:
  Location> ________________     (optional)
     Owner> CBL_____________  +
      Name> ZZSFUNC________________________________________  +

     Limit> __________0  Halt after processing this number of rows. (0=no limit

Structure File:                  (optionally used to load panel fields)
 _      Dsn> ____________________________________________  Member> ________


  Type OUTPUT  (O)   to set output dataset names.
  Type SELECT  (SEL) to select/order table columns.
  Type WHERE   (WH)  to set row selection criteria.
  Type OPTIONS (OPT) to set DB2 concurrency/locking options etc.
  Type TBROWSE (TB)  to browse the table.
  Type TEDIT   (TE)  to edit the table.
```

Figure 9. DB2: UNLOAD Utility Panel.

DB2 Rename Table and Index Panels

The DB2 Rename Object sequence of panel views (ZZS2R001) have been introduced in SELCOPYi 3.40 to generate an SQL SQL RENAME statement that renames the selected object type (Table or Index) on the current DB2 server. Note that the current DB2 server (subsystem name) is displayed in the panel window title bar.



```
SELCOPY/i - DB2(CBLA): Rename Object
  File Help                                                   wS wR
Command>                                                             Scroll> Csr
ZZS2R001                                                      Lines 1-20 of 21
View: Rename DB2 Table

  Use wildcard character "%" to select from a filtered list of DB2 tables.


Old Table:
  Owner> NBJ                                                    + (optional)
  Name > SELCTRN@MUSIC                                                      +


New Table:
  Name > SELCTRN@MUSIC-COPY                                                 +




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

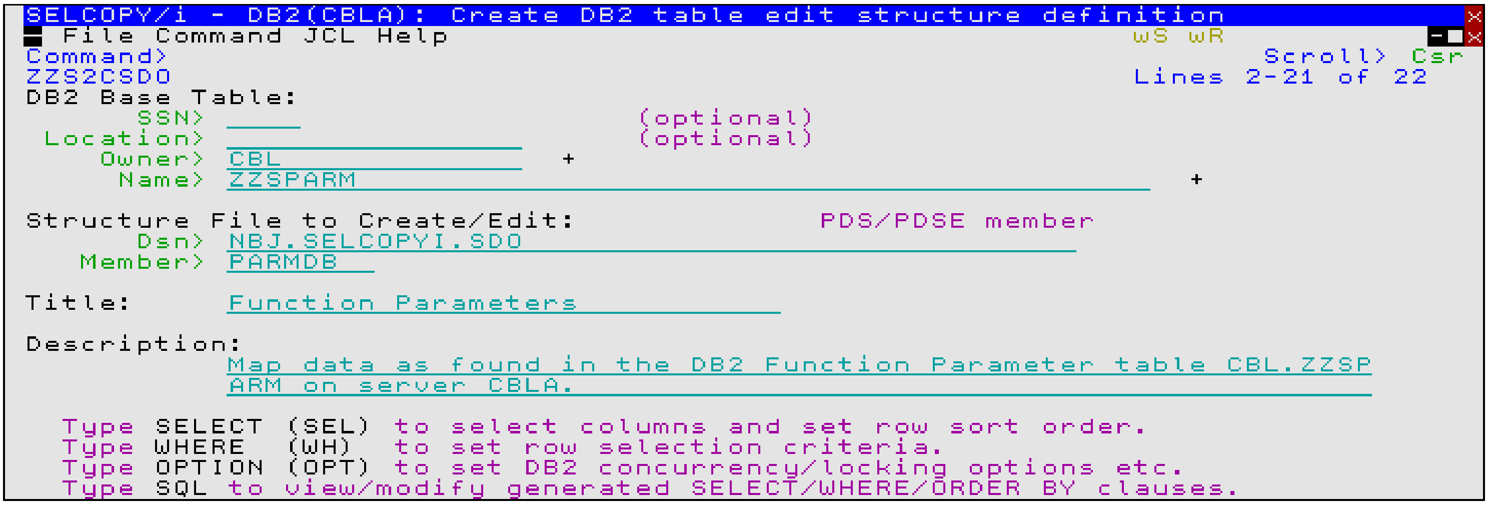
Figure 10. DB2: Rename Object Panel.

DB2 Create Structure Panels

New in SELCOPYi 3.40, the Create Structure from DB2 Table panel (ZZS2CSDO) helps to create a structure (SDO) for a table that may be referenced when browsing or editing that table using the SELCOPYi Data Editor.

While it is not necessary to specify an SDO when editing a table (the structure of the tables columns is known to DB2 itself) an external structure does allow the user to specify defaults such as:

- Columns selected for initial display, their order and the column widths.
- A generated WHERE clause for row selection.
- Various DB2 processing options such as concurrency level and commit frequency.



```
SELCOPY/i - DB2(CBLA): Create DB2 table edit structure definition
  File Command JCL Help                                       wS wR
Command>                                                             Scroll> Csr
ZZS2CSDO                                                      Lines 2-21 of 22
DB2 Base Table:
      SSN>                    (optional)
 Location>                    (optional)
    Owner> CBL             +
     Name> ZZSPARM                                         +

Structure File to Create/Edit:              PDS/PDSE member
      Dsn> NBJ.SELCOPYI.SDO
   Member> PARMDB

Title:     Function Parameters

Description:
           Map data as found in the DB2 Function Parameter table CBL.ZZSP
           ARM on server CBLA.

  Type SELECT (SEL) to select columns and set row sort order.
  Type WHERE  (WH)  to set row selection criteria.
  Type OPTION (OPT) to set DB2 concurrency/locking options etc.
  Type SQL to view/modify generated SELECT/WHERE/ORDER BY clauses.
```

Figure 11. DB2: Create Structure Panel.

DB2 Miscellaneous Updates

In addition to the significant DB2 features already mentioned, SELCOPYi 3.40 introduces a number of minor changes and enhancements to its DB2 functionality as follows:

DB2 Alter Tablespace Panel

Opened when the prefix (line) command "A" is entered against an entry in the DB2 Tablespace list window, the Alter Tablespace panel allows changes to be made to an existing table space on the current DB2 server.

The format of the Alter Tablespace panel views match those provided by the Create Tablespace utility. Where Create Tablespace parameter fields are not supported by the SQL ALTER TABLESPACE statement, they are omitted from the equivalent panel views for Alter Tablespace.

DB2 Drop Object Panels

Drop object panels have been updated so that they adopt the same panel view sequence as create DB2 object panels and use the common "Generate SQL" view as the last view in the sequence. This allows the generated SQL DROP statement to be added to an existing SQL syntax member or executed via a JCL batch procedure.

DB2 List Table Utilities

Execution of the File Utilities macro (FILEUTIL) against entries in a DB2 table list is now supported using prefix command "SC". The pop-up menu displays items to generate template SELCOPY batch program control statements to read and print the selected DB2 table.

For JCL batch job creation, the STEPLIB DD card is generated from the SELCOPYi INI variable DB2*xxxx*.LOADPATH where *xxxx* is the DB2 sub-system name. (Note that INI option variables may be assigned using the Text Editor SET INIVAR primary command.)

DB2 Table Compare Utility Panel

The DB2 Table Compare panel supports a compare of column data from two DB2 result tables specified as a cataloged DB2 table name, view name, alias name, table synonym or as an explicit SQL query (SELECT) statement.

The key columns defining the order in which row entries are fetched, may be specified individually or, for cataloged DB2 table names only, via a defined DB2 index name. SELCOPYi 3.40 extends support for DB2 index name specification to include DB2 cataloged alias names and synonyms and also DB2 views in which all columns of a single cataloged table are selected.

DB2 Settings - Audit Log

The SELCOPYi DB2 audit log dataset name prefix now defaults to be the installation default as defined for each user via the INI option SYSTEM.UserDSNPrefix. Furthermore, this prefix is no longer truncated at 8 characters, so allowing a prefix comprising a number of 8 byte qualifiers if needed.

Previously, the user's ACF security manager logon id was used which, for some installations, did not comply with dataset naming standards.

Note that the default DB2 audit log DSN prefix may be overridden using the DB2 Audit File Settings panel (=0.7.1).

SELCOPY IDE/Debug

The SELCOPY Interactive Development Environment (Debug) utility allows step through SELCOPY control statements and set break points. In doing so, a developer can track the values of variables and expressions and also display and update areas of storage.

SELCOPYi 3.40 includes a number of new enhancements to the SELCOPY IDE/Debug utility.

SELCOPY Debug - Processing Enhancements

The following new features have been introduced which relate to general operation of the SELCOPY Debug utility.

SELCOPY Debug - SLC (SELCOPY C++) Program

SELCOPYi 3.40 introduces support for debug of SELCOPY control statements processed using the SLC program.

SELCOPY source code is written in the Basic Assembly Language (BAL) whereas the SLC program, which also executes SELCOPY control statements, is written using the C++ language.

