Compute (Bridgend) Ltd

Product Suite New Features

Release 3.50

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\snf350\SELCOPY_Product_Suite_3.50_New_Features.pdf

Contents

Contents

SELCOPY Product Suite 3.50 New Features

Documentation Notes

Information in this New Features List details changes introduced to CBL software since SELCOPY Product Suite 3.40.

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 **www.cbl.com/selcdl.php**.

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

- SELCOPY Product Suite Customisation Guide
- SELCOPY User Manual
- SELCOPY C++ (SLC) Language Reference
- SELCOPY SLCIMS Call Module Interface for IMS/DL1 Checkpoint/Restart
- CBLVCAT User Manual
- SELCOPYi Reference and User Guide
- SELCOPYi Text Editor (CBLe) Manual
- SELCOPYi Structured Data Editor Manual
- SELCOPYi Quick Reference
- SELCOPYi REPORT Utility
- SELCOPYi SMF Utilities
- 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 3.50 includes updates to the following product components.

SELCOPYi

SELCOPYi has been updated extensively for z/OS systems. The significant new features in SELCOPYi 3.50 for z/OS are:

- Support for z/OS PDSE member generations.
- Report Generation Utility.
- z/OS SMF record extraction, formatted display and report generation.

The remainder of this document contains details of these features and other changes introduced in SELCOPYi 3.50.

CBLVCAT

The CBLVCAT 3.50 batch program has undergone no changes since version 3.40 and these versions are functionally equivalent.

SELCOPY

The SELCOPY 3.50 (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 RI34001 (s340z01)
Applicable to z/OS systems only, correct immediate end-of-file which is returned on READ of GDG generation (+1) which has been dynamically allocated by SELCOPY. This same problem exists in previous releases of SELCOPY.

Ref: APAR RI34001 (s340z02)
Applicable to z/VSE systems only, correct missing display of the BLKSIZE value for an output VSE sequential file in the SELCOPY SUMMARY block of the SYSLST output. This problem is caused by APAR fix QI32001 to SELCOPY release 3.20.

Ref: APAR RI34001 (s340z03)
Applicable to z/OS systems only, correct DB2 result table row length (LRECL) returned following a DB2 READ operation. This problem was introduced by PTF RI33002 to SELCOPY release 3.30.

Ref: APAR RI34002 (s340z04)
Correct an 0C4 program check which occurred at end-of-job processing when the last input or output dataset is a VSAM KSDS with a key position defined at an offset greater than 2025. This problem exists in all previous releases of SELCOPY.

Ref: APAR RI34003 & RI34004 (s340z05, s340z06 & s340z07)
Introduce support for I/O on z/OS variable spanned data sets (RECFM=VS or VBS) where LRECL is greater than 32760 (the maximum for BUFTEK=A or BUILDRCD use).

Previously, SELCOPY supported I/O on variable spanned data sets using the BUFTEK=A method. With this SYSMOD applied, SELCOPY will support I/O on spanned records of any length. QSAM XLRI (LRECL=X) will be used where a data set is allocated with LRECL>32760 and RECFM=VS or VBS.

Note that the QSAM BUFTEK=A method must be used if data set is to be opened for update (READ UPD). i.e. Input for update is only possible for spanned data sets with LRECL<=32760.

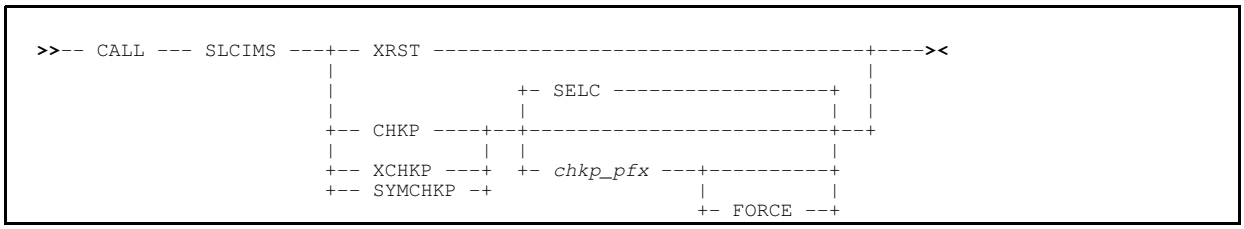
Ref: APAR RI34004 (s340z08)
Correct CEE3608I/CEE3611I/CEE3606I errors that occurred when CALLing a COBOL program that has been compiled using the Enterprise COBOL Version 5.0 compiler and above.

Ref: APAR RS34001
Introduce load module **SLCIMS** which allows SELCOPY to perform IMS/DL1 extended checkpoint/restart (CHKP/XRST) calls in jobs that process IMS/DL1 databases.

SLCIMS is called via the SELCOPY CALL operation to perform one of the following IMS DL1 function calls when SELCOPY is executed via the IMS batch region controller (DFSRRC00):

◊ A basic checkpoint (CHKP)
◊ An extended restart (XRST)
◊ An extended/symbolic checkpoint (CHKP)

Syntax:



Features:

1. Operational in DLI or BMP regions.
2. Supports PSBs generated for all compiler languages (PSBGEN LANG=COBOL, PL/I or ASSEM).
3. Restores current position in all IMS databases being processed by SELCOPY following a checkpoint call.
4. Extended restart (XRST) call supports restart of a SELCOPY job from a checkpoint location. The following SELCOPY data areas are restored:
   - The user work area buffer.
   - For each open IMS DL1 database connection, the input and update segment counts, current and longest segment lengths, file size value and end-of-file status flag.
   - The value assigned to each user @ pointer variable.
5. Support for BMC Application Restart Control (AR/CTL) FORCEd check point to override its checkpoint pacing rule.
6. POS PCB is updated to point at the I/O PCB for the CHKP/XRST call. Therefore, POS STATUS will reference the CHKP/XRST status code.
7. Following a CHKP call, the system @ pointer references the SLCIMS return object structure which contains the 8-character checkpoint identifier and status information on the Get Unique (GU) call performed on each database object in order to re-establish position.
8. POS RETCODE and RETXV values indicate success or otherwise of the SLCIMS call.

For further details and examples of use, please refer to the "*SELCOPY IMS Call Module*" publication.

SELCOPY (SLC) C++ Version

The SELCOPY (SLC) 3.50 C++ Version batch program has been updated with new features which are detailed as follows:

SELCOPY for x86-64 Linux
A 64-bit version of SELCOPY now exists for operation on Linux operating systems compiled for x86-64 (64-bit) architecture.

Use of the 32-bit SELCOPY for x86 Linux on a 64-bit Linux system requires install of 32-bit libraries for legacy software. These libraries are not required if SELCOPY for x86-64 Linux is installed.

Internal Field Definitions
For **64-bit Linux only:** Internal field definitions **UXADIFF**, **UXATPTR**, **UXINCNT** and **UXLRECL** are defined as 8-byte, binary value fields in big-endian format. In all other SELCOPY versions, these fields are defined as 4-byte binary value fields in big-endian format.

CHANGE Match Position/Length
The **CHANGE** operation now includes support for **PTR** and **MATCHLEN** (or **MLEN**) operands. The position of the **first** occurrence of the CHANGE ***search string*** found within the field will be assigned to the specified PTR @variable and its length assigned to the @variable or declared variable specified by MATCHLEN.

Note that, use of MATCHLEN is useful if the length of the search string is variable (i.e. if the search string is specified as a regular expression).

Regular Expressions
Regular expressions, which may be used in IF/AND/OR and CHANGE operations, now support additional *predefined_expression* specifiers "**:h**" (hexadecimal numbers), "**:i**" (C-language identifiers) and "**:n**" (numbers with or without a decimal point).

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.

SELCOPYi and PDSE V2 Member Generations

IBM z/OS 2.1 introduced PDSE version 2 (V2) libraries which include support for member generations (via the MAXGENS library data set allocation attribute).

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

IBM publication "*z/OS DFSMS Using Data Sets*" provides information on PDSE version 2 libraries and member generations.

SELCOPYi 3.50 extends support for PDS/PDSE libraries to include PDSE V2 libraries and member generations.

Member Generation Reference

When processing a PDSE version 2 library which supports mamber generations, SELCOPYi panel fields and primary commands that identify a library mamber name now also support reference to an individual member generation.

To reference an individual member generation, SELCOPYi supports specification of an absolute or relative generation number following the member name with a single separating "." (dot/period) between. A member generation may be identified by its relative or absolute generation number. e.g.

Using a relative generation number...

```
EDIT    NBJ.GEN25.XEC(APESUB.-5)
BROWSE  NBJ.SELCI.SDO(DB2FUNC1.-1)
```

Using an absolute generation number...

```
E       NBJ.GEN25.XEC(APESUB.56)
VIEW    NBJ.SELCI.SDO(DB2FUNC1.7)
```

Note that blank characters are not permitted within a member generation specification.

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

SELCOPYi allows reference to the base generation via its relative (0) or absolute (SELCOPYi attributed) generation number. Alternatively, the generation specification may simply be omitted as referencing the member name only will identify the base generation.

e.g. Where 12 previous generations have been created for member DB2FUNC1, the base generation may be referenced in SELCOPYi as DB2FUNC1.13. The following are equivalent:

```
VIEW    NBJ.SELCI.SDO(DB2FUNC1.13)
VIEW    NBJ.SELCI.SDO(DB2FUNC1.0)
VIEW    NBJ.SELCI.SDO(DB2FUNC1)
```

Member Generation Mask

SELCOPYi panel fields and commands that support specification of a member name mask to identify multiple library members now also support specification of a member generation mask. This mask may identify multiple generations of one or more member names.

A member generation mask is the same as a single member generation but with either:

1. A null or wildcard character "*" specified in place of the generation number. This indicates that all generations are to be selected.
2. A relational operator inserted between the "." (dot/period) and the (absolute or relative) generation number. This indicates that only generation numbers that satisfy the numeric comparison will be selected.

   Supported relational operators are:

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

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

e.g. The following are equivalent and will open a library list of member generations window to display all generations of all members:

```
LL    NBJ.JCLLIB(*.*)
LL    NBJ.JCLLIB(*.)
LL    NBJ.JCLLIB(*.<*)
```

e.g. To list all member generations whose member name begins with CALL and whose absolute generation number is greater than or equal to 12:

```
LL    NBJ.JCLLIB(CALL*.>=12)
```

e.g. To erase all generations except the base (i.e. relative generation 0):

```
ERASE NBJ.JCLLIB(NBJX.<0)
```

e.g. To list all member generations whose member name begins with SS and whose relative generation number is -1:

```
 LL   NBJ.JCLLIB(SS*.-1)
```

e.g. List all member generations whose entries match one of the member and generation masks:

```
 LL   NBJ.JCLLIB(SS*.>=-3  ADA%%%%.>=-5  CBL*.0)
```

SELCOPYi Utilities Member Generation Support

Support for PDSE version 2 library, member generation and member generation mask specification has been implemented for SELCOPYi as follows:

PDSE V2 Library Data Set Information

Data Set Information panel output now displays organisation type PDSE V2 for PDSE version 2 library data sets. Panel is opened via the primary command DSI (or INFO) or list window prefix command "I".

```
SELCOPY/i - Dataset Information - Non-VSAM (Page 1 of 3 )
 File Report                                          wS wR
Command>                                                  Scroll> Csr
                                                        Lines 1-21 of 21

Dataset: CBL.CBLI350.CBLE
Catalog: USERCAT.CBLCAT

General Data:                                  Current Allocation:
 Volume:                      CBLM15            Number of Tracks:          500
 Device Type:                 3390              Number of Extents:           9
 Organisation:                PDSE V2 GENS      Number of Volumes:           1
 Record Format:               VB                Maximum Pages:            6000
 Record Length:                  256
 Block Size:                   27998            Current Utilisation:
 First Extent Tracks:            100            Number of Members:         220
 Secondary Tracks:                50            Used Pages:               5384
 Max Member Generations:          10            Percent Used Pages:        90%
                                                Used Extents:                9

Dataset Date Information:                      SMS Classes:
 Created:                  2020/01/24            Storage:                CBLDFLT
 Expires:                                        Data:
 Last Referenced:          2020/06/02            Management:             CBLDFLT
```

Figure 1. z/OS Data Set Information Display.

PDSE V2 Library Allocation

Allocate NonVSAM dialog window has been updated to support allocation of a new PDSE version 2 library, optionally with member generations.

The dialog now includes field entries "Version>" (PDSE library version 1 or 2) and "Member Generations>" (PDSE V2 MAXGENS value). It may be opened using primary command ALLOC with no parameters or opened automatically by FSU or FCOPY when output is to a new DSN. Note that command ALLOC already supports parameters DSNTYPE(LIBRARY,2) and MAXGENS.

If the "Model>" input field is used to model the new dataset attributes on that of an existing PDSE version 2 library, then the "Version>" and "Member Generations>" input fields will update to reflect the model dataset values.

PDSE V2 Library Member Generations List

Library Member Generations list window introduced to display an entry for each generation that matches a supplied member and member generation mask. The list has the same columns as a Library Member list but with additional **GenA** and **GenR** columns to display absolute and relative generation numbers respectively.

The list may be opened via the following:

1. Execute the LL (ListLibrary) primary command with a member generation or member generation mask parameter specification. i.e.

   ```
   LL <libname>( <mbrmask>.<genmask> ...  )
   ```

2. Execute line command "G" against an entry in a VTOC, Library Member or Dataset type list. All generations will be displayed for the PDSE library or member name list entry.

3. From a window view displaying the contents of a member generation (i.e. a Text Editor or Data Editor view), execute one of the following primary commands:

   GEN LIST
   Opens a Library Member Generations list displaying an entry for each generation of the member in view.

   GEN ORPHAN or **GENORPH**
   This opens a Library Member Generations list displaying an entry for each orphaned generation in the same library as the member in view. An orphaned generation is one where the member base generation (generation 0) has been deleted or renamed.

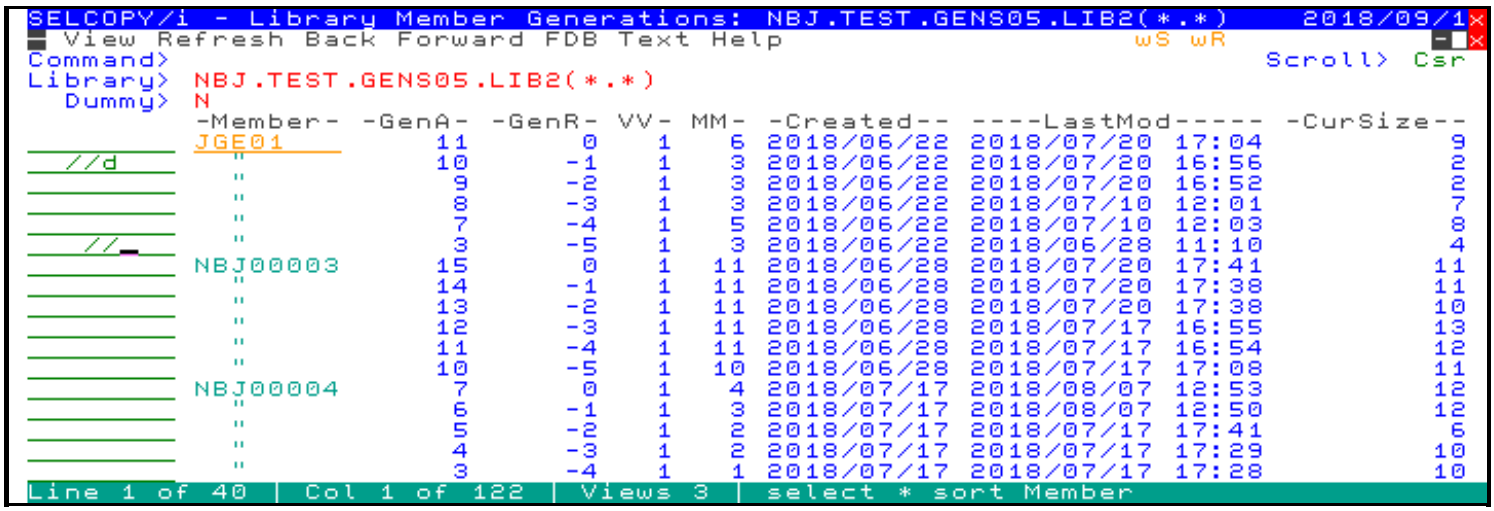


Figure 2. z/OS Library Member Generations List window.

Member Generation File Copy

Member generation masks may be used to select individual generations of one or more members to be copied from a PDSE V2 library to a target file. This target file may be an HFS/ZFS file, a VSAM, sequential or GDG dataset, a PDS or PDSE library or an individual library member.

Copy of member generations may be performed via the following methods:

1. Open the "File Copy" panel and enter a member generation mask in the "Member Mask>" field.
2. Execute FCOPY primary command with a member generation mask source data object specification. e.g.

```
FCOPY NBJ.JCLLIB(*.*)   NBJ.JCLLIB.COPY
```

3. Execute line command "C" against entries in a Library Member Generations list. Groups of list entries may be copied using "//C" and "//" line command pairs.

Copy of multiple member generations will be in ascending alphabetical order of selected member name and ascending order of absolute generation number. Thus, the oldest generation of a member is copied first.

The format of the copied data is dependent upon the organisation of the target data object as follows:

1. If copy is to a single target dataset, library member or HFS/ZFS file then the records in each selected member generation will be appended to the target file with optional member delimiter lines.
2. If copy is to a library data set which does **not** support member generations, then the target library will contain individual members with names matching those in the member generation mask. Each of these members will contain only data records from the newest member generation to match the member generation mask.
3. If copy is to a library data set which supports member generations, then the target library will contain an individual member generation for each generation selected by the member generation mask.

   Note that generation numbers in the target library may not match those selected from the source library by the generation mask. The internal management of generation numbers by the target PDSE will determine the assignment of the next generation number used and cannot be controlled by application programs.

   Since the target library supports generations, each copy to an existing member name will create a new generation for that member in the target library. Because each generation is copied in ascending order the generation hierarchy is preserved in the target library.

   Orphaned member generations (i.e. member generations for which the base generation has been deleted or renamed) may also be copied ultimately creating a new base generation for the member in the target library. File Copy recognises when a source library member's generations are orphaned and so deletes that member's base generation in the target library. This makes it the -1 generation and so orphans the generations in the target library matching those in the source.

   This type of copy is especially useful if a library needs to be redefined with a greater (or fewer) number of generations.

If a member generation mask has been supplied or entries have been selected for copy from a Library Member Generation list, then the SELECT command in the FCOPY or Library Member Copy panel will display a table of library member generations instead of library members.

Member Generation Delete

Delete (Erase) may be performed for a single generation or on multiple member generations that match a member generation mask.

Erase of member generations may be performed via the following methods:

1. Execute ERASE primary command with a member generation or member generation mask specification.

   e.g. To erase a relative generation -5 of member SSCALL:

```
ERASE  NBJ.JCLLIB(SSCALL.-5)
```

   e.g. To erase all generations of member SSREL other than the base member:

```
ERASE  NBJ.JCLLIB(SSREL.<0)
```

   e.g. To erase generations of all members that are older than relative generation -3:

```
ERASE  NBJ.JCLLIB(*.<-3)
```

2. Execute line command "D" (or "K" if no confirmation prompt is required) against entries in a Library Member Generations list. Groups of list entries may be deleted using "//D" and "//" (or "//K" and "//") line command pairs. Like library member delete, by default a prompt for each delete of a member generation will be displayed.

Note: When delete is performed without a member generation specification on a member of a PDSE library that supports generations, then only the base member generation will be deleted. To delete all generations of a member, a generation mask of "*" should be specified.

Member Generation Recovery

A specific member generation may be recovered so that it becomes the base member (generation 0). Following recovery, the relative generation number of each generation newer than the recovered generation is decremented by 1 so that the original base member becomes generation -1, etc.

Recovery of a member generation may be performed via the following methods:

1. Execute RECOVER primary command with an explicit member generation specification. e.g.

```
RECOVER  NBJ.JCLLIB(SSCALL.-1)
```

2. Execute line command "RC" against the non-base generation entry to be recovered in a Library Member Generations list.
3. From a window view displaying the contents of a member generation (i.e. a Text Editor or Data Editor view) which is not the base generation, execute **GEN RECOVER** to recover the base from the generation in view.

Member Generation Search

Member generation masks are supported as input to the FSU (File Search and Update) utility. Output to a member generation is not valid.

Data in member generations selected by the mask may be filtered, searched, updated, copied and remapped as supported for library members.

Use of the File Search and Update utility to process member generation input may be started via the following methods:

1. Open the FSU (Basic or Extended) File Search panel views and enter a member generation mask in the Member Mask field.
2. Execute FSU primary command with a member generation mask on the INPUT parameter specification.

   e.g. To report records in all generations of all members that contain the string "Lev":

```
FSU    INPUT('NBJ.JCLLIB(*.*)')     FIND(C'Lev')
```

3. Execute line command "F" against an entry in a Library Member Generations list.

Member Generation Compare

Member generations may be used as one or both of the file specifications in execution of the Compare Files utilty.

A compare of files involving a member generation may be started via the following methods:

1. Specify a member generation in the NEW and/or OLD file "Gen>" input field of the "Compare Files (Basic or Extended) Option" panel (=7.1). Alternatively, specifiy a member generation mask to select from a list of generations.
2. Execute the COMPFILE primary command with a member generation value in the NEW and/or OLD input fileids. e.g. To compare the base generation of member SSREL with its previous generation.

```
COMPFILE   NBJ.JCLLIB(SSREL)    NBJ.JCLLIB(SSREL.-1)
```

3. Execute line command "CF" against an entry in a Library Member Generations list.
4. From a window view displaying the contents of a member generation (i.e. a Text Editor or Data Editor view), execute **GEN COMPARE** or **GENCOMP** to compare the contents of the member generation in view with another generation of the same member.

Member Generation Text Edit/View

The Text Editor may be used to EDIT or VIEW (i.e. edit read-only) a member generation.

The absolute and relative generation numbers of the member generation in the focus text edit window view is displayed in the window title bar. e.g. G=10(-1)

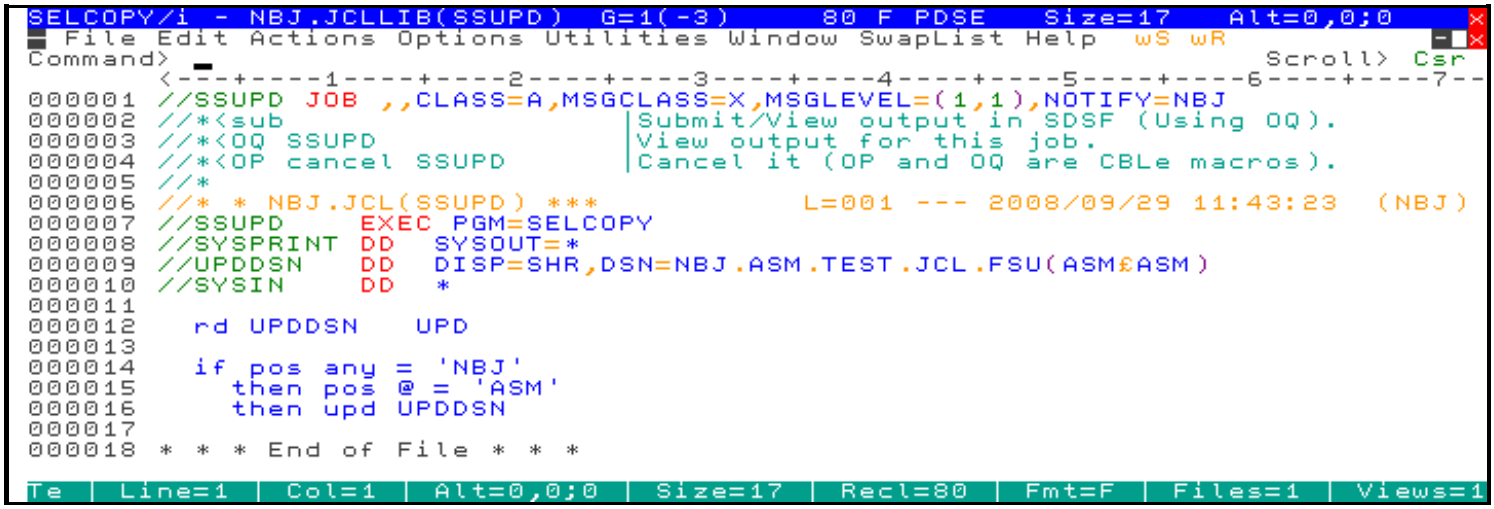


Figure 3. z/OS Library Member Generation Text Edit window.

Edit or view of a member generation may be started via the following methods:

1. Specify a member generation value in the "Gen>" input field of the "Text Edit Entry" panel (=1). Alternatively, specifiy a member generation mask to select from a list of generations.
2. Execute the EDIT or VIEW primary command with a member generation value specification. e.g.

```
EDIT   NBJ.JCLLIB(SSCALL.-5)
VIEW   NBJ.JCLLIB(SSREP.23)
E      NBJ.JCLLIB(SSREP)          (Generation=0)
```

3. Select (position the cursor and hit Enter) an entry from a Library Member Generations list.
4. Execute line command "E" or "V" against an entry in a Library Member Generations list.
5. From a window view displaying the contents of a member generation (i.e. a Text Editor or Data Editor view), execute **GEN** or **GEN EDIT** to VIEW or EDIT another generation of the same member respectively.

Edit of a member generation places an exclusive SYSEDIT ENQ on the member name and so it is not possible to open for output more than 1 generation of the same member at the same time.

When saving changes to a member generation, the prevailing value of the **GENSAVE** option dictates whether a new generation is created (NEWGEN), the current generation is updated (NOGEN) or a popup window is opened to prompt the user for the action to be taken on save (PROMPT). AUTO is the default and will perform SAVE NEWGEN for the member base generation, or SAVE NOGEN for older generations.

"**SAVE (NOGEN**" or "**SAVE (NEWGEN**" may be executed to specifically save the data back to the current generation or to create a new generation respectively.

If a fileid is also specified on the SAVE, then the NOGEN/NEWGEN parameter is ignored. If fileid is a member of a PDSE supporting generations, then NEWGEN is used.

Member Generation Data Edit/Browse

If the library member contains data records that are mapped by a defined language structure (SELCOPYi SDO, COBOL, PL1, HLASM, SYMNAMES), then the Data Editor may be used to EDIT or BROWSE member generation data with or without the structure.

Like the Text Editor, the absolute and relative generation numbers are displayed in the Data Editor window view title bar.

Data Edit or Browse of a member generation may be started via the following methods:

1. Specify a member generation or member generation mask in the "Gen>" input field of the "Structured Data Browse/Edit" panel (=2). If a mask is specified, a list of matching generation entries are displayed in which one entry may be selected for use by the edit/browse operation.
2. Execute the structured data (SD) EDIT or BROWSE primary command with a member generation or member generation mask specification. e.g.

```
SD EDIT NBJ.DATA(EMP.-2)       USING NBJ.SDO(EMPMAP)
BROWSE  NBJ.JCLLIB(SSREP.-1)
```

3. Execute line command "SD E" or "B" against an entry in a Library Member Generations list to edit or browse the member generation respectively.

4. From a window view displaying the contents of a member generation (i.e. a Text Editor or Data Editor view), execute **GEN BROWSE**, **GEN SDE** or **GEN UPD** to VIEW, EDIT or EDIT for UPDATE only another generation of the same member respectively.

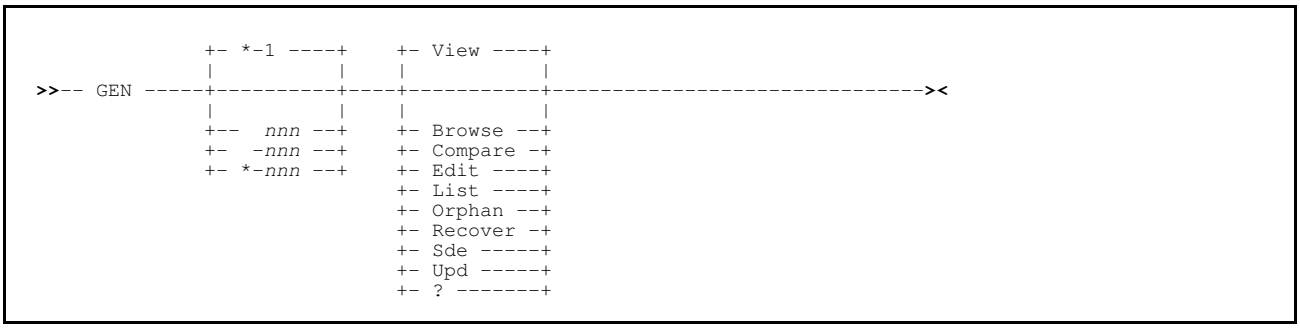
Saving changes to member generation data obeys the **GENSAVE** option as for the Text Editor. "**SAVE NOGEN**" and "**SAVE NEWGEN**" will override the value set for this option as appropriate.

Member Generation Primary Commands

The following primary commands have been introduced in SELCOPYi 3.50 and specifically relate to processing of PDSE version 2 member generations. These commands apply **only** to **Text Editor and Data Editor window views**.

GEN

Syntax:



Description:

GEN will perform utility operations based on the library DSN, member name or generation number of the member generation in the current Text Editor or Data Editor view.

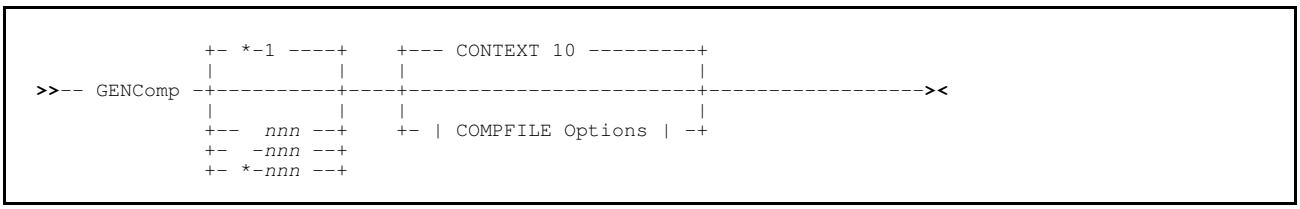
A relative generation number (*-nnn*), absolute generation number (*nnn*) or generation number relative to the generation in the current view (**-nnn*) may be specified. This identifies the target generation for a BROWSE, EDIT, SDE or UPD operation or the OLD file in a COMPARE files operation.