SLC was first developed to provide a version of SELCOPY that executes on Microsoft Windows, Linux and various Unix platforms. Since its inception, this C++ version of SELCOPY has introduced many new facilities that are not available in the BAL version. So that programmers on mainframe systems could take advantage of these new facilities, the C++ version was compiled for z/OS and z/VM CMS systems and included as the executable load module, SLC, in the SELCOPY Product Suite package.

Although the SELCOPY (BAL) program remains the default SELCOPY language interpreter for legacy programs, developers are encouraged to explore and use the additional facilities offered by SLC in new applications.

Since SLC supports data elements not recognised by SELCOPY (i.e. declared variables), a new debug WATCH window (see below) has been introduced to monitor the status of any selected variable or expression.

To support development of SLC, the SELCOPYi on-line help has been expanded to include all sections of the *SELCOPY C++ (SLC) Language Reference* that accompanies this 3.40 product release.

From the SELCOPY Debug startup menu and panels (=8.1), use of the SLC program to debug control statement input is automatic if that input is a JCL job step which has EXEC PGM=SLC. If input is directly from a control statement file, then SLC must be entered in the Version> panel field, or, if starting debug via the SELCOPY primary command, -SLC must be specified as a keyword parameter.

SELCOPY Debug - WATCH List

The SELCOPY Debug Watch List window is introduced in SELCOPYi 3.40 to display the current value of specific variables and locations in storage. This list is opened using the WATCH primary command and allows the user to focus on specific values of interest throughout the debug session.

Any number of Watch lists may be opened. Each instance of a Watch list has a unique name identifier which is assigned when the list window is opened. The name used may be specified on the WATCH command or otherwise allowed to defualt to W*n* (where *n* is a number in an internally maintained sequence).

Each entry of the list must either be a storage location, as supported by SELCOPY POS *expression* syntax, or any variable supported by the SELCOPY or SLC programs. Specifically, these are:

- SELCOPY internal variables (e.g. LRECL, DIFF, RETCODE).
- SELCOPY internal variable named source fields (e.g. UXLRECL, UXADIFF, UXATPTR).
- User @pointer variables.
- User and SELCOPY generated declared variables. (SLC only)

Entries in the list may be inserted, moved, excluded and deleted, gap lines may be inserted (SPACE) and scale lines may be displayed (SCALE ON) for individual character values. Furthermore, character values may be scrolled horizontally and all types of values may be displayed with or without their source hexadecimal representation (HEX ON/OFF).

Columns displayed in the watch list are as follows:

`Name`

The name of the variable or the name generated by SELCOPY Debug for a POS *expression*. A generated expression name is of the form "POS_*expression*" where *expression* contains no blank characters.

`DataType`

The built-in data type of the variable or storage field at POS *expression*. For decimal data types (packed and zoned), the precision and scale of the value immediately follows in parentheses. For all other data types, the length of the value/field follows in parentheses.

Display of this column is optional and may be included or excluded using SET DATATYPE ON/OFF. Possible data type entries are as follows:

bfp(n)	Binary floating point with source field length of *n* bytes.
bin(n)	Binary integer with source field length of *n* bytes.
char(n)	Character of fixed length *n*.
charb(n)	Non-printable character (binary data) of fixed length *n* bytes.
charv(n)	Blank padded variable length character of maximum length *n*.
cstring(n)	Null terminated character string of maximum length *n*.
dec(p,s)	Packed decimal of precision *p* and scale *s*.
dfp(n)	Decimal floating point with source field length of *n* bytes.
hfp(n)	Hexadecimal floating point with source field length of *n* bytes.
ptr(4)	Storage address pointer field of length 4 bytes.
vchar(n)	Variable length character of maximum length *n*.
zd(p,s)	Zoned decimal of precision *p* and scale *s*.

Col

Applicable to values of a character data type only, this column displays the horizontal scrolling position of the first character displayed in the Value column.

The column position may be displayed as either a decimal or hexadecimal value using SET COLUMN DEC/HEX. This value may also be overtyped to reposition the data within the Value column.

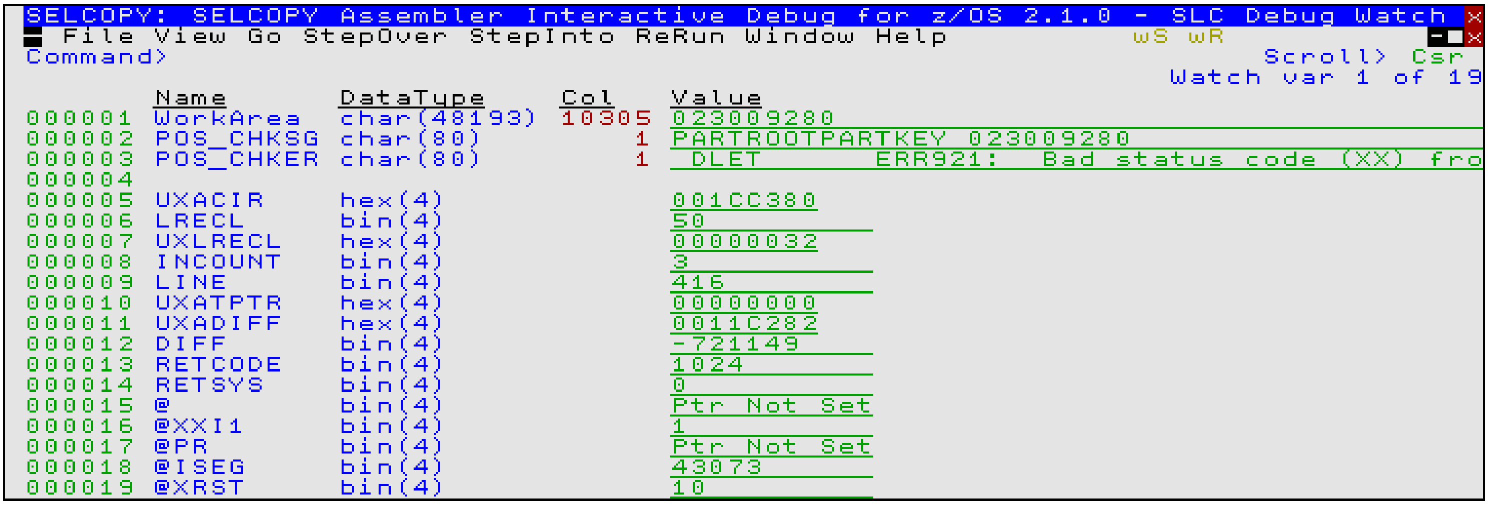
Value

The value assigned to the named variable or field at POS *expression*.

For numeric fields, the value is displayed in decimal with "." (dot/period) representing a decimal point and a "-" (minus) prefix representing a negative value. Positive fixed point values have no sign prefix. Floating point values are normalised, are always displayed with a leading "+" (plus) or "-" (minus) sign and include a signed exponent of the form "E+nn" or "E-nn".

For variables and fields of character data type, an individual value may be scrolled horizontally (LEFT and RIGHT) without scrolling other values in the Value column. Furthermore, a scale line may be displayed above the value using SET SCALE ON/OFF. This scale line scrolls with the value's text.

For @pointer variable values that are unset, the Value field displays "Ptr Not Set". If a storage field position expression cannot be determined, the Value column displays "?? Location unresolved ??" (or a sub-string of this message if the value length is less than 25).



```
SELCOPY:  SELCOPY Assembler Interactive Debug for z/OS 2.1.0 - SLC Debug Watch
  File View Go StepOver StepInto ReRun Window Help          wS wR
Command>                                                           Scroll> Csr
                                                              Watch var 1 of 19
       Name      DataType    Col   Value
000001 WorkArea  char(48193) 10305 023009280
000002 POS_CHKSG char(80)        1 PARTROOTPARTKEY 023009280
000003 POS_CHKER char(80)        1  DLET      ERR921:  Bad status code (XX) fro
000004
000005 UXACIR    hex(4)            001CC380
000006 LRECL     bin(4)            50
000007 UXLRECL   hex(4)            00000032
000008 INCOUNT   bin(4)            3
000009 LINE      bin(4)            416
000010 UXATPTR   hex(4)            00000000
000011 UXADIFF   hex(4)            0011C282
000012 DIFF      bin(4)            -721149
000013 RETCODE   bin(4)            1024
000014 RETSYS    bin(4)            0
000015 @         bin(4)            Ptr Not Set
000016 @XXI1     bin(4)            1
000017 @PR       bin(4)            Ptr Not Set
000018 @ISEG     bin(4)            43073
000019 @XRST     bin(4)            10
```

Figure 12. SELCOPY Debug: Watch List.

The Watch list includes a prefix area which, unless modified or removed using the SET PREFIX option, is displayed on the left of the list entry data and occupies 6 display columns.