For example.

```
GEN -1      (View/browse relative generation -1)
GEN 22 E    (Edit        absolute generation 22)
GEN L       (List all generations of the current member)
GEN C       (Compare current generation with previous *-1 version)
```

GENCOMP

Syntax:

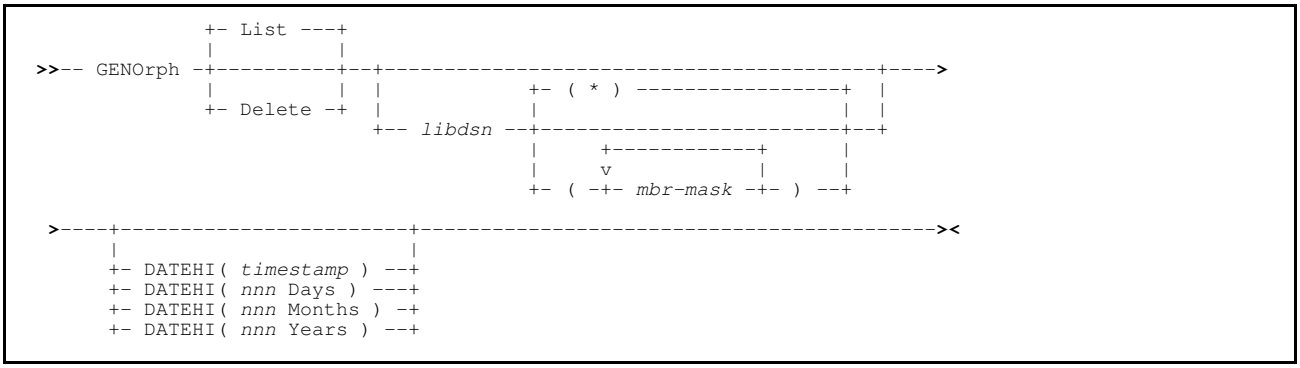


Description:

GENCOMP will execute the Compare Files Utility to compare the member generation in the current Text Editor or Data Editor view with a previous generation of the same member. Unlike **GEN COMPARE**, GENCOMP allows specification of options supported by the COMPFILE primary command.

GENORPH

Syntax:



Description:

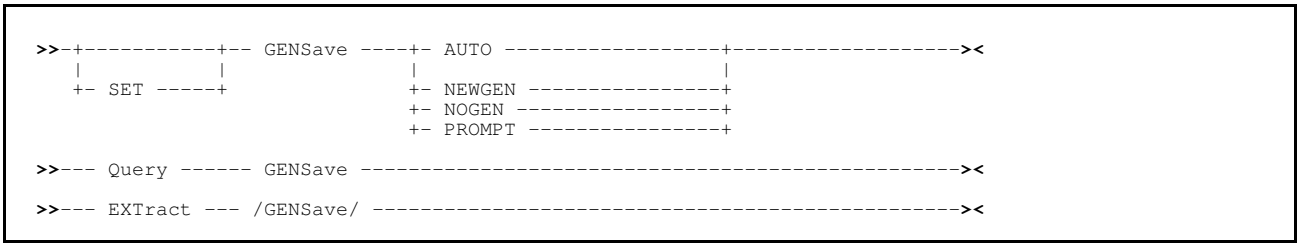
GENORPH will either list or delete orphaned member generations that belong to the specified library and match a member mask. If LIST/DELETE parameters are omitted, then LIST is the default. However, if no parameters are specified at all, then the "List/Delete PDSE v2 Orphaned Member Gens" panel will be displayed.

If a library name (*libdsn*) is specified with no member mask (*mbr-mask*), the default *mbrmask* is "(*)", i.e. all members. If no *libdsn* is specified, GENORPH uses the name of the library containing the member in the current Text Editor or Data Editor view and a *mbr-mask* of "(*)".

DATEHI will filter the generation entries based on their last modified recorded timestamp. Generations with a "LastMod" timestamp that is greater than the DATEHI specification will be omitted. The DATEHI timestamp may be expressed as a *nnn* number of days/months/years relative to the current date.

GENSAVE - SET/QUERY/EXTRACT

Syntax:



Description:

This option controls the default action taken when a SAVE or FILE command (or, for the Text Editor only, SSAVE or FFILE) is executed for a member generation in a PDSE version 2 library allocated with a MAXGENS value. The default action set by GENSAVE may be temporarily overridden by specifying the NEWGEN or NOGEN option on the SAVE, SSAVE, FILE and FFILE primary commands.

The following table identifies the action taken for each of the GENSAVE values.

GENSAVE	Action Taken
AUTO	If the edited member text is for the primary generation (generation 0) then NEWGEN is used. Otherwise, NOGEN is used. This is the SELCOPYi system default and matches ISPF Edit.
NEWGEN	The data is automatically saved to a new primary generation of the member. e.g. When editing generation -3 of a member, data will be saved as a new generation (generation 0) of the member and the original generation -3 data becomes generation -4. The absolute and relative generation values in the text edit window view title bar is updated to be that of the new generation 0.
NOGEN	The data is automatically saved to the same generation of the member. e.g. When editing generation -3 of a member, data will be saved back to generation -3. The generation numbers of all other generations of the same member are unchanged.
PROMPT	The CBLEDIT Save Member Generation popup window is displayed prompting the user to reply "Yes" to save a new generation, "No" to save to the same generation or "Cancel" to terminate the SAVE command and return to the edit view.

List/Delete PDSE V2 Orphaned Member Generations Panel

The "List/Delete PDSE V2 Orphaned Member Generations" panel has been introduced to manage member generations in PDSE V2 libraries that have been orphaned. An orphaned generation is one where the member base generation (generation 0) has been deleted or renamed so that no generation 0 exists for the member.

The panel may be displayed by selecting option 20 from the SELCOPYi Utilities menu (=8.20) or by entering the **GENORPH** primary command with no parameters.

This panel allows the user to identify and optionally to automatically delete orphaned member generations for a given library.

The **GENORPH** list or delete processing may be executed in the foreground or as a batch job.

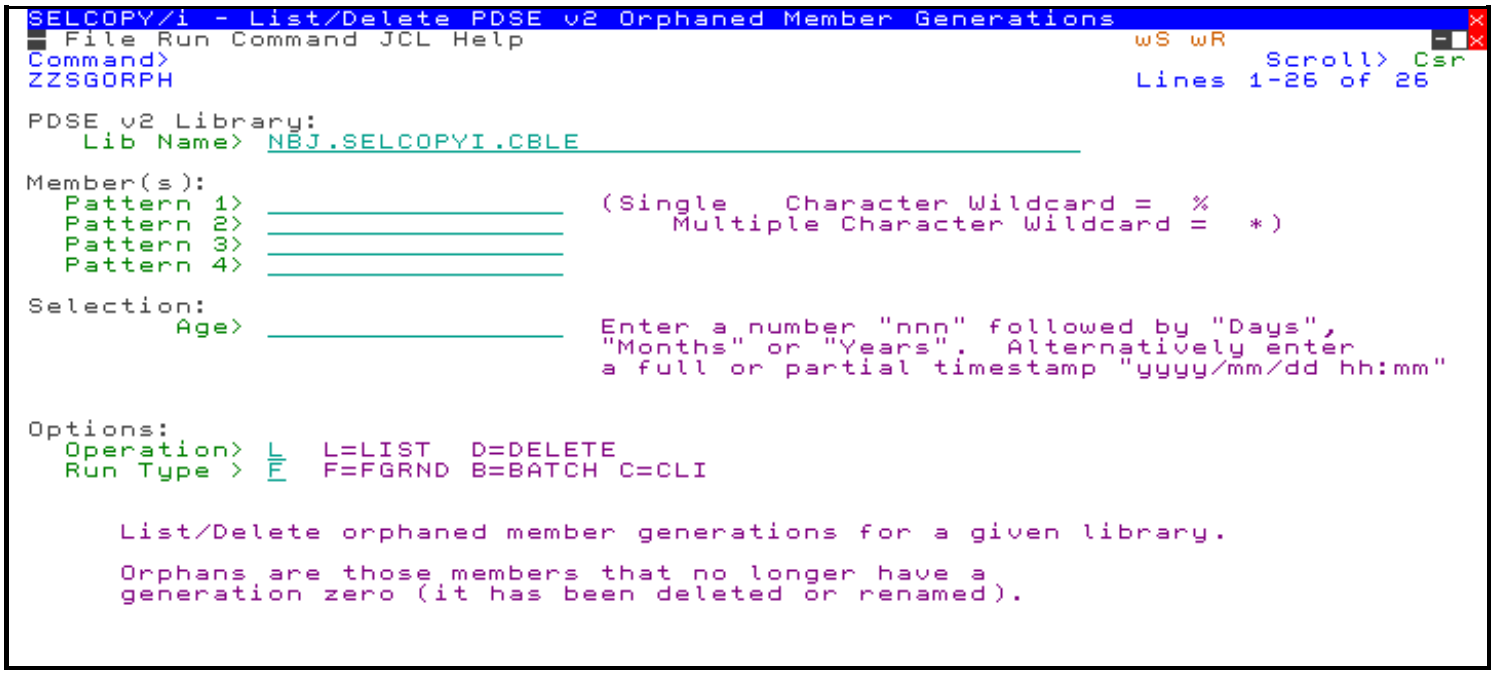


Figure 4. SELCOPYi - List/Delete PDSE V2 Orphaned Member Generations

SELCOPYi REPORT Utility

SELCOPYi 3.50 introduces a report utility that draws on the functionality of SELCOPYi structured data processing to produce attractive printed reports. As for all SELCOPYi utilities, the REPORT utility is provided as part of the SELCOPYi toolkit at no additional cost.

Full description of the REPORT utility including working samples, panels and report definition control statement syntax may be found in the CBL publication, "*SELCOPYi REPORT Utility*". REPORT Utility panels are on item 11 of the primary options panel.

The utility reports the values, or values derived from values obtained from specific data fields within formatted input. This input may originate from any one of the following sources:

1. Data sets, library members or HFS/ZFS files where records are mapped by a specified SELCOPYi SDO structure, HLASM DSECT, COBOL or PL/1 copybook.

2. SMF data sets. SMF records are mapped by the SELCOPYi SDO structures provided by SELCOPYi SMF Utilities.

3. A DB2 result table produced by a specific or SELCOPYi generated SQL query statement.

REPORT Printed Output

The input fields used and the format of the printed report are determined by basic, easy to write report definition control statements which get passed to the REPORT utility at execution.

For example, the following report definition control statements may be used to generate a report of Formula 1 Grand Prix circuits from data set records mapped by a COBOL Copybook structure:

```
HEAD:
  #TIMESTMP    /      "PAGE" #PAGE (RIGHT,5)
  "Formula 1 Grand Prix Circuits"

COLUMNS:
  COUNTRY
  TRACK
  LAPS
  LAP-LENGTH-KM
  RACE-LAP-RECORD
  RACE-LAP-RECORD-DATE
  RACE-LAP-RECORD-HOLDER
  RACE-LAP-RECORD-TEAM
```

Figure 5. SELCOPYi report definition.

The printed report generated by the utility might be:

```
12020/06/02 17:21                                                                                  PAGE      1
                                            Formula 1 Grand Prix Circuits

                                                                   RACE       RACE                 RACE
                                                 LAP     RACE      LAP        LAP                  LAP
                                                 LENGTH  LAP       RECORD     RECORD               RECORD
 COUNTRY    TRACK                          LAPS  KM      RECORD    DATE       HOLDER               TEAM
 ---------- ------------------------------ ----- ------- ---------------- ---------- -------------------- ----------
 Australia  Albert Park Circuit               58   5.303  00:01:24.124672 2004/03/07 Michael Schumacher   Ferrari
 Bahrain    Bahrain International Circuit     57   5.412  00:01:31.447296 2005/04/03 Pedro de la Rosa     McLaren
 China      Shanghai International Circuit    56   5.451  00:01:32.237824 2004/09/26 Michael Schumacher   Ferrari
 Azerbaijan Baku City Circuit                 51   6.003  00:01:43.009280 2019/04/28 Charles Leclerc      Ferrari
 Spain      Circuit de Catalunya              66   4.566  00:01:18.441472 2018/05/13 Daniel Ricciardo     Red Bull
 Monaco     Monte Carlo                       78   3.337  00:01:14.260480 2018/05/27 Max Verstappen       Red Bull
 Canada     Circuit Gilles Villeneuve         70   4.361  00:01:13.077760 2019/06/09 Valtteri Bottas      Mercedes
 France     Paul Ricard                       53   5.842  00:01:32.739584 2019/06/23 Sebastian Vettel     Ferrari
 Austria    Red Bull Ring                     71   4.318  00:01:06.957312 2018/07/01 Kimi Raikkonen       Ferrari
 UK         Silverstone                       52   5.891  00:01:27.368704 2019/07/14 Louis Hamilton       Mercedes
 Germany    Hockenheimring                    64   4.574  00:01:13.780224 2004/07/25 Kimi Raikkonen       McLaren
 Hungary    Hungaroring                       70   4.381  00:01:17.103104 2019/08/04 Max Verstappen       Red Bull
 Belgium    Spa-Francorchamps                 44   7.004  00:01:45.108480 2018/08/26 Valtteri Bottas      Mercedes
 Italy      Monza                             53   5.793  00:01:21.045504 2004/09/12 Rubens Barrichello   Ferrari
 Singapore  Marina Bay                        61   5.063  00:01:41.905408 2018/09/16 Kevin Magnussen      Haas
 Russia     Sochi Autodrom                    53   5.848  00:01:35.761408 2019/09/29 Louis Hamilton       Mercedes
 Japan      Suzuka                            52   5.807  00:01:30.983424 2019/10/13 Louis Hamilton       Mercedes
 Mexico     Autodromo Hermanos Rodriguez      71   4.304  00:01:18.741504 2018/10/28 Valtteri Bottas      Mercedes
 USA        Circuit of the Americas           56   5.513  00:01:36.168960 2019/11/03 Charles Leclerc      Ferrari
 Brazil     Interlagos                        71   4.309  00:01:10.540288 2018/11/11 Valtteri Bottas      Mercedes
 UAE        Abu Dhabi Yas Marina              55   5.554  00:01:39.282944 2019/12/01 Louis Hamilton       Mercedes
                                           ===== =======
 == Grand Totals (21 Items)                 1262 108.634
                                           ===== =======
```

Figure 6. SELCOPYi printed report output.

REPORT CSV, XML and JSON Output

In addition to generating printable reports, the REPORT utility may use the same report definition input to generate output in the following formats:

- Comma Separated Variable (CSV)
- Extensible Markup Language (XML)
- Java Script Object Notation (JSON)

For example, the sample report definition above may also be used to generate the following CSV output:

```
"COUNTRY","TRACK","LAPS","LAP-LENGTH-KM","RACE-LAP-RECORD","RACE-LAP-RECORD-DATE","RACE-LAP-RECORD-HOLDER","RACE-LAP-RECORD-TEAM"
"Australia ","Albert Park Circuit           ","   58","  5.303"," 00:01:24.124672","2004/03/07","Michael Schumacher  ","Ferrari "
"Bahrain   ","Bahrain International Circuit ","   57","  5.412"," 00:01:31.447296","2005/04/03","Pedro de la Rosa    ","McLaren "
"China     ","Shanghai International Circuit","   56","  5.451"," 00:01:32.237824","2004/09/26","Michael Schumacher  ","Ferrari "
"Azerbaijan","Baku City Circuit             ","   51","  6.003"," 00:01:43.009280","2019/04/28","Charles Leclerc     ","Ferrari "
"Spain     ","Circuit de Catalunya          ","   66","  4.566"," 00:01:18.441472","2018/05/13","Daniel Ricciardo    ","Red Bull"
"Monaco    ","Monte Carlo                   ","   78","  3.337"," 00:01:14.260480","2018/05/27","Max Verstappen      ","Red Bull"
"Canada    ","Circuit Gilles Villeneuve     ","   70","  4.361"," 00:01:13.077760","2019/06/09","Valtteri Bottas     ","Mercedes"
"France    ","Paul Ricard                   ","   53","  5.842"," 00:01:32.739584","2019/06/23","Sebastian Vettel    ","Ferrari "
"Austria   ","Red Bull Ring                 ","   71","  4.318"," 00:01:06.957312","2018/07/01","Kimi Raikkonen      ","Ferrari "
"UK        ","Silverstone                   ","   52","  5.891"," 00:01:27.368704","2019/07/14","Louis Hamilton      ","Mercedes"
"Germany   ","Hockenheimring                ","   64","  4.574"," 00:01:13.780224","2004/07/25","Kimi Raikkonen      ","McLaren "
"Hungary   ","Hungaroring                   ","   70","  4.381"," 00:01:17.103104","2019/08/04","Max Verstappen      ","Red Bull"
"Belgium   ","Spa-Francorchamps             ","   44","  7.004"," 00:01:45.108480","2018/08/26","Valtteri Bottas     ","Mercedes"
"Italy     ","Monza                         ","   53","  5.793"," 00:01:21.045504","2004/09/12","Rubens Barrichello  ","Ferrari "
"Singapore ","Marina Bay                    ","   61","  5.063"," 00:01:41.905408","2018/09/16","Kevin Magnussen     ","Haas    "
"Russia    ","Sochi Autodrom                ","   53","  5.848"," 00:01:35.761408","2019/09/29","Louis Hamilton      ","Mercedes"
"Japan     ","Suzuka                        ","   52","  5.807"," 00:01:30.983424","2019/10/13","Louis Hamilton      ","Mercedes"
"Mexico    ","Autodromo Hermanos Rodriguez  ","   71","  4.304"," 00:01:18.741504","2018/10/28","Valtteri Bottas     ","Mercedes"
"USA       ","Circuit of the Americas       ","   56","  5.513"," 00:01:36.168960","2019/11/03","Charles Leclerc     ","Ferrari "
"Brazil    ","Interlagos                    ","   71","  4.309"," 00:01:10.540288","2018/11/11","Valtteri Bottas     ","Mercedes"
"UAE       ","Abu Dhabi Yas Marina          ","   55","  5.554"," 00:01:39.282944","2019/12/01","Louis Hamilton      ","Mercedes"
```

Figure 7. SELCOPYi CSV report output.

SELCOPYi SMF Utilities

SELCOPYi 3.50 introduces utilities that process records written by the IBM System Management Facility (SMF). As for all SELCOPYi utilities, these SMF utilities are provided as part of the SELCOPYi toolkit at no additional cost.

SMF records are mapped by a library of SELCOPYi structured data objects (SDOs) provided as part of the SELCOPY Product Suite package (see target install library "*prefix*.SZZSDIST.SDO"). The source CREATE STRUCTURE primary command for each SDO library member is found in a member of the same name in target install library "*prefix*.SZZSDIST.SMFMAP". SELCOPYi SMF utilities use these SDO structures to format the SMF record segments.

Full descriptions of the SMF utilities may be found in the CBL publication, "*SELCOPYi SMF Utilities*".

A brief description of each of the supplied SMF Utilites are as follows:

SMF Browse

Opens a Data Editor window view to browse a formatted display of SMF records. Selection criteria options are supported allowing the user to filter the records on input so that only records that satisfy the criteria will be displayed.

The SMF Browse Utility panel is accessible via item 1 of the "**SMF Features Menu**" (=13.1).

Two levels of SMF Browse record formatting (**Basic** and **Full**) is available.

SMF Basic Browse - SMFBB

The SMF Browse for basic record formatting utility (SMFBB) will use the same record mapping to display all records regardless of SMF record type. The mapping contains only selected fields that exist in the SMF record header. The remainder of the record is presented in the display as plain text in a field of character data type.

Basic formatting is useful when a general overview of the SMF records is required.

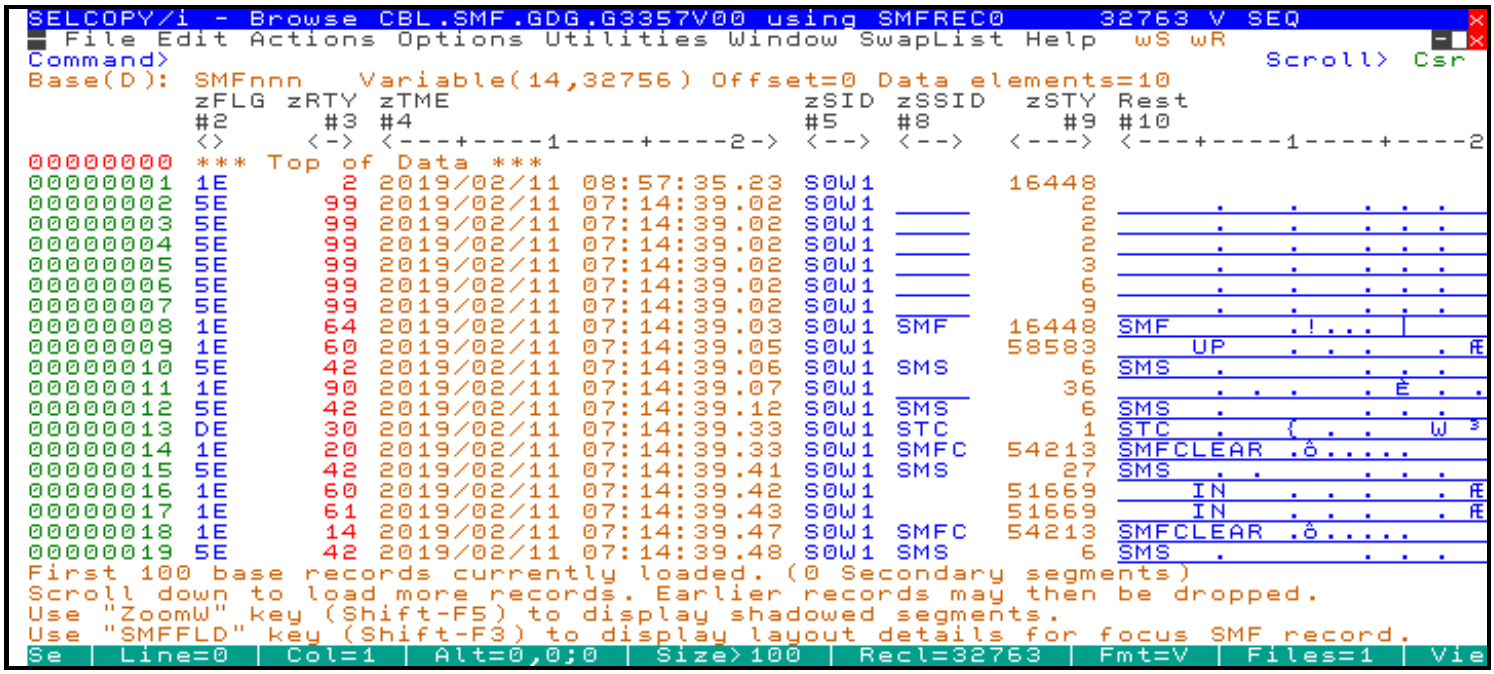


Figure 8. Browse Basic Layout Display.

SMF Full Browse - SMFB

Full SMF record formatting displays **all** fields in an SMF record.

The SMF Browse for full record formatting utility (SMFB) will use multiple SMF record type segment mappings (record-types) to format and display all fields belonging to the SMF records.

Record-types representing secondary segments of the SMF record may optionally be shadowed when the browse view is opened, thus displaying only fields from the primary segment containing the header information (including the SMF record type, sub-type and timestamp). Shadowed segments may be brought back into view using the SET VBASE OFF primary command or "V"/"V+" line commands.

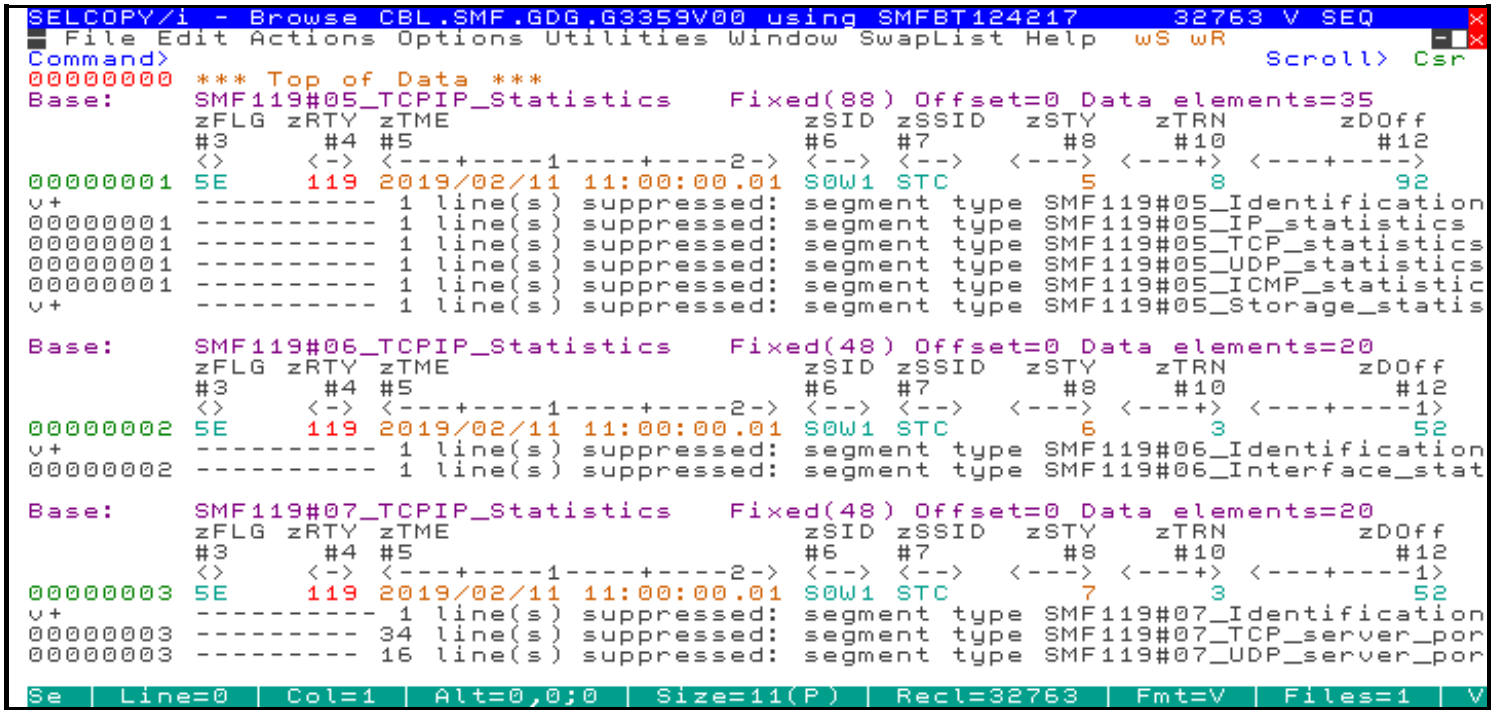


*Figure 9.*Browse Full Layout Display.

SMF Record Extract

The SMF Extract utility (SMFEXTRC) is used to create a subset of SMF records. The utility will input records from an SMF log or archive data set, extract only records that match specified criteria and output them to another data set.

Record selection criteria may comprise tests on SMF record type/sub-type, high and low timestamp, User Id, Job Name and generic search string.

The SMF Extract Utility panel is accessible via item 2 of the "**SMF Features Menu**" (=13.2).

SMF Report

The SELCOPYi REPORT Utility (REPORT) may be used to produce printed reports or CSV, XML or JSON format output from SMF input records and includes processing that is specific to SMF record input.

Typically, the structure (e.g. COBOL or PL1 Copybook) required to map input data set records must be specified to the REPORT utility on every execution. However, when an SMF report is requested, the REPORT utility will automatically determine the SMF record mapping SELCOPYi SDO structures it needs to process the input records based on control statements in the report definition. It then uses record-type definitions within these SDO structures to format and report on fields within the SMF record segments.

Therefore, the report definition need only specify the record-type and name of the SMF record field to be reported, and the REPORT utility will automatically use the correct SELCOPYi SMF SDO structure name.

The REPORT Utility panel for SMF input is accessible via item 4 of the "**Print/Report Features Menu**" (=11.4) or item 3 of the "**SMF Features Menu**" (=13.3).

See also CBL publication "*SELCOPYi REPORT Utility*".

In the following example, the report definition control statements may be used to generate a report of SMF record type 80 (RACF or Other Security Product Processing) records. SMF records of other types/sub-types are bypassed.

```
HEAD:
  #TIMESTMP / "RACF Event Log" / "PAGE" #PAGE (RIGHT,4)

REQUIRED:
  SMF080_*.zRESOURCE
  SMF080_*.zDSN

COMPUTE:
  select
    when  zDSN      <> '' then Resource=zDSN
    when  zResource <> '' then Resource=zResource
    otherwise                  Resource=''
  end

COLUMNS:
  SMF080_Security_Product_Processing.zTME        19 "Time Stamp"
  :Resource                                      25 "Resource Name"
  SMF080_Security_Product_Processing.zUSERID        "User|Name"
  SMF080_Security_Product_Processing.zGRP           "Group|Name"
  SMF080_Auth_Access_Requested.zAUTHREQ           9 "Access|Requested"
  SMF080_Auth_Access_Allowed.zAUTHALLOW           9 "Access|Allowed"
  SMF080_Security_Product_Processing.zEVENT_NAME 15 "Event"
  SMF080_Security_Product_Processing.zEVENT_QUAL 22 "Comment"
```

Figure 10. SELCOPYi SMF report definition.