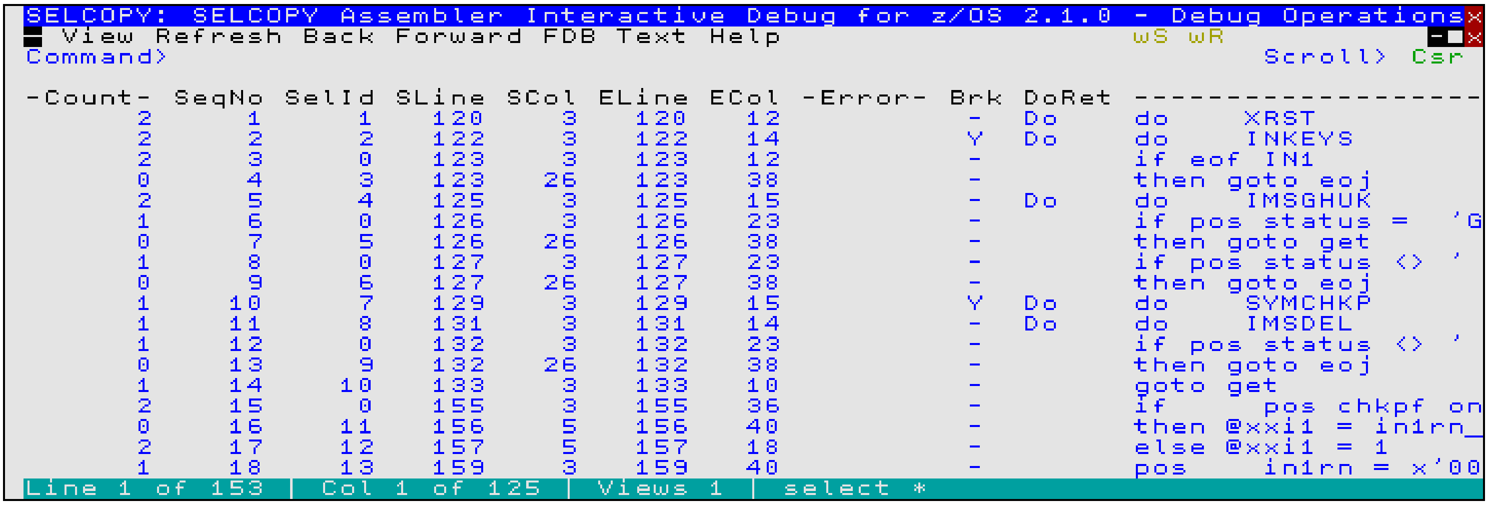
The following Watch list prefix area (line) commands are supported. The list entry against which a line command is entered, is considered to be the focus entry for the command.

Command	Description
`.name`	Set a line pointer (line name) at the focus list entry.
`A`	Set the focus entry as the target of a prefix area move command. (Move entries after the focus entry).
`B`	Set the focus entry as the target of a prefix area move command. (Move entries before the focus entry).
`D[n] D* DD`	Delete an entry or a block of entries. D*n* deletes a block of *n* entries starting at the focus entry. D* deletes a block of all entries from the focus entry to the end of the list. A DD pair is used to mark the first and last entry in a block of entries to be deleted. Note that excluded enries are included within a block of entries to be deleted.
`F[n] F*`	Applicable only if the focus entry is a shadow line (SHADOW ON), this command includes and displays the first excluded entry or block of entries represented by the shadow line. F*n* includes the first block of *n* entries. If *n* is greater than or equal to the number of excluded entries represented by the shadow line, then all its excluded entries are included. F* includes all excluded lines represented by the shadow line.
`I`	Open the Add WATCH Var/PosExp panel to insert a Watch List entry following the focus entry.
`L[n] L*`	Applicable only if the focus entry is a shadow line (SHADOW ON), this command includes and displays the last excluded entry or block of entries represented by the shadow line. L*n* includes the last block of *n* entries. If *n* is greater than or equal to the number of excluded entries represented by the shadow line, then all its excluded entries are included. L* includes all excluded lines represented by the shadow line.
`M[n] M* MM`	Mark an entry or a block of entries to be moved before or after a target list entry line which has been selected using prefix command B or A respectively. M*n* marks a block of *n* entries starting at the focus entry. M* marks a block of all entries from the focus entry to the end of the list. An MM pair is used to mark the first and last entry in a block of entries to be moved. Note that excluded entries are included within a block of entries to be moved.
`PW`	Open a POS Expression Window for the focus entry.
`S`	Applicable only if the focus entry is a shadow line (SHADOW ON), this command includes and displays all excluded entries represented by the shadow line.
`SC`	Applicable only if the focus entry is of character or hex data type, this command toggles display of a counting scale line above the Value column of the focus list entry.
`SP`	Insert a blank (spacer) entry after the focus entry.
`X[n] X* XX`	Exclude an entry or a block of entries. X*n* excludes a block of *n* entries starting at the focus entry. X* excludes a block of all entries from the focus entry to the end of the list. An XX pair is used to mark the first and last entry in a block of entries to be excluded. Note that entries that are already excluded are still included within a count of entries to be excluded.

SELCOPY Debug - Operations List

The SELCOPY Debug Operations List window is introduced in SELCOPYi 3.40 to display the current status held by SELCOPY Debug for each executable statement (SELCOPY operation) in the control statements input file.

The list is opened using the LIST OPERATIONS primary command and may be used to audit a SELCOPY execution and demonstrate execution of individual SELCOPY operations within the run.



```
SELCOPY: SELCOPY Assembler Interactive Debug for z/OS 2.1.0 - Debug Operations
 View Refresh Back Forward FDB Text Help                        wS wR
Command>                                                          Scroll> Csr

-Count- SeqNo SelId SLine SCol ELine ECol -Error- Brk DoRet -------------------
      2     1     1   120    3   120   12          -  Do    do     XRST
      2     2     2   122    3   122   14          Y  Do    do     INKEYS
      2     3     0   123    3   123   12          -        if eof IN1
      0     4     3   123   26   123   38          -        then goto eoj
      2     5     4   125    3   125   15          -  Do    do     IMSGHUK
      1     6     0   126    3   126   23          -        if pos status =  'G
      0     7     5   126   26   126   38          -        then goto get
      1     8     0   127    3   127   23          -        if pos status <> '
      0     9     6   127   26   127   38          -        then goto eoj
      1    10     7   129    3   129   15          Y  Do    do     SYMCHKP
      1    11     8   131    3   131   14          -  Do    do     IMSDEL
      1    12     0   132    3   132   23          -        if pos status <> '
      0    13     9   132   26   132   38          -        then goto eoj
      1    14    10   133    3   133   10          -        goto get
      2    15     0   155    3   155   36          -        if      pos chkpf on
      0    16    11   156    5   156   40          -        then @xxi1 = in1rn_
      2    17    12   157    5   157   18          -        else @xxi1 = 1
      1    18    13   159    3   159   40          -        pos     in1rn = x'00
Line 1 of 153 | Col 1 of 125 | Views 1 | select *
```

Figure 13. SELCOPY Debug: Operations List.

Columns displayed in this list are as follows:

Name	Type	Description
Count	Integer	Operation execution count.
SeqNo	Integer	Operation sequence number.
SelId	Integer	Operation SELCOPY selection id.
SLine	Integer	Control statement start line.
SCol	Integer	Control statement start column.
ELine	Integer	Control statement end line.
ECol	Integer	Control statement end column.
Error	Enum	Control statement is in error. (Seq, Invalid or Overlap)
Brk	Flag	Operation has a breakpoint set.
DoRet	Enum	Operation is DO or RETURN.
Text	Character	Operation text.

SELCOPY Debug - IMS/DL1

The SELCOPY program supports data management calls to IMS/DL1 data bases to perform segment I/O operations for offline DLI processing or via an IMS online batch message processing (BMP) region.

When SELCOPY control statements that perform IMS/DL1 database I/O are executed the SELCOPY program is started under the IMS/DL1 region controller (DFSRRC00). When run in SELCOPY Debug, the PSB name, region type (DLI or BSP) and 4-character IMS system identifier are passed as parameters to DFSRRC00. DL/I calls with IMS DB are executed using the ASMTDLI assembler interface from a sub-task of SELCOPYi.

SELCOPY Debug processing of IMS/DL1 data bases has been enhanced in SELCOPYi 3.40 to support the following:

Check Point ID

SELCOPY does not have native support for extended check point (CHKP) and restart (XRST) which is required to restart the program (SELCOPY) from a check point. Extended restart will recover user buffers (e.g. the SELCOPY workarea) and reposition database pointers to their status at the time an extended check point was performed.

Although not supported natively, the SELCOPY CALL operation may be used to call the ASMTDLI load module with a parameter list that executes the call to extended CHKP or XRST. Doing this, it is possible to write a re-startable SELCOPY procedure.

To restart a program from a check point, the check point id must be passed on the call to XRST. This can now be achieved when SELCOPY Debug is started by specifying the value in the CKPTID> input field of the SELCOPY/Debug panels or the -CKPTID of the SELCOPY primary command.