The printed report generated by the utility might be:

```
12020/06/04 11:24                                      RACF Event Log                                                    PAGE    1

                                                 User     Group    Access    Access
 Time Stamp          Resource Name               Name     Name     Requested Allowed   Event           Comment
 ------------------- ------------------------- -------- -------- --------- --------- --------------- ----------------------
 2019/03/15 11:50:11                           NBJ      CBL                          ADDUSER         No violations detected
 2019/03/19 14:04:01                           NBJ      CBL                          DELUSER         No violations detected
 2019/03/19 14:09:55 NBJ.TEST01.RACF           NBJ      CBL                          ALTDSD          No violations detected
 2019/03/19 14:11:22 NBJ.TEST01.RACF           NBJ      CBL                          PERMIT          No violations detected
 2019/03/19 14:11:38                           NBJ      CBL                          SETROPTS        No violations detected
 2019/03/19 14:14:04 NBJ.TEST01.RACF           NBJ3     CBL      READ      NONE      RESOURCE ACCESS Insufficient authority
 2019/03/19 14:15:42 NBJ.CBLI.MBRLIST.SEQ.BIG  NBJ3     CBL      UPDATE    READ      RESOURCE ACCESS Insufficient authority
 2019/03/19 14:20:28                           NBJX                                  JOB INITIATION  Undefined user id
 2019/03/19 14:40:44 NBJ.TEST01.RACF           NBJ      CBL                          ALTDSD          No violations detected
 2019/03/19 14:40:48                           NBJ      CBL                          SETROPTS        No violations detected
 2019/03/19 14:41:58                           NBJ2     CBL                          JOB INITIATION  Invalid password
 2019/03/19 16:02:25 NBJ.TEST01.RACF           JGE      CBL      READ      ALTER     RESOURCE ACCESS Successful access
 2019/03/19 16:02:25 NBJ.TEST01.RACF           JGE      CBL      READ      ALTER     RESOURCE ACCESS Successful access
 2019/03/19 16:02:33 NBJ.TEST01.RACF           JGE      CBL      UPDATE    ALTER     RESOURCE ACCESS Successful access
 2019/03/19 16:02:33 NBJ.TEST01.RACF           JGE      CBL      UPDATE    ALTER     RESOURCE ACCESS Successful access
 2019/03/19 16:05:09 NBJ.TEST01.RACF           NBJ2     CBL      READ      READ      RESOURCE ACCESS Successful access
 2019/03/19 16:05:09 NBJ.TEST01.RACF           NBJ2     CBL      READ      READ      RESOURCE ACCESS Successful access
 2019/03/19 16:05:40 NBJ.TEST01.RACF           NBJ2     CBL      UPDATE    READ      RESOURCE ACCESS Insufficient authority
 2019/03/19 16:09:24 NBJ.TEST01.RACF           NBJ2     CBL      UPDATE    READ      RESOURCE ACCESS Insufficient authority
 2019/03/19 16:10:57 NBJ.TEST01.RACF           NBJ      CBL                          PERMIT          No violations detected
 2019/03/19 16:12:14 NBJ.TEST01.RACF           NBJ2     CBL      UPDATE    ALTER     RESOURCE ACCESS Successful access
 2019/03/19 16:12:14 NBJ.TEST01.RACF           NBJ2     CBL      UPDATE    ALTER     RESOURCE ACCESS Successful access
 2019/03/19 16:21:50 NBJ.TEST01.RACF           NBJ2     CBL      READ      ALTER     RESOURCE ACCESS Successful access
 2019/03/19 16:21:50 NBJ.TEST01.RACF           NBJ2     CBL      READ      ALTER     RESOURCE ACCESS Successful access
 2019/03/19 16:21:58 NBJ.TEST01.RACF           NBJ2     CBL      UPDATE    ALTER     RESOURCE ACCESS Successful access
 2019/03/19 16:21:58 NBJ.TEST01.RACF           NBJ2     CBL      UPDATE    ALTER     RESOURCE ACCESS Successful access

 == Grand Totals (26 Items)
```

Figure 11. SELCOPYi SMF printed report output.

SMF Record Layout

The layout of each SMF record type and sub-type, which includes descriptions of fields plus the record-type (SMF record segment) names, field data types and field names defined in the SELCOPYi SDO structures, are documented in Appendix B. of the "*SELCOPYi SMF Utilities*" publication.

Alternatively, the SMF record layout utility (SMFFLD) may be executed to display the layout information for a particular SMF record type in a SELCOPYi Help window view.

The following example shows the layout of SMF record 119 (TCP/IP Statistics), sub-type 1 (TCP/IP Connection) as displayed by the SMFFLD utility.

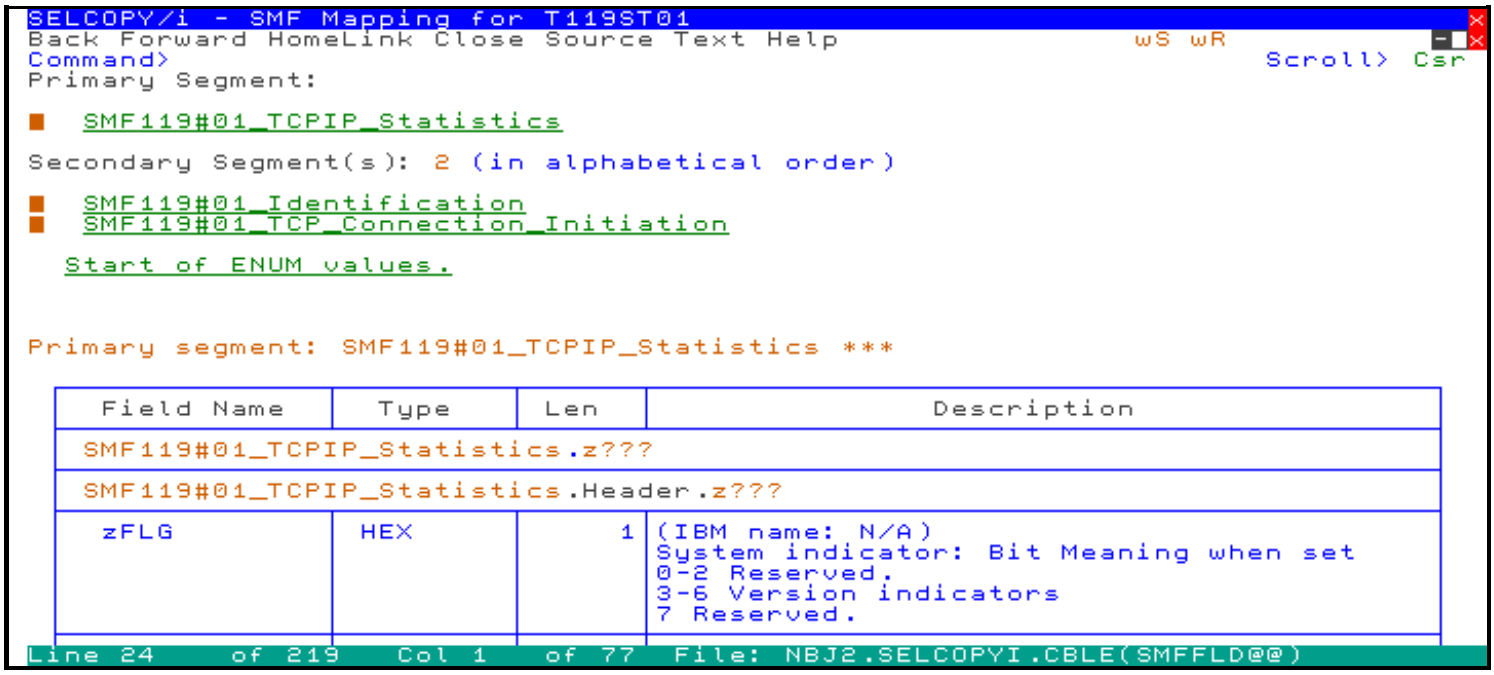


Figure 12. SELCOPYi SMF Record Type 119 Subtype 1 Layout.

SELCOPYi Text Editor

The SELCOPYi Text Editor provides data set, library member and HFS/ZFS file text edit capabilities that matches the ISPF PDF Editor but also provides additional functionality.

Processing Enhancements

Only very minor enhancements have been applied to the SELCOPYi Text Editor in SELCOPYi 3.50. These are as follows:

BOXTOT - Total for Numerics in a Marked Box

SELCOPYi includes a library of Text Editor and Data Editor macros created as part of the product package install. This library is usually the last specified in the SELCOPYi macro path and has a DSN of "*prefix*.SZZSDIST.CBLE" where *prefix* is your installation DSN high level qualifier for the SELCOPY Product Suite. Use `QUERY MACROPATH` to display the editor macro search path.

The BOXTOT macro provides a utility for returning the sum of a list of decimal or hexadecimal values that has been marked and highlighted in the text display using the MARK command (the default for F17 and F18). This macro has been updated to support totalling decimal values that include a fraction (i.e. values containing a decimal point).

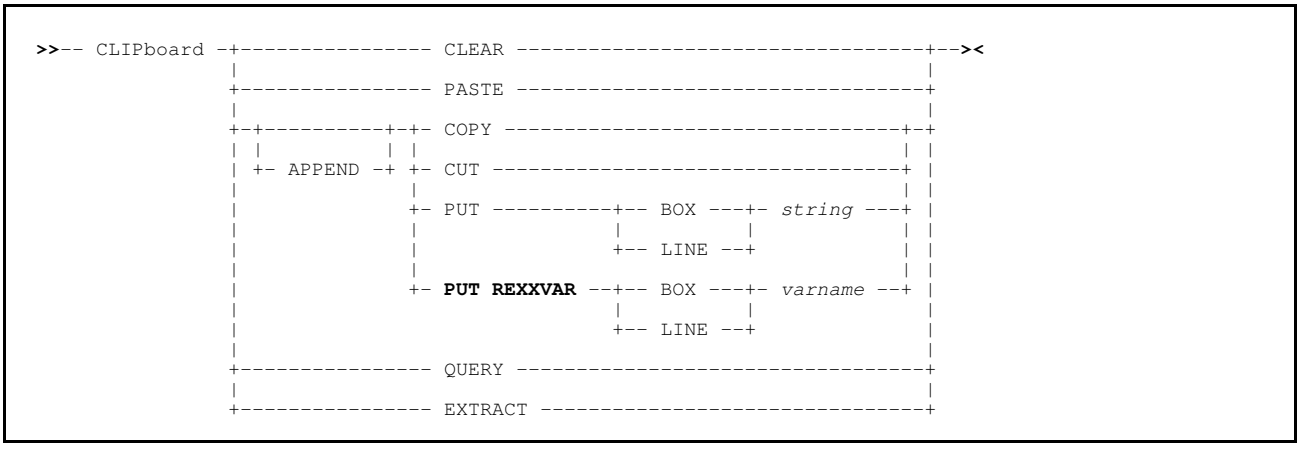
For example, if all values in the following list exist within the text of the display area and are marked, then BOXTOT would return "**BoxTot: DECIMAL Total=4.9**" in the message line.

```
  2.1
   1
 1.8
```

SELCOPYi Clipboard

The SELCOPYi clipboard allows a user to copy, cut and paste text between Text Editor and Data Editor views displaying text from different sources.

For use specifically in Text Editor and Data Editor REXX macros, SELCOPYi 3.50 includes an update to the CLIPBOARD primary command so that PUT now also supports adding text to the clipboard that is assigned to a named REXX variable.



REXX Macros

The distributed product suite includes a library of REXX macros used by SELCOPYi utilities. This library usually has a DSN in the format "*prefix*.SZZSDIST.CBLE". Where possible, to improve performance **compiled versions** of these macros are included.

For non-compiled SELCOPYi REXX macros, you may now also include "**RXCCSID=***ccsid*" in position 3 of any comment line within the macro to identify the CCSID of the system on which the macro was written. This will trigger automatic CCSID conversion on the REXX macro text to convert it from the specified CCSID (*ccsid*) to that of the local system.

This is necessary if a macro may be executed on systems with different local CCSIDs to that of the system on which the macro was written. For all CBL supplied macros that reference certain special characters, RXCCSID="285" is specified.

SELCOPYi Structures

SELCOPYi utilities perform functions on data which comprises multiple individual fields of potentially different data type formats. These data records (or DB2 table rows) are usually mapped by structures written for different programming or utility languages (e.g. COBOL, PL1, Assembler and DFSORT).

To process structured data, SELCOPYi uses its own structure definition objects (SDO). An SDO may be created using the Structured Data **CREATE STRUCTURE** utility command. Alternatively, a temporary SDO may be created automatically by SELCOPYi when a COBOL, PL1, HLASM or DFSORT programming language structure is passed to a SELCOPYi utility as the data mapping source.

The CREATE STRUCTURE utility can create an SDO from one or more existing COBOL, PL1, HLASM or DFSORT programming language record mapping structures. It also supports creating an SDO without a pre-existing mapping source. The command syntax allows the user to define one or more record structure layouts each containing any number of field definitions. The utility's record and field mapping syntax actually provides more options and is more flexible than that provided by programming language record mapping structures.

SELCOPYi 3.50 includes a number of enhancements to SELCOPYi SDO structures.

Field Data Types

Using the CREATE STRUCTURE direct field definition syntax, SDO structures may be created with field mappings of different data types. The data type of a field determines how the field data is interpreted and displayed.

SELCOPYi 3.50 introduces support for a number of new data types and includes updates to some existing data types. These generally apply to fields belonging to SMF generated records and are used in the SELCOPYi SDO structures distributed as part of the product package to map SMF records.

New and updated data types specifications are as follows:

Data Type	Description
IPADDRESS(4) or **IP(4)**	Defines a field to be interpreted as an IP address occupying **4** bytes. IP(4) is assumed to be an IPv4 address. The value will display as 4, 3-digit decimal values each separated by a "." (dot/period) with an overall length of 15 bytes. e.g. *192.168.001.064*
IPADDRESS(16) or **IP(16)**	Defines a field to be interpreted as an IP address occupying **16** bytes. This format will detect whether the 16-byte source represents an IPv4 or IPv6 address. If the first 10 bytes of the source are X'00' and the next 2 bytes are X'FF', then the field is determined to be an IPv4 address. The junior 4 bytes of the source value will be processed as for IPADDRESS(4) and the displayed value will be left justified within a 39-byte display area. Otherwise, the source is determined to be an IPv6 address. The value will display as 8, 4-digit hexadecimal values each separated by a ":" (colon) with overall length of 39 bytes. e.g. 0123:4567:89AB:CDEF:0123:4567:89AB:CDEF
TIME(DECIMAL2)	Packed decimal format of length 4-bytes (X'00HH,MMSS') with decimal values for hours (HH), minutes (MM) and seconds (SS). Note that the data is unsigned.
TIME(DECIMAL3)	Packed decimal format of length 4-bytes (X'0HHM,MSSC') with decimal values for hours (HH), minutes (MM) and seconds (SS). Note that the data **is** signed with X'C'.
TIME(STCK)	STCK format of length 8 bytes. A 64-bit unsigned binary value elapsed time value in the same format as the system TOD clock.
TIME(STCK,*nbits***)**	The low order, unsigned 32-bits (4-bytes) of a STCK format value which has been shifted right a number of bits specified by *nbits*. (This STCK format is often used for elapsed time values in SMF records.) The following shows the number of microsecond (us) time units represented by a single bit in the 32-bit value identified by TIME(STCK,*nbits*). TIME(STCK,12) 1us TIME(STCK,16) 16us TIME(STCK,19) 128us TIME(STCK,22) 1024us TIME(STCK,32) 1048576us
TIME(UNIX)	UNIX format time of length 4 bytes. A 32-bit unsigned binary value which is the number of seconds elapsed since midnight.

TIMESTAMP(BINARY)	Binary date and time format of length 10-bytes (X'yyyy,00mm,00dd,nnnn,nnnn') with binary values for year (yyyy), month of year (mm) and day of month (dd) followed by a 32-bit unsigned binary value equal to the number of hundredths of seconds (0.01 second) since midnight.
TIMESTAMP(DECIMAL) or **TIMESTAMP(TIMEDEC)**	Packed decimal date and time format of length 8-bytes (X'0cyy,dddF,HHMM,SSTH') with decimal values for year of century (yy) and Julian day of year (ddd). 'F' is a 4-bit sign and 'c' is the century indicator X'0'(1900) or X'1'(2000). The packed decimal time format that follows the date is unsigned and has decimal values for hours (HH), minutes (MM), seconds (SS) and hundredths of a second (TH).
TIMESTAMP(DECIMAL2) or **TIMESTAMP(TIMEDEC2)**	Packed decimal date and time format of length 8-bytes (X'0cyy,dddF,00HH,MMSS') with date format as described for TIMESTAMP(DECIMAL) and packed decimal time format as described by TIME(DECIMAL2)
TIMESTAMP(DECIMAL3) or **TIMESTAMP(TIMEDEC3)**	Packed decimal date and time format of length 8-bytes (X'0cyy,dddF,0HHM,MSSC') with date format as described for TIMESTAMP(DECIMAL) and packed decimal time format as described by TIME(DECIMAL3)
TIMESTAMP(TIMEBIN)	Packed decimal date and binary time format of length 8-bytes (X'0cyy,dddF,nnnn,nnnn') with date format as described for TIMESTAMP(DECIMAL) and time format as described by TIMESTAMP(BINARY)
TIMESTAMP (STCK,*nbytes***)**	The high order bytes of a store clock value of length *n-bytes*. The source value is padded with nulls to length 8-bytes so that it is a 64-bit unsigned binary value in the same format as the system TOD clock. STCK,4 is used as the timestamp format in a number of SMF records fields.
TIMESTAMP(SMF)	SMF record timestamp format (X'nnnn,nnnn,yyyy,dddF') which is the same as TIMEBIN except that the time value precedes the date. TIMESTAMP(SMF) is used as the timestamp format in SMF record headers.

Compressed Record Data

SELCOPYi 3.50 introduces support for mapping records which contain CSRCE compressed data. This feature has been introduced to support SMF 110 subtype 1 (CICS Monitoring) records which may contain this type of compressed data.

Using the CREATE STRUCTURE direct field definition syntax, SDO structures may be created so that a compressed area of data within a record may be expanded when mapped by a particular record mapping definition (record-type).

The **EXPAND** sub-clause may be specified as part of a record mapping (record-type) definition. This sub-clause identifies the location within the record of the compressed data (**EXPLOC**), the length of the compressed data (**EXPSIZE**) and the condition that identifies the data as being compressed (**WHEN** *expression*).

The following example executes the CREATE STRUCTURE utility to create an SDO containing a single record-type mapping definition (**UserInfo**) which comprises a header structure (**Header**) followed by an area of data mapped by a data structure (**CompData**). The command is in a format suitable for saving in a command centre file and execution using the ACTION key (Shift-F4).

If the compressed data indicator in field **HFlag** is set, then the area of data starting at the record position specified by field **HCOff** for a length specified by field **HCLen** will be expanded. Typically, the header structure occupies a fixed, uncompressed area at the start of the record and the remainder of the record data may be compressed.

```
<CREATE STRUCTURE  CBL.SELCOPYI.SITE.SDO(COMP001)  \
       (UserInfo  struct                           \
          ( Header    STRUCT                       \
              ( HType     INT(1)                   \
               ,HFlag     INT(1)                   \
               ,HTitle   CHAR(20)                  \
               ,HCOff     INT(4)                   \
               ,HCLen     INT(4)                   \
              )                                    \
           ,CompData  STRUCT                       \
              ( UserId   CHAR(8)                   \
               ,DescLen   INT(4)                   \
               ,Desc    XVARCHAR(16384,DescLen)    \
              )                                    \
          )                                        \
        DEFAULT                                    \
                                                   \
        EXPAND                                     \
          ( EXPLOC   (HCOff)                       \
            EXPSIZE  (HCLen)                       \
            WHEN     (HFlag = X'01')               \
          )                                        \
       )
```

LEVEL 1 Name for HLASM Source Members

Where an SDO is created from a single COBOL or PL1 copy book using the CREATE STRUCTURE utility's basic copybook specification, a level 1 entry may be specified to name the record mapping structure/group-field if no level 1 entry exists in the copy book member.

SELCOPYi 3.50 has been updated so that this same functionality will also apply to HLASM source members that contain DS and/or DC instructions but no DSECT instruction.

In the following example, the HLASM source member NODSECT contains DS instructions but no DSECT instruction. The CREATE STRUCTURE command prefixes the contents of the source member with "MapData DSECT" so that a record mapping of record-type name MapData is created.

```
<CREATE STRUCTURE  CBL.SELCOPYI.SITE.SDO(TEST01)    \
             FROM HLASM   LEVEL 1 MapData   CBL.ASM.SOURCE(NODSECT)
```

SELCOPYi Data Editor

The SELCOPYi Data Editor provides facilities to perform browse and edit of data mapped by a structure, VSAM data sets, DB2 table rows and also data sets that are too large to be loaded entirely into available storage.

SELCOPYi 3.50 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.

Extended Compressed Data Sets

A restriction that prevented processing of an SMS Extended-format, compressed data sets (EXTENDED, COMPACTION=YES) has been addressed in SELCOPYi 3.50.

Browse and Full Edit (but not In-place Edit) may be performed on Extended-format, compressed data set records. For large compressed data sets, Auxiliary Edit may occur using a temporary, non-compressed auxiliary data set.

Extended-format, compressed data sets may also be used as input and output for all other SELCOPYi utilities that the structured data processing environment. e.g. FCOPY, FSU, COMPFILE, etc.

DB2 VIEW

Edit or Browse of a DB2 VIEW definition will display the result table rows returned by the SQL Query defined to the DB2 VIEW definition.

For Edit or Browse of a DB2 TABLE (base table definition), values belonging to DB2 table columns defined as being part of the table primary key are highlighted and flagged in the Data Editor display. If the rows being displayed belong to a result table generated by an SQL Query, primary key columns are not applicable and so no highlighting occurs.

SELCOPYi 3.50 includes additional processing for an SQL Query result table created via a DB2 VIEW definition so that, provided the following conditions are met, primary key columns will be flagged and their values highlighted.

1. All primary key columns belonging to all DB2 tables specified in the SQL Query are selected.
2. Primary key column names are not referenced as different names.

BROWSE/EDIT Operation

The following new features have been introduced which relate to starting a new Data Editor BROWSE or EDIT view.

BROWSE Tape Data Sets

For BROWSE only, a Data Editor view may be opened to display the contents of a cataloged tape data set.

For an uncataloged tape dataset, the ALLOC command should be used to first allocate the data set to a DD name and, in doing so, provide the tape volume details (e.g. UNIT, POSITION and VOLUME). The allocated DD name may then be used as the BROWSE data source. (See BROWSE DD=*ddname* below.)

As for browse of DASD data set records, the display of the tape data set records may be scrolled forwards and backwards. Records will be loaded and unloaded from storage as necessary as the display is scrolled.

BROWSE DD=*ddname*

For BROWSE only, the source of the data to be displayed in the Data Editor view may now also be specified as an allocated DD name (*ddname*).

To browse a DD name, the input source *fileid* specified on the BROWSE primary command or in the Name> field of the "Structured Data Browse/Edit" panel, should be replaced with **DD=***ddname*. The *ddname* may be allocated to any of the following:

1. A DASD data set or library member.
2. A TAPE data set.
3. A system temporary data set.
4. An HFS/ZFS file id.
5. A concatenation of data sets.

For example, the following will browse the data belonging to a data set allocated to DD MYDATA:

```
BROWSE  DD=MYDATA
```

Structured Data Browse/Edit Panel

The previous version of SELCOPYi introduced operand **KEYRANGE** for BROWSE and EDIT primary commands. This provided a method whereby only VSAM KSDS records whose key values fall within a range of key values are selected for edit or browse.

SELCOPYi 3.50 introduces the **End>** input field in the "Structured Data Browse/Edit" panel. This field is applicable only to input VSAM KSDS data sets where record selection by "Key" has been selected.

The **Start>** input field already exists and may be used to specify the lowest possible key value of the first KSDS record to be selected. If a value also exists in the **End>** field, then this will be the highest possible KSDS record key, thus identifying a range of records keys.

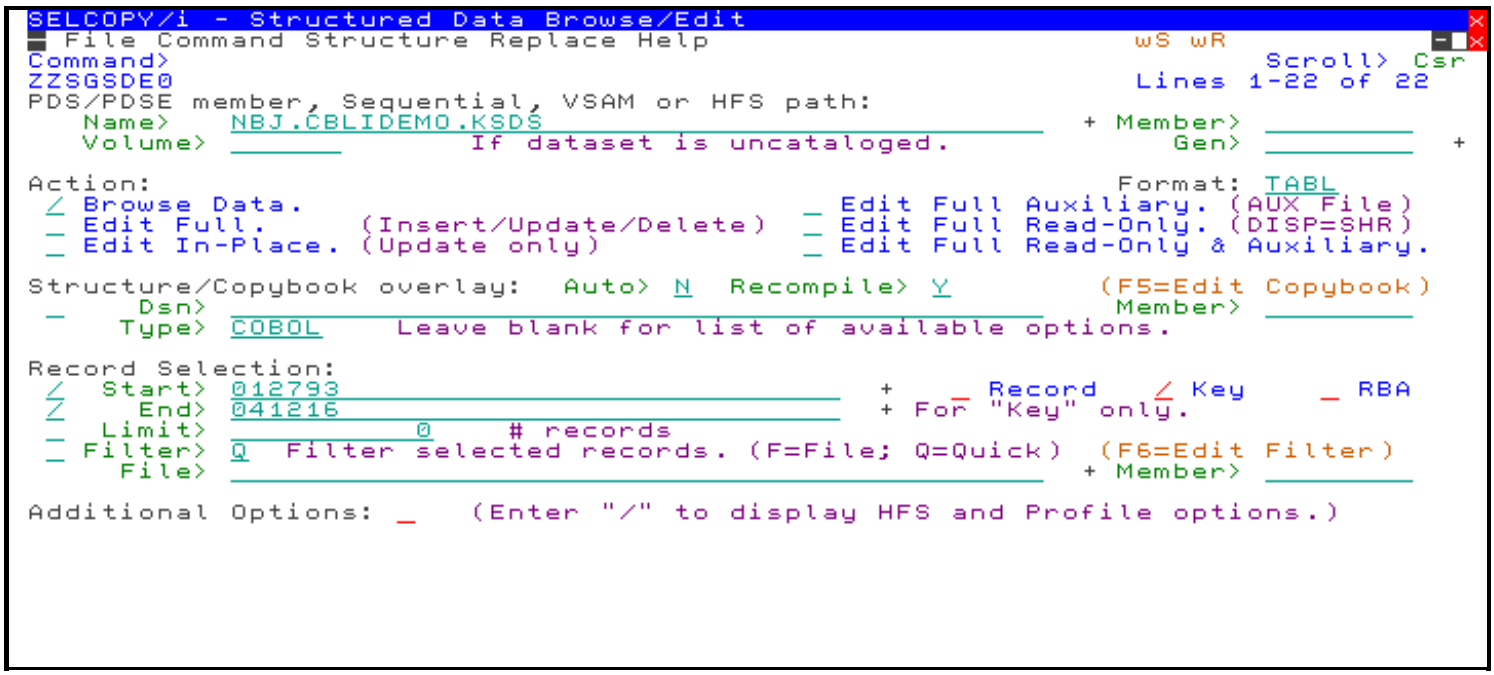


Figure 13. SELCOPYi Structured Data Browse/Edit Panel - KSDS Key Range.

PL1 Copybook without DECLARE

BROWSE or EDIT of a formatted data mapped by a PL1 copybook no longer returns an error if the copy book does not contain a DECLARE statement. For example:

```
EDIT   'NBJ.ZZST2DAT'   USING  PL1  NBJ.COPYBOOK.PL1(XXTEST)
```

SELCOPYi 3.50 has been updated so that, like COBOL copybooks where no LEVEL 1 entry exists, a level 1 entry will be generated automatically for PL1 copy books where no DECLARE statement exists. Specifically, the following will be added before the PL1 copy book records if a numeric integer value other than 1 is detected at the start of the first PL1 statement.

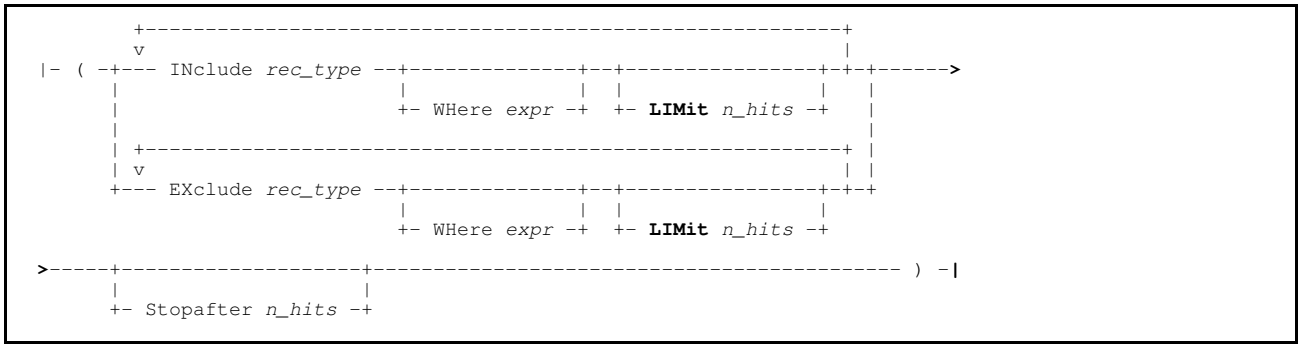
```
DECLARE 01 SELCOPY01 ,
```

This will generate a temporary SDO structure with a record-type mapping name of "SELCOPY01".

FILTER Limits

A filter may be specified on BROWSE and EDIT operations and (also on FCOPY, FSU, PRINT, REPORT, CSVGEN, JSONGEN and XMLGEN utilities) to include or exclude records that match specified selection criteria.

Filter clause syntax has been enhanced in SELCOPYi 3.50 to allow specification of a limit for the number of records selected by an individual INCLUDE sub-clause (or deselected by an individual EXCLUDE sub-clause). The Filter clause syntax is now as follows:



The LIMIT operand may be specified as part of each INCLUDE or EXCLUDE sub-clause. Each time a sub-clause returns a true condition, the hit count for the sub-clause is incremented by 1. Once the hit count reaches the LIMIT threshold, the sub-clause plays no further part in the filter of subsequent input records.

In this following example, the maximum number of records edited is 88 of which a maximum of 50 will contain "NBJ" and a maximum of 60 will contain "JGE" at position 11 of the record text. The command is in a format suitable for saving in a command centre file and execution using the ACTION key (Shift-F4).

```
<EDIT CBL.USERS                                \
   FILTER (                                    \
      INCLUDE RECORD  LIMIT 50                 \
             WHERE(SUBSTR(RECORD,11,3) = 'NBJ') \
                                                \
      INCLUDE RECORD  LIMIT 40                 \
             WHERE(SUBSTR(RECORD,11,3) = 'JGE') \
                                                \
      STOPAFTER 88                             \
          )
```

Data Editor Primary Commands

The following Data Editor primary commands have been updated or included in SELCOPYi 3.50.