ASMTDLI Call

SELCOPY Debug now intercepts a SELCOPY CALL operation that specifies ASMTDLI as the called program name and executes it as a sub-task. This matches the action taken by the debugger for SELCOPY's internal calls to ASMTDLI and so prevents a possible 0C4 program check that occurs in IMS/DL1 modules.

SELCOPY Debug - JCL Input

Using the SELCOPY Debug panels, input may reference an existing JCL batch procedure that is comprised of any number of job steps.

The debug panel procedure examines the supplied JCL input and creates work files that represent in-stream datasets and executable REXX procedures which correspond to each job step. The panel procedure then displays the Jop-Step Selection List panel which contains a table entry for each job step. Each job step may be deselected or selected for execution.

On executing the next selected job step in the table, all DDnames allocated by DD statements in the job step (or passed to it via DISP=PASS) are allocated in the TSO environment and the program specified by the EXEC statement executed as a sub-task. Each job step that executes program SELCOPY or SLC has an option to either run the program using the SELCOPY Debugger (the defualt) or as a sub-task.

When the program has completed (either as a sub-task or in SELCOPY Debug) the return code is set for that job step entry and the allocated datasets are freed. If the last job step executed the program in SELCOPY Debug, then processing pauses. Otherwise the next selected job step entry in the table is executed without user interaction.

SELCOPY Debug panel processing of JCL has been enhanced in SELCOPYi 3.40 to support the following:

IMS/DL1

Job steps that execute the IMS/DL1 region controller DFSRRC00 may now be executed as a sub-task. In previous releases, only IMS/DL1 DFSRRC00 job steps that executed program SELCOPY were supported and then use of SELCOPY Debug was mandatory.

JCL input which includes IMS/DL1 DFSRRC00 job steps to execute a program other than SELCOPY, may now be passed to the SELCOPY Debug panels and, where SELCOPY is the program used, the Debug selection field may be set to "N" (No Debug).

IMS/DL1 Return Code

The return code passed back to the Job-Step Selection List panel and displayed against the appropriate job step entry is now correct for job steps that execute the IMS/DL1 region controller, DFSRRC00.

Work File DSN Prefix

The work files generated by the SELCOPY Debug panels are saved for later reference using high level qualifier(s) equal to the user's standard DSN prefix. If this prefix has a length greater than 12, SELCOPY Debug no longer attempts to allocate a work file DSN of length greater than 44. Instead, work file DSN qualifiers are used which restrict the DSN length to 44.

In previous releases, the user would be prompted to specify different DSN prefix qualifiers.

Sequence Numbers

Sequence numbers in the last 8 columns of the fixed length (RECFM=F) JCL records are now correctly interpreted as such.

Job-Step Selection List Table Fields

The "Debug" column has been updated to display underscores to indicate that the field is enterable. Similarly, "Step Name" and "Info" columns have been made non-enterable.

SELCOPY Debug - Other Enhancements

Other minor changes to the SELCOPY Debug application windows area as follow:

DB2 Data Types

The SELCOPY DB2 interface, SELCOPQL, has been upgraded to support all available DB2 data types that are supported by DB2 v11. This includes TIMESTAMP data types with specified fractions of a second precision and time zone specification (see DB2 Version 11 - TIMESTAMP Datatypes under SELCOPYi Data Editor above).

Note that only the first 100 characters are fetched from XML and LOB columns.

Current Input Record Window

When the SELCOPY procedure does not specify a work area but includes input (READ) operations, then SELCOPY Debug "Current Input Record" storage window is opened to display records from the input buffer. The length of data displayed in this window now changes to reflect the length of the last record read.

Previously, the length of data displayed was fixed at the length of the first record read.

SELCOPY Primary Command

Syntax:

```
>>-- SELCopy ------+------------------------------+-----------------------><
                   |                              |
                   +-- | Command Line Options | --+


Command Line Options:

    +-- -Selcopy -+  +-- -PROGRAM ---- SELCOPY --+
    |             |  |                           |
 |--+-------------+--+---------------------------+------------------------>
    |             |  |                           |
    +-- -SLC -----+  +-- -PGM ------+- progname -+
      (1)            |              |
                     +-- -PROGram --+

 >--+-- -CTL ---+- ctl_filename ---+-------------------------------+------>
    |           |                  |                               |
    +-- -SYSIN -+                  +-- -LST ------+- lst_filename -+
                                   |              |
                                   +-- -SYSPRint -+

 >--+--------------------------------------+--+-----------------------+--->
    |                                      |  |                       |
    +-- -LIBrary -+-------- libdd ---------+  +-- | IMS/DL1 Parms | --+
                  |                        |
                  +-- ( --- libdd ---- ) --+
                  |                        |
                  |      +--- , ----+      |
                  |      v          |      |
                  +------+- libdsn -+------+
                  |                        |
                  |      +--- , ----+      |
                  |      v          |      |
                  +-- ( -+- libdsn -+- ) --+

 >--+----------------------+-+-----------------------+-+----------------+-|
    |                      | |                       | |                |
    | +------------------+ | | +-------------------+ | +-- parm_string -+
    | v                  | | | v                   | |  (2)
    +-+-- -XFREE ddname -+-+ +-+- -XDELete fileid -+-+


 IMS/DL1 Parms:

    +-- -DLI -------------------+
    |                           |
 |--+---------------------------+-- -PSB psbname ---+------------------+-->
    |                           |                   |                  |
    +-- -BMP -+-----------------+                   +-- -IMSid ssname -+
              |                 |
              +-- -AGN grpname -+
                 (3)

 >--+---------------------+-----------------------------------------------|
    |                     |
    +-- -CKPTid ckptname -+
```

Notes:

1. Parameter -SLC is default if -PROGRAM SLC is specified.
2. A SELCOPY/SLC paramater string, if present, must be specified after all keyword parameters.
3. Parameter -AGN is effective only for IMS 9.1 or below where AGN security is supported, otherwise the parameter is ignored.

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

- New **-SLC** and **-SELCOPY** keyword parameters which specify the SELCOPY language interpreter to be used.

  Parameter -SLC identifies the program source as being for the SELCOPY C++ version and implies a default program name (-PGM) of SLC. Parameter -SELCOPY or (-S) is default, identifies the program source as being for the SELCOPY Assembler (BAL) version and implies a default program name of SELCOPY.

- New **-XFREE** keyword parameter and *ddname* argument identifies an allocated DDname which is to be freed when the SELCOPY Debug session ends.

- New **-XDELETE** keyword parameter and *fileid* argument identifies the fileid (DSN) of a sequential or VSAM dataset, library member or HFS file which is to be erased when the SELCOPY Debug session ends.

- New **-CKPTID** keyword parameter and *ckptname* argument identifies the check point id to be passed to IMS/DL1 via the region controller (DFSRRC00). This is necessary when debugging a SELCOPY job which is restarted and executes a DL1 XRST (extended restart) call.

- Parameter string *parm_string* arguments follow all other parameter keywords and arguments on the primary command and are passed for input to the SELCOPY (or SLC) program.

  The SELCOPY program references these parameters via POS PARM. Additionally, the SLC program references parameters via POS ARG and %*n* substitution variables.

SELCOPY Debug - Primary Commands

The following SELCOPY Debug primary commands have been updated or included in SELCOPYi 3.40.

BREAKPOINT

Syntax:

```
                          +- TOGgle --+   +- Cursor ------------+
                          |           |   |                     |
>>--+- BReakpoint --+--+--+-----------+---+---------------------+--+------><
    |               |  |  |           |   |                     |  |
    +- BRK ---------+  |  +- ON ------+   +- line --+-----------+  |
                       |  |           |             |           |  |
                       |  +- OFF -----+             +- column --+  |
                       |  +- CLear ---+                            |
                       |                                           |
                       |                                           |
                       +--+- OFF -----+---+- ALL --+---------------+
                       |  +- CLear ---+   +- * ----+               |
                       |                                           |
                       +---- SUSPENDALL ---------------------------+
                       |                                           |
                       +---- RESUMEALL ----------------------------+
```

Existing SELCOPY Debug primary command BREAKPOINT has been enhanced to support the following:

- New **TOGGLE**, **ON** and **OFF** (synonym **CLEAR**) keyword parameters specify whether a new break point is set (ON) or an existing break point is unset (OFF/CLEAR) at a particular SELCOPY input control statement.

  Parameter TOGGLE will set or unset a break point at a specific control statement whichever is the alternate option to the current break point status.
- New **ALL** keyword parameter or its synonym "*" (asterisk) is applicable to the OFF (CLEAR) option only and indicates that all existing break points are to be unset.
- New *line* and *column* integer arguments specify the line and column number within the control file Text Edit view at which the control statement is located. This control statement is the target of the BREAKPOINT command.

  CURSOR is default, indicating that the control statement at the cursor position is the target of the BREAKPOINT command. If the cursor is not positioned within the window display area, the last known position of the cursor within the display area is used. If not positioned on a line containing a control statement, the first control statement within the display area is used.
- New **SUSPENDALL** keyword parameter indicates that all existing break points are to be temporarily suspended (de-activated) until a BREAKPOINT RESUMEALL is executed.
- New **RESUMEALL** keyword parameter indicates that suspended break points are to be re-activated.

LIST OPERATIONS

Syntax:

```
>>-- LIst --- OPerations -------------------------------------------------><
```

Description:

New in SELCOPYi 3.40, LIST OPERATIONS will display the Debug Operations List window which provides statistical analysis information for each executable SELCOPY statement.

RERUN

Syntax:

```
                   +--- CLear --+
                   +- NOKeep ---+
                   |            |
>>--+- RErun --+---+------------+-----------------------------------------><
    |          |   |            |
    +- RR -----+   +- NOCLear --+
                   +--- Keep ---+
```

Existing SELCOPY Debug primary command RERUN has been enhanced to support the following:

- New **CLEAR** (synonym **NOKEEP**) and **NOCLEAR** (synonym **KEEP**) parameter keywords indicate whether or not the SELCOPY list output Text Edit view is cleared when the program is rerun without closing the debug session. Default is CLEAR.

WATCH

Syntax:

```
                              +- * ------------------------------------+
                              |                                        |
>>-- Watch --+--------------+-+----------------------------------------+--><
    (1)      |              | |                                        |
             +--------+- : -+ +---------- varmask ---------------------+
             |        |       |                                        |
             +- name -+       +------ | SLC Field Definition | --------+
                              |      (2)                               |
                              |                                        |
                              +- Position - pos_expr -- | Data Type | -+
                              |                                        |
                              +- WORKAREA -----------------------------+

Data Type:

 |---+- ASCii -------+---------------------------------+--------------+---|
     |               +- ( n_bytes ) -------------------+              |
     |                                                                |
     +- BFP ------+--+---------------------------------+--------------+
     +- BINFloat -+  +- ( n_bytes ) -------------------+              |
     |                                                                |
     +- BINary ------+---------------------------------+--------------+
     |               +- ( n_bytes ) -------------------+              |
     |                                                                |
     +- Character ---+---------------------------------+--------------+
     |               +- ( n_bytes ) -------------------+              |
     |                                                                |
     +- CHARBin --+--+---------------------------------+--------------+
     +- CHABin ---+  +- ( n_bytes ) -------------------+              |
     +- CHB ------+                                                   |
     |                                                                |
     +- CHARVar --+--+---------------------------------+--------------+
     +- CV -------+  +- ( n_bytes ) -------------------+              |
     |                                                                |
     +- CString -----+---------------------------------+--------------+
     |               +- ( n_bytes ) -------------------+              |
     |                                                                |
     +- DFP ------+--+---------------------------------+--------------+
     +- DECFloat -+  +- ( n_bytes ) -------------------+              |
     |                                                                |
     +- DECimal -----+---------------------------------+--------------+
     |               |                                 |              |
     |               +- ( precision -+-----------+- ) -+              |
     |                               +- , scale -+                    |
     |                                                                |
     +- EBCdic ------+---------------------------------+--------------+
     |               +- ( n_bytes ) -------------------+              |
     |                                                                |
     +- HFP ------+--+---------------------------------+--------------+
     +- HEXFloat -+  +- ( n_bytes ) -------------------+              |
     |                                                                |
     |               +- (4) ---------------------------+              |
     +- PTR ------+--+---------------------------------+--------------+
     +- POINTER --+                                                   |
     |                                                                |
     +- VARChar --+--+---------------------------------+--------------+
     +- VChar ----+  +- ( n_bytes ) -------------------+              |
     |                                                                |
     +- Zoned -------+---------------------------------+--------------+
                     |                                 |
                     +- ( precision -+-----------+- ) -+
                                     +- , scale -+
SLC Field Definition:

                      + LENGTH - 32 --------+ +- TYPE ----- C -----------+
                      |                     | |                          |
 |-+ Pos -+ pos_expr -+---------------------+-+--------------------------+-|
   |      |           |                     | |                          |
   |      |           + LENGTH + len_expr  -+ +- TYPE -+-+- B -----------+
   |      |           + LEN ---+            | +- TY ---+ |               |
   |      |           + L -----+            |            |     +- NAT -+ |
   |      |                                 |            |     |       | |
   |      |                                 |            +- F -+-------+-+
   |      + pos_expr ---- pos_expr ---------+            |     |       | |
   |                                        |            |     +- HEX -+ |
   |                                        |            |     +- HEX -+ |
   |                                        |            |     +- HFP -+ |
   +----- len_expr -- AT -- pos_expr -------+            |     |       | |
                                                         |     +- BIN -+ |
                                                         |     +- BFP -+ |
                                                         |               |
                                                         +- P -----------+
                                                         +- Z -----------+
```

Notes:

1. If specified with no parameters, WATCH will insert an entry for every internal and user defined variable identified within the SELCOPY control statements.
2. SELCOPY Field Definition syntax is supported for SLC only.

Description:

New in SELCOPYi 3.40, WATCH may be used to insert new variable (*varmask*), work area (WORKAREA) or field definition entries in a Debug Watch List window immediately following the focus entry. A new entry will replace an existing entry of the same name if it already exists within the target Watch list.

A variable entry displays the value currently assigned to one or more named SELCOPY internal variables (e.g. LRECL, DIFF, RETCODE), user @pointer variables or SLC declared variables that match the specified variable mask. A variable mask may contain wildcard characters "%" (percent), representing any single character, and/or "*" (asterisk), representing zero or more characters.

If a work area is defined, WORKAREA is a special form of field entry displaying a character field starting at position 1 of the work area for a length equal to the defined work area length.

A field entry displays a field in storage and may be specified using either **SLC Field Definition** syntax (SLC debug only) or the **SELCOPY Debug** format (POSITION and Data Type) parameters.

Multiple Watch list windows may exist, each having a unique name. Therefore, the *name* parameter may be used to identify the Watch list instance to which the list entry will be added.

If *name*: is not specified, the list entry will be added to the current Watch list window (i.e. the last Watch list window to have the focus). If *name* does not match that belonging to an open Watch list window, then a new Watch list window is opened for *name* and the list entry added accordingly. If ":" (colon) is specified without *name*, the list entry will be added to a new Watch list window with a generated name in the format W*n* (where *n* is the next number in an internally maintained sequence.)

All Watch list entries are re-evaluated and the contents of the Value columns updated each time processing is paused by SELCOPY Debug. Therefore, any changes to the value of a variable or a variable on which a field position or length expression is based, will be reflected in the watch list Value columns.

If SELCOPY Debug is running with maximised debug windows, execution of WATCH will also place focus on the Watch List window. Otherwise, focus will remain on the window from which the command was executed.

WCOMMAND

Syntax:

```
>>--+- WCommand --+--+-------------+----- | Watch List Command | ---------><
    |             |  |             |
    +- WCMD ------+  +- name -- : -+
```

New in SELCOPYi 3.40, the WCOMMAND primary command will pass any command string to a Watch list window for execution, without first having to place focus on that window.

Multiple Watch list windows may be opened and so the *name* parameter is used to identify the Watch list instance to which the command will be passed. If *name* is not specified, the command will be passed to the current Watch list window (i.e. the last Watch list window to have the focus). An error occurs if no Watch list window exists with the specified name.

If SELCOPY Debug is running with maximised debug windows, execution of WCOMMAND will also place focus on the Watch List window. Otherwise, focus will remain on the window from which the command was executed.

WINDOW

Syntax:

```
>>-- WIndow --+--- @ ----------------+------------------------------------><
              +--- AT----------------+
              |                      |
              +--- Ctl --------------+
              |                      |
              +--- EQuates ----------+
              |                      |
              +--- List -------------+
              |                      |
              +--- PCB --------------+
              |                      |
              +--- POS expression ---+
              |                      |
              +--- SQL --------------+
              |                      |
              +--- Workarea ---------+
              |                      |
              +--- WTO --------------+
              |                      |
              +--- TRace ------------+
```

Existing SELCOPY Debug primary command WINDOW has been changed so that its minimum abbreviation is "WI". Previously, the minimum abbreviation was "W" which is now the minimum abbreviation for SELCOPY Debug command WATCH.

SELCOPY Debug - SET/QUERY/EXTRACT Options

The following SELCOPY Debug options have been included or updated in SELCOPYi 3.40.

BREAKIN

Syntax:

```
>>-+-----------+-- BREAKIn ----- n_execs ---------------------------------><
   |           |
   +- SET -----+

>>--- Query ------ BREAKIn -----------------------------------------------><

>>--- EXTract --- /BREAKIn/ ----------------------------------------------><
```

Description:

New in SELCOPYi 3.40, BREAKIN controls the number of times any operation may be executed (*n_execs*) before a virtual break point is encountered and processing is paused.