CHANGE/EXCLUDE/FIND/ONLY

Primary commands CHANGE, EXCLUDE, FIND and ONLY each involve use of a search string to locate values within the records being edited or browsed.

In previous releases, where a structure is used to map record data and multiple record-type mappings are in use (i.e. records have different layouts), then the search for the string specified by these primary commands will occur only in records mapped by the **default** record-type mapping. These are records that have the same record-type mapping as the focus record (the record on which the cursor is positioned, otherwise the first record in the current display area).

SELCOPYi 3.50 introduces support for keyword "**ANY**" that may be specified immediately before the search string in any of these primary commands. This operand will ensure that the search is not restricted to data mapped by the default record-type but all records of any record-type will be searched.

The following example will find the next occurrence of a field containing the character string "SYS1" in a record of **any** record-type mapping that follows.

```
FIND  ANY 'SYS1'
```

Keyword "**FOCUS**" may be used to revert back to searching only data assigned to the default record-type.

The default search option (ANY or FOCUS) is determined by the SET/QUERY/EXTRACT option RTSCOPE which may also be set via option input field "`F/C/X Scope>`" in the "Data Editor Settings" panel (=0.4).

ENUMS

Syntax:

```
>>--+-- ENUMs ----+----------------------+-----------------------------------><
    |             |    |                 |
    +-- ENUML ----+    +-- struct_name ---+
    +-- LENums ---+
    +-- LSENums --+
```

Description:

New in SELCOPYi 3.50, ENUMS will open a SELCOPYi List window to display the enum definitions for each record-type mapping in the specified SDO structure. If no structure name is specified, the structure for the current SDE window is used.

FILEIO

Syntax:

```
>>--+- FILEIO -+-- ddname ---+- OPENRead ----------------------+-------------><
    |          |             |                                 |
    +- FIO ----+             +- OPENWrite ---------------------+
                             |                                 |
                             +- CLOSE  ------------------------+
                             |                                 |
                             +- READ ----- rexxvar ------------+
                             |                                 |
                             +- CREAD ------------+---------+--+
                             |                    |         |  |
                             |                    +- nrecs -+  |
                             |                                 |
                             +- SREAD ---- stem --+---------+--+
                             |                    |         |  |
                             |                    +- nrecs -+  |
                             |                                 |
                             +- WRITE ---- record-text --------+
                             |                                 |
                             +- SWRITE --- stem ---------------+
```

Description:

New in SELCOPYi 3.50 for use in Data Editor REXX macros, FILEIO is used to perform file input/output on a physical sequential or VSAM data set, a PDS/PDSE library member or an HFS/ZFS file allocated to the specified DD name.

Unlike EXECIO, which is a TSO/E REXX extension, FILEIO may be executed in REXX macros run by SDEAMAIN (the SELCOPYi batch interface) or via a SELCOPYi VTAM login. It can also perform I/O on VSAM data sets and HFS/ZFS files which is not supported by EXECIO.

PRINTS

Syntax:

```
>>--- PRINTS ---- string ------------------------------------------------------><
```

Description:

New in SELCOPYi 3.50 and intended for use by SDEAMAIN (SELCOPYi batch) execution, PRINTS will output a specified line of text to DD SDEPRINT, where the first character is a valid ASA print character.

SELECT

Primary command SELECT is used in a view of formatted records to display a subset of fields belonging to records mapped by a specific record-type.

SELCOPYi supports structured browse and edit of records that are comprised of a single primary segment and zero or more secondary segments where each segment is mapped by a specific record-type definition. The field data for each record segment is displayed on a new line in the Data Editor table view. In single view mode, each record segment occupies the entire display.

For BROWSE processing of segmented records, the SELECT functionality has been enhanced so that one or more fields belonging to a secondary record segment may be displayed with the record's primary segment fields. This means that secondary segment fields may be displayed on the same line as the primary segment in a table format view and also appear together in the the same single format view display.

When selecting fields for display in a primary segment, a field from one of its secondary segments may be selected using a qualified field name. The high level qualifier is the secondary segment record-type name and the low level qualifier is the field name. If necessary, intervening qualifiers may be specified to distinguish the field from other fields of the same name within the secondary segment type.

The following example selects only the "zRTY" field from the record's primary segment and the "zJobname" field from its secondary segment mapped by record-type "SMF020_Job_Initiation" to be displayed together for primary segment data.

```
SELECT zRTY, SMF020_Job_Initiation.zJobname
```

TASK

Syntax:

```
>>-- TASK --- pgmname ---+------------------+---+----------------+-----------><
                         |                  |   |                |
                         +-- -LIB libpath --+   +-- -PARM parm --+
```

Description:

As for SELCOPYi interactive execution, TASK is now also supported a a Data Editor (SDE) command so that it may be used by SDEAMAIN (SELCOPYi batch) execution to start a program as a new sub-task.

VIEW

The VIEW command is used to select or deselect record-type mappings in the Data Editor display.

Records or record segments that are assigned a record-type mapping that has been deselected, will not be displayed and flagged as SUPPRESSED. If suppressed record-type shadow lines are set on, then groups of consecutive, suppressed records/record segments of the same record-type will display as a single suppressed record-type shadow line.

VIEW has been enhanced in SELCOPYi 3.50 so that a record-type name identifier may be specified with a trailing "*" (asterisk) symbol and so treated as a record-type name mask. The "*" symbol represents zero or more characters and so all record-types with names that match the mask will be identified by the VIEW operation.

In the example that follows, record-types with names beginning "SMF015" will be selected by the VIEW operation and so all records or record segments assigned one of these record-types will be **added** to the existing display of records/record segments.

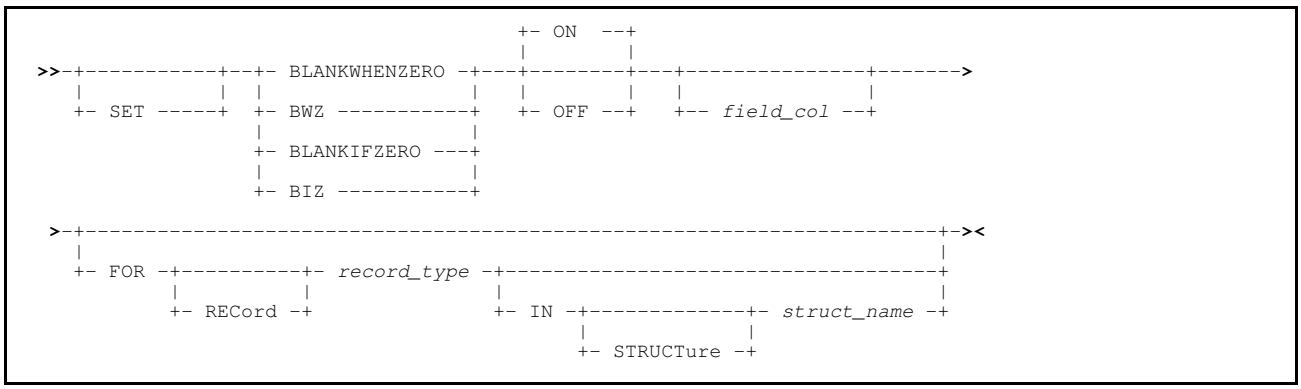
```
VIEW  +  SMF015*
```

Data Editor SET/QUERY/EXTRACT Options

The following Data Editor options have been included or updated in SELCOPYi 3.50.

BLANKWHENZERO - SET Only

Syntax:

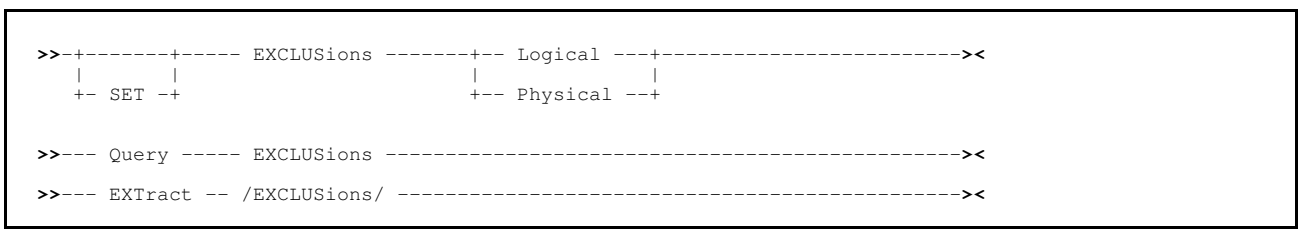


Description:

New in SELCOPYi 3.50, the BLANKWHENZERO option applies to fields with a numeric datatype and will cause a value of zero to be displayed as blank instead. Typically this is required when the value is predominantly zero, in order to make non-zero values stand out visually. BLANKWHENZERO will affect the field specified. Otherwise, it affects the focus field (field on which the cursor is positioned in the display area).

EXCLUSIONS

Syntax:



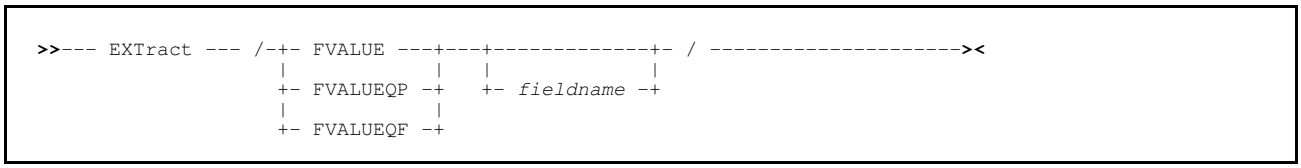
Description:

New in SELCOPYi 3.50, the EXCLUSIONS option applies only to the display of segmented records and determines the scope of an operation that includes/excludes lines from the display (e.g. primary commands EXCLUDE, FIND, etc).

If LOGICAL is set, then an individual segment within a record may be included or excluded. If PHYSICAL is set, then all segments of a record will be included or excluded when the include/exclude criteria is met in any of the record's segments.

FVALUE/FVALUEQF/FVALUEQP - EXTRACT Only

Syntax:



Description:

New in SELCOPYi 3.50 and for use in Data Editor macros, the FVALUEQP and FVALUEQF options perform the same function as FVALUE to generate a REXX variable with an assigned value for an individual field or all the fields selected for display in the focus record.

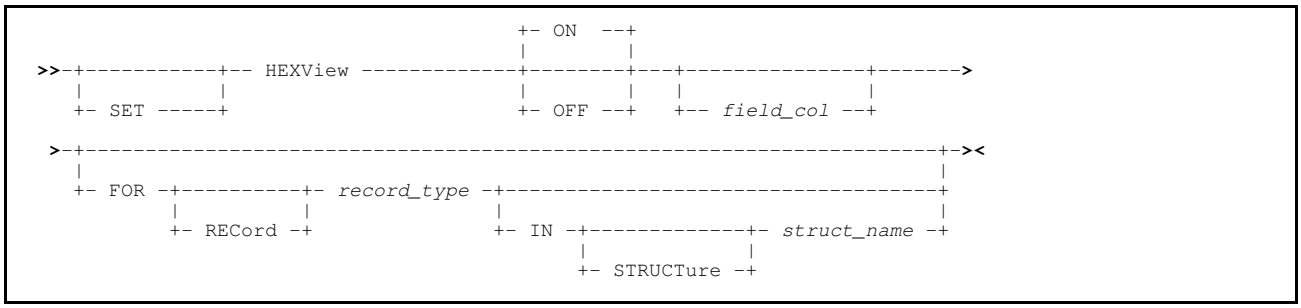
FVALUE will generate REXX variable names that match the elementary field name with no leading record-type or parent structure (group field) names. (e.g. `RELEASE_MM`)

FVALUEQP will generate REXX variables names that match the partially qualified field name but does not include the record-type name. The name qualifiers are separated by a "." (dot/period) and are the names of each parent structure (group field) in the group field hierarchy to which the field belongs (e.g. `RELEASE_DATE.RELEASE_MM`).

FVALUEQF will generate REXX variables names that match the fully qualified field name. It includes qualifiers that are the record-type name followed by the names of each structure (group field) in the hierarchy to which the field belongs. (e.g. `TRACK.RELEASE_DATE.RELEASE_MM`).

HEXVIEW - SET Only

Syntax:



Description:

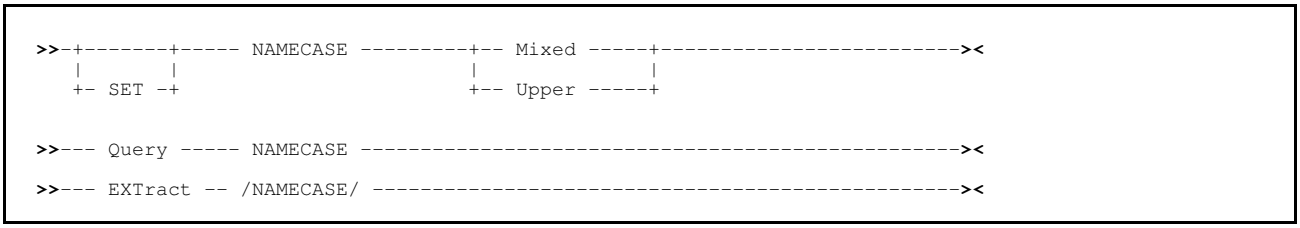
New in SELCOPYi 3.50 the HEXVIEW option is used in edit or browse of formatted record to display a field's source data in horizontal hexadecimal (data type HEXADECIMAL) format.

The "Hex Punctuation>" input field in the "Data Edit Settings" determines whether or not punctuation is used in the display of HEXADECIMAL format data. If so, a blank is inserted between every 4-bytes of source data and a "," (comma) inserted between every 2-bytes of source data. For example, the following hex value will be displayed for a character field containing "`** SELCOPYi 3.50`".

```
5C5C,40E2 C5D3,C3D6 D7E8,8940 F34B,F5F0
```

NAMECASE

Syntax:



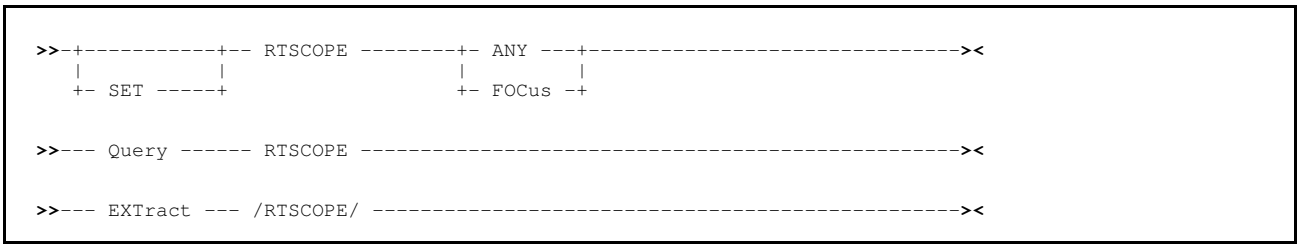
Description:

New in SELCOPYi 3.50, the NAMECASE option applies only to the XMLGEN and JSONGEN (XML and JSON generation) utilities. It defines whether or not field names used as XML and JSON tags are upper cased by default.