DEBUGCOLOUR

Syntax:

```
                                                             +- NONe ---+
                                                             |          |
>>-+--------+-+- DEBUGColour +-+- BReakpoint +-+ Blue -----+-+----------+-><
   |        | |              | |             | |           | |          |
   +- SET --+ +- DEBUGColor -+ +- OPeration -+ + Green ----+ +- BLInk --+
                                               |           | |          |
                                               + Pink -----+ +- REVerse +
                                               |           | |          |
                                               + Red ------+ +- Uscore -+
                                               |           |
                                               + Turquoise +
                                               |           |
                                               + White ----+
                                               |           |
                                               + Yellow ---+
                                               |           |
                                               + Default --+

>>--- Query ---------+- DEBUGColour -+------------------------------------><
                     |               |
                     +- DEBUGColor --+

>>--- EXTract --- / -+- DEBUGColour -+- / --------------------------------><
                     |               |
                     +- DEBUGColor --+
```

Description:

New in SELCOPYi 3.40, DEBUGCOLOUR controls the colour of break point and focus operation highlighting.

SELCOPY Debug - WATCH List Primary Commands

The following SELCOPY Debug Watch List primary commands have been introduced in SELCOPYi 3.40.

BOTTOM

Syntax:

```
>>-- BOttom -------------------------------------------------------------><
```

Description:

Display the last page of list data. BOTTOM is functionally equivalent to **DOWN MAX**.

DELETE

Syntax:

```
>>-- DELete -------------------------------------------------------------><
```

Description:

Delete (remove) the focus entry from the list (assigned to F17 by default). If the cursor is not position on an entry in the list, the focus entry is the first in the current display (this may not be the first entry in the list). Delete may also be actioned using line (prefix area) command "D" and, for groups of entries, a "DD" pair, "D*n*" or "D*".

DOWN

Syntax:

```
>>-- DOwn ---+-------------------+---------------------------------------><
            |                   |
            +-- Cursor ---------+
            +-- CSR ------------+
            |                   |
            +-- Data -----------+
            |                   |
            +-- Half -----------+
            |                   |
            +-- Max ------------+
            |                   |
            +-- Page -----------+
            |                   |
            +-- n_lines --------+
```

Description:

Scroll the view of entries within the watch list down towards the bottom of the list (assigned to F8 by default). Where no parameter is specified, the scroll amount will be the value specified in the **Scroll>** field in the top right corner of the window display.

Parameter keywords CURSOR, DATA, HALF, MAX and PAGE are the standard scrolling values that determine the number of lines to be scrolled. The *n_lines* argument explicitly specifies the number of lines to be scrolled. See the Data Editor DOWN command for details.

CLOSE

Syntax:

```
>>--+- END -----+--------------------------------------------------------><
    |           |
    +- CLOse ---+
```

Description:

Close the Watch List window (assigned to F3 by default). The list entries are preserved and are included in the watch list display if it is re-opened.

INSERT

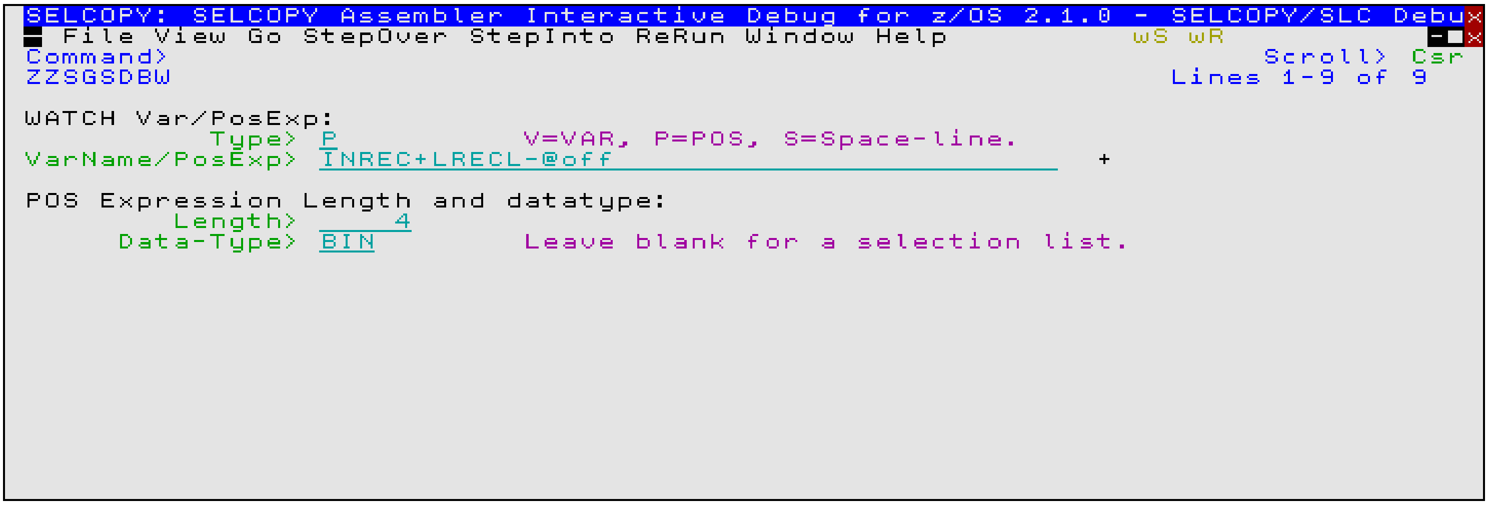
Syntax:

```
>>-- INSert ------------------------------------------------------------><
```

Description:

Open the **Debug - Add New WATCH Var/PosExp** panel to insert a new variable, POS *expression* or SPACE entry following the focus entry in the watch list (assigned to F18 by default).

The panel provides a dialog interface to the VARIABLE, POSITION and SPACE commands.



```
SELCOPY: SELCOPY Assembler Interactive Debug for z/OS 2.1.0 - SELCOPY/SLC Debu
  File View Go StepOver StepInto ReRun Window Help              wS wR
Command>                                                         Scroll> Csr
ZZSGSDBW                                                     Lines 1-9 of 9

WATCH Var/PosExp:
          Type> P          V=VAR, P=POS, S=Space-line.
VarName/PosExp> INREC+LRECL-@off                            +

POS Expression Length and datatype:
        Length>     4
     Data-Type> BIN         Leave blank for a selection list.
```

Figure 14. SELCOPY Debug: Add New WATCH Var/PosExp.

LEFT

Syntax:

```
>>-- LEft ---+-------------------------+--+------------------+----------><
             |                         |  |                  |
             +-- VARiable -- varname --+  +-- Cursor ---------+
             |                         |  +-- CSR ------------+
             +-- Workarea -------------+  |                  |
                                          +-- Data -----------+
                                          |                  |
                                          +-- Half -----------+
                                          |                  |
                                          +-- Max ------------+
                                          |                  |
                                          +-- Page -----------+
                                          |                  |
                                          +-- n_cols ---------+
```

Description:

For character data type entries only, scroll the contents of the Value column to the left (assigned to F10 by default). The Value column display in all other list entries, remains unchanged. Where no scrolling parameter is specified, the scroll amount will be the value specified in the **Scroll>** field in the top right corner of the window display.

Parameter keywords VARIABLE (with argument *varname*) and WORKAREA identify the particular watch list variable or workarea entry to be scrolled. If neither of these parameters are specified, the focus entry is scrolled. Note that the focus entry is the entry on which the cursor is located, otherwise the first entry in the current display.

Parameter keywords CURSOR, DATA, HALF, MAX and PAGE are the standard scrolling values that determine the number of lines to be scrolled. The *n_cols* argument explicitly specifies the number of columns to be scrolled. See the Data Editor LEFT command for details.

LOCATE

ISPF Syntax:

```
   (1)
>>---- Locate -----+--------------------- line_num -------------------+--><
                   |                                                  |
                   +--------------------- .label ---------------------+
                   |                                                  |
                   +--+- + (plus) --+---- relative_line --------------+
                   |  |             |                                 |
                   |  +- - (minus) -+                                 |
                   |                                                  |
                   +--------------------- string ---------------------+
```

XEDIT Syntax:

```
>>--+--------------+---- : -------------- line_num -------------------+--><
    |           |  |                                                  |
    +- Locate --+  +--------------------- .label ---------------------+
  (2)              |                                                  |
                   |  +- + (plus) --+                                 |
                   |  |             |                                 |
                   +--+-------------+---- relative_line --------------+
                   |  |             |                                 |
                   |  +- - (minus) -+                                 |
                   |                                                  |
                   +-------------------- /string/ --------------------+
                                        (3)
```

Notes:

1. Provided the locate operand is not *string*, the ISPF syntax LOCATE command verb may be omitted in which case the XEDIT form of the LOCATE syntax is used instead.

2. Provided the locate operand begins with a non-alpha character, the XEDIT syntax LOCATE command verb may be omitted.

3. A *string* delimiter is optional if the LOCATE verb is specified. If present it must be a special character in the following list: ¬`¦"£$%^_-+={}[]:;~'#<>|\,?/

   The terminating delimiter is optional but if present must be the same as the starting delimiter.

Description:

Locate and scroll to a watch list entry that matches the locate criteria.

The format of the LOCATE syntax is based on the prevailing value of the Text Edit option INTERFACE, as set in the input control statements (SYSIN) window view.

List entries may be located by:

- An absolute line number (*line_num*).
- An absolute line label name (*.label*) which was previously assigned to the line.
- A relative number of lines (*relative_line*) before (-) or after (+) the current list entry line.
- A character search string (*string*) which is matched by the value in the list entry's Name column.

If *string* is used, it is compared with the Name value in each list entry that follows the current entry until a match is found. If End-of-List is encountered, the search wraps to the first entry in the list and continues until either a match is found or the current list entry line is reached.

The LOCATE command verb may be omitted. However, its should be noted that the VARIABLE command will be executed instead if all the following conditions are true:

1. LOCATE command syntax is specified as parameters on the SELCOPY debug WATCH command.
2. The LOCATE command verb is omitted.
3. the locate operand matches the name of a SELCOPY variable.

POSWINDOW

Syntax:

```
>>-- POSWINdow -----------------------------------------------------------><
```

Description:

Opens a SELCOPY Debug storage window for the focus watch list entry (assigned to F16 by default).

The value in the Name column is passed as the *expression* on the implicit WINDOW POS command. If the Name column value is that generated for a position entry (WATCH POSITION), then the "POS_" prefix is stripped from the expression.

RESET

Syntax:

```
>>-- RESet --+-------------------+--+----------+-------------------------><
             |                   |  |          |
             +-- EXcluded -------+  +-- ALL ---+
```

Description:

Reset flags that affect the display of watch list entries. Support currently exists for the excluded entry flag only. Therefore, parameter keywords EXCLUDED and ALL perform equivalent operation (i.e. redisplay excluded list entries).

RIGHT

Syntax:

```
>>-- RIght --+-------------------------+--+-------------------+----------><
             |                         |  |                   |
             +-- VARiable -- varname --+  +-- Cursor ---------+
             |                         |  +-- CSR ------------+
             +-- Workarea -------------+  |                   |
                                          +-- Data -----------+
                                          |                   |
                                          +-- Half -----------+
                                          |                   |
                                          +-- Max ------------+
                                          |                   |
                                          +-- Page -----------+
                                          |                   |
                                          +-- n_cols ---------+
```

Description:

For character data type entries only, scroll the contents of the Value column to the right (assigned to F11 by default). The Value column display in all other list entries, remains unchanged. Where no parameter is specified, the scroll amount will be the value specified in the **Scroll>** field in the top right corner of the window display.

Parameter keywords VARIABLE (with argument *varname*) and WORKAREA identify the particular watch list variable or workarea entry to be scrolled. If neither of these parameters are specified, the focus entry is scrolled. Note that the focus entry is the entry on which the cursor is located, otherwise the first entry in the current display.

Parameter keywords CURSOR, DATA, HALF, MAX and PAGE are the standard scrolling values that determine the number of lines to be scrolled. The *n_cols* argument explicitly specifies the number of columns to be scrolled. See the Data Editor RIGHT command for details.

SPACE

Syntax:

```
>>-- SPace ---------------------------------------------------------------><
```

Description:

Used for aesthetic purposes only, SPACE inserts an empty line following the focus list entry (assigned to F20 by default).

TOP

Syntax:

```
>>-- TOP -----------------------------------------------------------------><
```

Description:

Display the first page of list data. TOP is functionally equivalent to **UP MAX**.

UP

Syntax:

```
>>-- UP -----+-------------------+---------------------------------------><
             |                   |
             +-- Cursor ---------+
             +-- CSR ------------+
             |                   |
             +-- Data -----------+
             |                   |
             +-- Half -----------+
             |                   |
             +-- Max ------------+
             |                   |
             +-- Page -----------+
             |                   |
             +-- n_lines --------+
```

Description:

Scroll the view of entries within the watch list upwards towards the top of the list (assigned to F7 by default). Where no parameter is specified, the scroll amount will be the value specified in the **Scroll>** field in the top right corner of the window display.

Parameter keywords CURSOR, DATA, HALF, MAX and PAGE are the standard scrolling values that determine the number of lines to be scrolled. The *n_lines* argument explicitly specifies the number of lines to be scrolled. See the Data Editor UP command for details.

WORKAREA

Syntax:

```
>>-- WORKAREA ------------------------------------------------------------><
```

Description:

If WORKLEN has been specified in the SELCOPY control statements, WORKAREA opens the watch list window (if not already open) and inserts a special position entry for the work area buffer following the focus list entry. The data type used is character with a length equal to the defined work area length.

SELCOPY Debug - Watch List SET/QUERY/EXTRACT Options

The following SELCOPY Debug Watch List options have been introduced in SELCOPYi 3.40. Note that the SET command keyword is optional unless the SET operation is a parameter of the SELCOPY Debug WATCH command, in which case it must be specified.

COLOUR / COLOR

Syntax:

```
                                                             +- NONe ---+
                                                             |          |
>>-+--------+-+- COLOur -+-+- COMMANDLine ---+-+ Blue -----+-+----------+-><
   |        | |          | |                 | |           | |          |
   +- SET --+ +- COLOr --+ +- COMMANDPrompt -+ + Green ----+ +- BLInk --+
                           |                 | |           | |          |
                           +- MESsage -------+ + Pink -----+ +- REVerse +
                           |                 | |           | |          |
                           +- OFFset --------+ + Red ------+ +- Uscore -+
                           |                 | |           |
                           +- OFFSETHilight -+ + Turquoise +
                           |                 | |           |
                           +- PREfix --------+ + White ----+
                           |                 | |           |
                           +- PREFIXCommand -+ + Yellow ---+
                           |                 | |           |
                           +- PREFIXHilight -+ + Default --+
                           |                 |
                           +- SCALE ---------+
                           |                 |
                           +- SHADow --------+
                           |                 |
                           +- TEXT ----------+
                           |                 |
                           +- TItle ---------+
                           |                 |
                           +- TOLeol --------+
                           |                 |
                           +- VALue ---------+
                           |                 |
                           +- VERror --------+

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

>>--- EXTract --- / -+- Colour -+- / -------------------------------------><
                     |          |
                     +- COLOr --+
```

Description:

New in SELCOPYi 3.40, COLOUR controls the colour of watch list display attribute highlighting. Attributes and their default highlighting are as follows:

Attribute	Default	Description
COMMANDLINE	GREEN NONE	Command line input text.
COMMANDPROMPT	BLUE NONE	Command and Scroll prompt.
MESSAGE	RED NONE	Message line text.
COLUMN	RED NONE	Col column offset value hilight.
PREFIX	GREEN NONE	Prefix area. (PREFIX ON)
PREFIXCOMMAND	RED NONE	Prefix area (line) command text.
PREFIXHILIGHT	RED NONE	Prefix area hilight.
SCALE	WHITE NONE	Scale line. (SCALE ON)
SHADOW	WHITE NONE	Excluded shadow line. (SHADOW ON)
TEXT	BLUE NONE	All other window text.
TITLE	WHITE USCORE	Column header text.
TOLEOL	WHITE NONE	Top of List/End of List lines.
VALUE	GREEN NONE	Value column text.
VERROR	RED REVERSE	Value column error text.

COLUMN

Syntax:

```
>>-+--------+---------- COLumn -----+-- Decimal ------+-------------------><
   |        |                       |                 |
   +- SET --+                       +-- Hexadecimal --+

>>--- Query ----------- COLumn -------------------------------------------><

>>--- EXTract --- / --- COLumn --- / -------------------------------------><
```

Description:

New in SELCOPYi 3.40, COLUMN controls whether the Value column displayed offset that appears in the Col column is displayed as a decimal or hexadecimal value.

DATATYPE

Syntax:

```
>>-+--------+---------- DATAType ---+-- ON -----------+-------------------><
   |        |                       |                 |
   +- SET --+                       +-- OFf ----------+

>>--- Query ----------- DATAType -----------------------------------------><

>>--- EXTract --- / --- DATAType - / -------------------------------------><
```

Description:

New in SELCOPYi 3.40, DATATYPE controls whether or not the DataType column is included in the watch list display.

HEX

Syntax:

```
>>-+--------+---------- HEX --------+-- ON -----------+-------------------><
   |        |                       |                 |
   +- SET --+                       +-- OFf ----------+

>>--- Query ----------- HEX ----------------------------------------------><

>>--- EXTract --- / --- HEX ------ / -------------------------------------><
```

Description:

New in SELCOPYi 3.40, HEX controls whether or not the Value column includes the value's source field displayed in hex. Two additional lines displaying hex digits in up/down notation are included for each list entry.

POINT

Syntax:

```
                                    +- ON ------------+
>>-+--------+--- Point --- .name ---+-----------------+-------------------><
   |        |                       |                 |
   +- SET --+                       +-- OFf ----------+

>>--- Query ------- Point -- * -------------------------------------------><

>>--- EXTract ---- /Point -- * -- / --------------------------------------><
```

Description:

New in SELCOPYi 3.40, POINT will unassign or assign a label name to a line in the Watch list window.

PREFIX

Syntax:

```
>>-+--------+-------- PREfix -----+-- ON ---+--+---------+--+-----------+-><
   |        |                     |         |  |         |  |           |
   +- SET --+                     +-- OFf --+  +- Left --+  +- n_bytes -+
                                               |         |
                                               +- Right -+

>>--- Query ----------- PREfix -------------------------------------------><

>>--- EXTract --- / --- PREfix --- / -------------------------------------><
```

Description:

New in SELCOPYi 3.40, PREFIX defines whether the prefix area is displayed, whether it is displayed on the left or right of the window view and the number of columns it occupies.

SCALE

Syntax:

```
>>-+--------+---------- SCALe ------+-- ON -----------+-------------------><
   |        |                       |                 |
   +- SET --+                       +-- OFf ----------+

>>--- Query ----------- SCALe --------------------------------------------><

>>--- EXTract --- / --- SCALe ---- / -------------------------------------><
```

Description:

New in SELCOPYi 3.40, SCALE controls the display of a counting scale line above a character type value in the Value column. SCALE ON and OFF applies to the focus watch list entry but will do nothing if this entry is of numeric or hexadecimal data type.

SHADOW

Syntax:

```
>>-+--------+---------- SHADow -----+-- ON -----------+-------------------><
   |        |                       |                 |
   +- SET --+                       +-- OFf ----------+

>>--- Query ----------- SHADow -------------------------------------------><

>>--- EXTract --- / --- SHADow --- / -------------------------------------><
```

Description:

New in SELCOPYi 3.40, SHADOW controls the display of a shadow line in place of one or more watch list entries that have been excluded from the display.

SELCOPYi Miscellaneous Enhancements

Additional miscellaneous enhancements included in SELCOPYi 3.40 are documented here.

Diagnostic Trace Facility

To assist with SELCOPYi problem determination, load modules ZZSCTRAC and ZZSATRAC (alias CBLATRAC and APEATRAC respectively) are included with SELCOPYi 3.40 to support the SELCOPYi TRACE facility.

The trace facility is primarily used to generate formatted and/or unformatted trace output for trouble shooting internal SELCOPYi issues and, therefore, should only be started if requested to do so by CBL support. The trace facility is started and stopped using SELCOPYi primary command, TRACE.

Syntax:

```
                               (1)                (2)
                            +- -PGM APEATRAC -+  +- -DSN SYSLST ---+
                            |                 |  |                 |
>>-- TRACE ----+-- ON ---+--+-----------------+--+-----------------+------><
               |         |  |                 |  |                 |
               +-- OFF --+  +- -PGM CBLATRAC -+  +- -DSN outdsn ---+
```

Notes:

1. Default -PGM program name is APEATRAC unless INI option variable TRACE.PROGRAM is assigned. If so the value of option TRACE.PROGRAM is used as the default.
2. Parameter -DSN is mandatory for z/OS systems unless INI option variable TRACE.DATASET is assigned. If so the value of option TRACE.DATASET is used as the default. For z/VSE systems, output trace records are always written to SYSLST.

Description:

TRACE ON will start output of SELCOPYi trace records to the nominated output dataset, whose DSN is specified as *outdsn*. Furthermore, if DDname CBLILIBT is allocated, additional trace records, relating to SELCOPYi's internal function calls to library activation/deactivation, are written to the dataset allocated to CBLILIBT.

Both *outdsn* and CBLLIBT must represent existing, physical sequential datasets which have been allocated as LRECL=256, RECFM=VB.

TRACE OFF will stop output of SELCOPYi trace records.

Alternate Key Lists

For each window view in the SELCOPYi environment, there exists a standard list of PFKey definitions that may be tailored for local installation wide use or for use by an individual user.

For most SELCOPYi application windows, the limit of 24 PFKey settings is sufficient. However, for applications that support a large number of primary commands (like the Text and Data Editors), this limit may be restrictive. Furthermore, it may be desirable to use local, 3270 emulator keyboard macros to assign edit functions to non-PFkey key strokes. For example, you may wish to assign the Ctl-S key combination to perform the SAVE command.

To overcome the 24 PFkey restriction and support 3270 keyboard macro development, SELCOPYi 3.40 introduces the ability to switch between pre-defined application keylists. This involves an enhancement to primary command KEYLIST to support PUSH and POP parameters.

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

Syntax:

```
>>-- KEYLIST --+-------------------------------------+--------------------><
               :                                     :
               +---- POP ----------+--------------+--+
               |                   |              |  |
               |                   +-- command ---+  |
               |                                     |
               +--+- PUSH -----+---+--------------+--+
                  |            |   |              |
                  +- PUSHFka --+   +-- key-list --+
```

Description:

KEYLIST PUSH (or PUSHFKA) and POP may be assigned to PFKeys to support simple switching between keylist definitions.

PUSH sets the named key-list for the focus window view but, before doing so, the active keylist name is saved. **PUSHFKA** performs the same action as PUSH but does not automatically refresh the function key display area at the bottom of the screen (as displayed by command **PFSHOW**).

POP will restore the keylist saved by the last PUSH or PUSHFKA. Optional POP parameter *command*, representing any supported primary command, may be specified so that the reset of the keylist occurs only after *command* has been executed.

For an example of KEYLIST use in conjunction with 3270 emulator keyboard macros, see the latest edition of the "*SELCOPYi Reference and User Guide*" manual or SELCOPYi online help for primary command KEYLIST.

Window Positioning

SELCOPYi 3.40 introduces support for new primary commands that reposition the focus window view so that it is justified relative to the limits of the available screen display area (ISPF logical screen size). The window size is not affected.

JTop	Justify Top. The window is positioned so that its top border occupies the first row of the client display area.
JBottom	Justify Bottom. The window is positioned so that its bottom border occupies the last row of the client display area.
JLeft	Justify Left. The window is positioned so that its left border occupies the first column of the client display area.
JRight	Justify Right. The window is positioned so that its right border occupies the last (right most) column of the client display area.

Miscellaneous Updates

SELCOPYi 3.40 includes the following minor changes and enhancements:

Editors	Text Editor and Data Editor status bars now display the cursor's last line and column position within the file area of the window display. i.e. the CURSOR HOME location.
Editors	Text Editor and Data Editor LIST command updated to support type CLD which lists members in a concatenated library directory (DDname).
General	Display expiry information in the "ABOUT" product information panel.
General	Support INFO as a synonym for DSI in all window classes.
General	Unrecognised commands now passed to TSO if ISPF OFF is set.
General	Primary command PFSHOW ALL supported to display all key values with maximum 12 entries per line.
General	MDIPREV/MDINEXT now skips hidden windows.
Install	Modifications and corrections to text in the install document and in CMX files: FIRSTUSE and VSEINST.
Lists	For list windows only, enclosing quotation marks or apostrophes are no longer necessary on a character string specified as the second term of a WHERE expression. e.g. WHERE DSORG = PS
Lists	The contents of List window Yes/No flag fields are now centralised beneath the column header and display "Y" if on or "-" (hyphen) if off. Previously displayed "Y" or "N".
Panels	Prevent vertical scrolling when there are no further panel lines to be displayed.
Panels	Default initial sort order for entries in list and table child windows.
Panels	Support a DDName in place of a dataset name or library member specification in relevant panel fields.
Panels	Display "No matching entries" message if no entries match the wildcard mask provided in a field for which a pop-up list selection panel would otherwise be displayed.
Startup	FIRSTUSE SITE ammended to report error conditions and conflicts that may occur based on selection of install library HLQ.
Startup	SELCOPYi start up elapsed time reduced significantly.
Utility	Improved efficiency in execution of CSVGEN, XMLGEN or PRINT when run against a very large dataset within the current Data Edit view with parameter .ZLAST specified.
Utility	Updates to the save and redisplay of an FSU report. Particualrly with relation to DSN and DSORG assigned to the FSU report structure (SDO).
Utility	EXECSQL utility now truncates blanks that follow the last SQL statement, so reducing the size of the output SQL execution report.
Utility	RUNSELC (SELCOPY execution) and RUNSLC (SLC execution) utilities updated so that the temporary SYSPRINT output dataset, displayed in a text edit view on completion, is now assigned RECFM=VBA, LRECL=1024. This allows display of long error messages and user printed text.

Maintenance Applied

The following table identifies maintenance to SELCOPY Product Suite 3.30 that has been applied at source to SELCOPY Product Suite 3.40.

Details of each SYSMOD may be found at the CBL web page entitled "SELCOPY Product Suite 3.30 Maintenance Summary". Individual descriptions are referenced by links to this web page in the following table.

Service Package Id	SYSMOD
X2016032	RS33001, RS33002, RI33001
X2016144	RS33003, RI33002
X2016162	RS33004