The NAMECASE option may be overridden by an option specified on primary commands or selected in the utility panel input fields.

RTSCOPE

Syntax:



Description:

New in SELCOPYi 3.50, the RTSCOPE option applies only where records in the display are assigned different record-types. The RTSCOPE values determines whether a search value specified on a FIND, CHANGE, EXCLUDE or ONLY operation may be found in records of ANY record-type or only records assigned the same record-type as the FOCUS record (i.e. the default record type).

The RTSCOPE option may be overridden by the individual FIND, CHANGE, EXCLUDE or ONLY primary command.

SELCOPYi File Copy

The SELCOPYi File Copy utility performs copy operations between file objects of different types and DCB geometry. SELCOPYi 3.50 includes a the following enhancements to the SELCOPYi File Copy Utility.

Record-type and Field Name Remap

For structured file copy with remap of records mapped by 2 different structures, SELCOPYi 3.50 introduces support for remap of a field that has a different name in the output file mapping to that in the input file mapping.

By default, where input and output file structures are specified, a connection will only be made between a field in the input and output file structures if the field name and the record-type name to which the field belongs is the same in both structures.

This new feature allows the user to connect a record-type in the input file structure with a record-type of a different name in the output file structure. If field names in the connected record-types are different, a field belonging to the input file record-type may be connected with a field of a different name in the output file record-type.

See parameters **MAPDIALOG**, **MAPRECORD [EXPLICIT]** and **MAPFIELD** in the syntax diagram for the FCOPY command and also use of the **MAP** command in the File Copy utility panel.

File Copy Utility Panel

The File Copy Utility panel has been updated to accommodate record-type and field name remapping features.

From the "File Reformat" view of the "File Copy" utility panel, enter primary command "MAP" to open the "Remap Record-Types" panel view. Here, you can choose how source and destination file record-types match up.

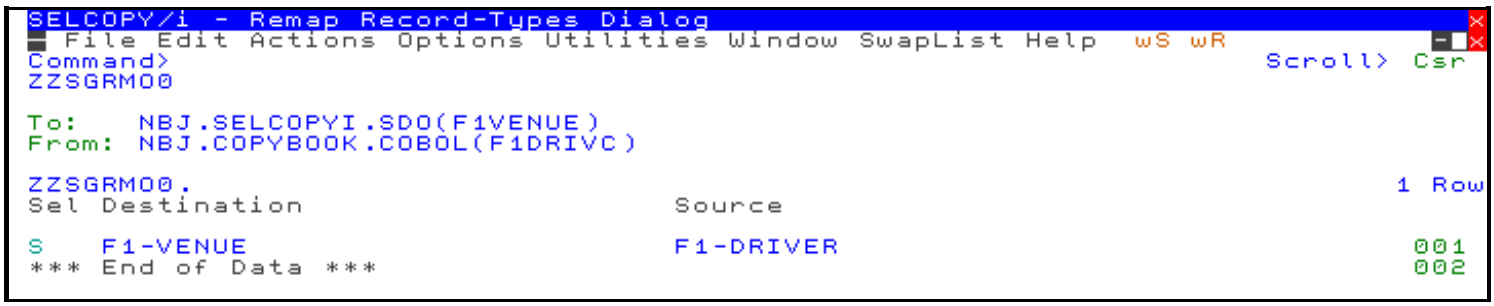


Figure 14. SELCOPYi File Copy Remap Record-Types panel view.

Select any of the source/destination table entries to open the "Remap Record Layout" panel view. This view displays all fields in the destination record-type. Source record-type field names may be entered (or selected from a list if wildcards are used) for each destination field as required.

```
SELCOPY/i - Remap record Layout Dialog
File Edit Actions Options Utilities Window SwapList Help  wS wR
Command>                                                  Scroll> Csr
ZZSGRMR0
To:    NBJ.SELCOPYI.SDO(F1VENUE)
       RECORD TYPE: F1-VENUE    FIXED(101) OFFSET=0 DATA ELEMENTS=13

From:  NBJ.COPYBOOK.COBOL(F1DRIVC)
       RECORD TYPE: F1-DRIVER    FIXED(85) OFFSET=0 DATA ELEMENTS=8

ZZSGRMR0.                                                      12 Rows
Lv Destination                          Type Len Source

 2 CODE                                 AN     3 FIRST-RACE-CIRCUIT      001
 2 COUNTRY                              AN    10 COUNTRY                 002
 2 TRACK                                AN    30                         003
 2 TYPE                                 AN    10                         004
 2 LAPS                                 BN     2                         005
 2 LAP-LENGTH-KM                        FB     2                         006
 2 TOTAL-DISTANCE-KM                    FB     4                         007
 2 TURNS                                BN     2                         008
 2 RACE-LAP-RECORD                      TI     4                         009
 2 RACE-LAP-RECORD-DATE                 DT     4                         010
 2 RACE-LAP-RECORD-HOLDER               AN    20 NAME                    011
 2 RACE-LAP-RECORD-TEAM                 AN    10                         012
*** End of Data ***                                                      013
```

Figure 15. SELCOPYi File Copy Remap Record Layout panel view.

FCOPY Command

Syntax:

Output:

```
>-- to_fileid ---+-------------------------------------------------------+->
                 |                                                       |
                 |         +- SDO ---+                                   |
                 |         |         |                                   |
                 +- USING -+---------+-- out_struct --+-------------------+
                           +- COBOL -+                |                   |
                           +- PL1 ---+                +--| Map Options |--+
                           +- ADAta -+
```

Map Options:

```
 |---+-----------------------------------------------+----+--------------+----|
     |                                               |    |              |
     +-- MAPRecord -----------| MapRecord Clause |--+    +-- MAPdialog -+
     |
     +-- MAPRecord EXPLICIT --| MapRecord Clause |--+
```

MAPRecord Clause:

```
       +-------------------------------------------------------------+
       v                                                             |
 |- (-+- OutRecTyp -+------------------+--+-----------------------+---+- ) --|
                    |                  |  |                       |
                    +- FROM InpRecTyp -+  +--| MapField Clause |--+
```

MAPField Clause:

```
    +-----------------------------------------------------------+
    v                                                           |
 |--+-- MAPField -- ( -- output_field -- FROM -- input_field -- ) -+---------|
```

Description:

Updated in SELCOPYi 3.50, the FCOPY primary command supports record-type and field name mapping options for formatted input and output records. These MAP options must be entered immediately following the output structure (*out_struct*) specification.

For example, the following command will copy **only** values from fields X1, X2 and X3 belonging to input records mapped by record-type XDATA, to fields A1, A2 and A3 respectively, in output records mapped by record-type ADATA.

```
<FCOPY  CBL.XXX   USING  COBOL  CBL.COPYBOOK.COBOL(XXSTRUCT)  \
        CBL.AAA   USING  COBOL  CBL.COPYBOOK.COBOL(AASTRUCT)  \
         MAPRECORD EXPLICIT                                   \
          (          ADATA FROM XDATA     /* Record-type */   \
            ( MAPFIELD(A1  FROM  X1)                          \
```

```
            MAPFIELD(A2  FROM  X2)                          \
            MAPFIELD(A3  FROM  X3)                          \
          )                                                 \
        )
```

SELCOPYi File Search & Update

The SELCOPYi File Search & Update utility is capable of performing a range of operations on both formatted and unformatted data in one or more data sets and/or library members:

1. File Search.
2. File Search, Change and Update.
3. File Search, Change and Copy to a different data set.
4. File Copy with Remap of record fields. (Formatted records only)
5. File Search, Change and Remap fields. (Formatted records only)

SELCOPYi 3.50 includes a the following enhancements to the SELCOPYi File Search & Update Utility.

Record-type and Field Name Remap

As for the File Copy utility, the File Search & Update utility supports structured file copy with remap of records mapped by 2 different structures. Similarly, SELCOPYi 3.50 introduces support for remap of a field that has a different name in the output file mapping to that in the input file mapping.

See description of File Copy "Record-type and Field Name Remap" for details of this feature.

Report Context Records

In order to be able to provide context in the utility's output report, SELCOPYi 3.50 introduces an option to report a number of records before and after a record containing a match for the search criteria.

When run in the SELCOPYi foreground, the utility's formatted report output may contain entries "**L**" and "**T**" in the **zT** column, which identify records that have been included as leading and trailing context lines respectively.

However, if the report is to be printed (the default for batch execution), then the zT column entries for context record lines are left blank. This has the benefit of making non-context lines within the plain text report (where colour highlighting is not possible) more easily identifiable. Note that non-context report lines have zT column entries "**H**", "**B**" or "**A**").

Restrict Input/Output

The File Search & Update utility supports input from multiple files and, for library copy operations, supports output to multiple members.

The utility has the capability to restrict the number of records read from **each** input file (**FOR** *nrecs*) and also restrict the number of records written to **each** library member (**LIMIT** *nrecs*).

SELCOPYi 3.50 introduces further control by providing the capability to restrict the number of records read across **all** input files (**MAXINPUT** *nrecs*), and the number of records written in total across **all** output members (**MAXOUTPUT** *nrecs*).

File Search & Update Utility Panel

The File Search & Update utility panel views have been updated to accomodate the new features introduced in SELCOPYi 3.50.

Both the "Basic File Search" panel view and "REPORT" panel view for "Extended File Search/Update/Copy/Remap Tasks" have been updated to include a "**Context>**" input field to include a specified number of report context records.

Extended task input "FILTER" and "OUTPUT" panel views now also contain "Max>" input fields to specify a maximum number of input and/or output records to be processed. Furthermore, "OUTPUT" panel views for formatted copy with or without remap, now support primary command **MAP** to open the "File Reformat" and "Remap Record Layout" panel views. (See "File Copy Utility Panel" for sample displays.)

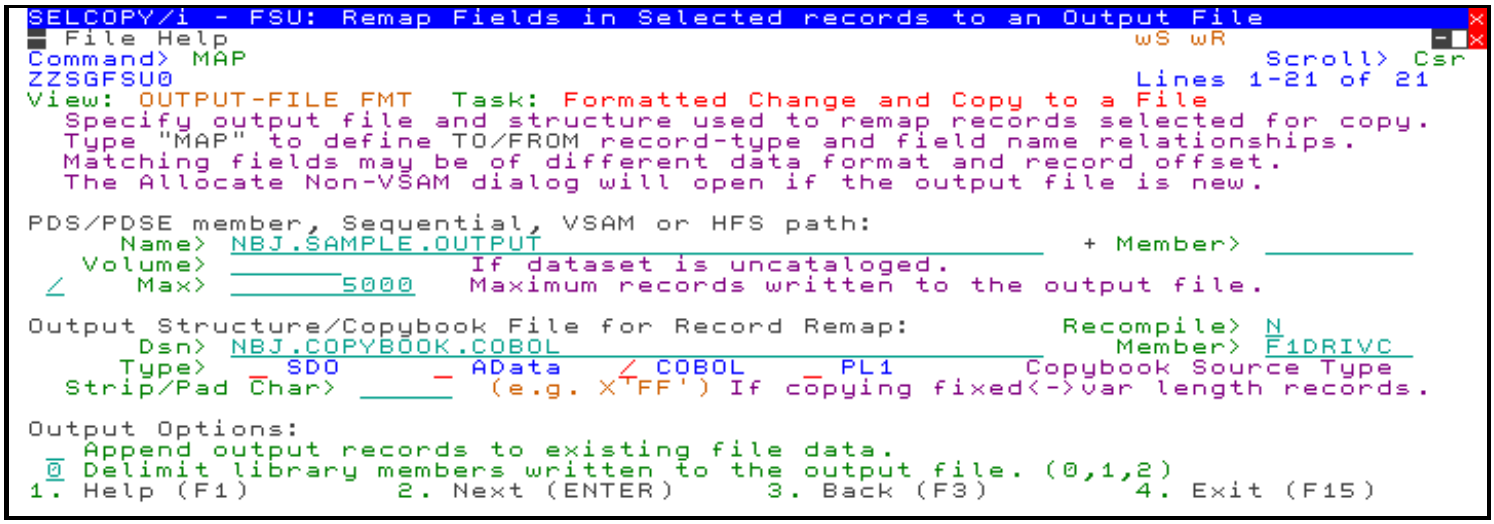
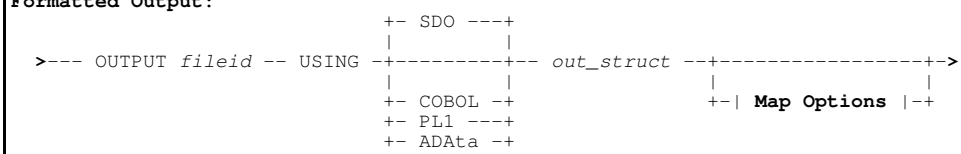


Figure 16. SELCOPYi File Search & Update Formatted OUTPUT panel view.

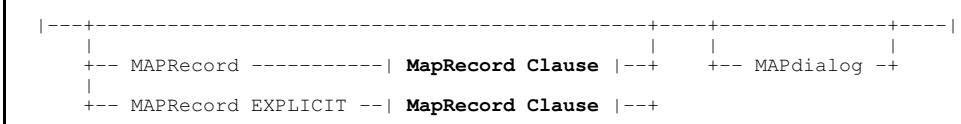
FSU Command

Syntax:

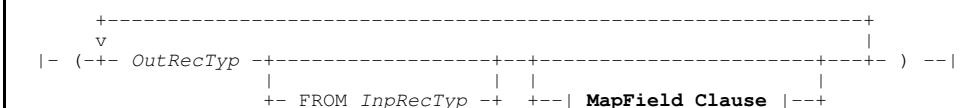
Formatted Output:



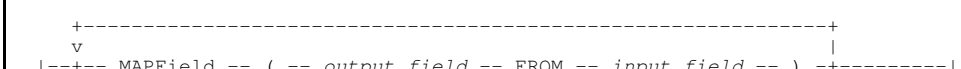
Map Options:



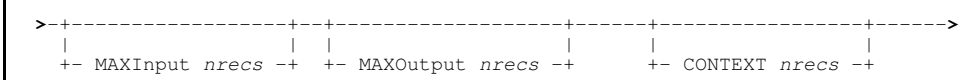
MAPRecord Clause:



MAPField Clause:



New Common Options:



Description:

Updated in SELCOPYi 3.50, the FSU primary command supports record-type and field name mapping options for formatted input and output records. These MAP options must be entered immediately following the output structure (*out_struct*) specification. See FCOPY Command for example of the MAPRECORD clause.

New options MAXINPUT, MAXOUTPUT and CONTEXT are also included and are common to all FSU utility operations.

SELCOPYi File Compare

The SELCOPYi File Compare utility is used to compare formatted and unformatted record data belonging to 2 input data set sources and produce a comprehensive report.

SELCOPYi 3.50 includes a the following enhancements to the SELCOPYi File Compare Utility.

Record-type and Field Name Map

The File Compare utility supports the compare of 2 data sets containing structured data where records from the first (new) data set are mapped using a different structure to that of the second (old) data set.

By default, where new and old file structures are specified, a connection will only be made between a field in the new and old file structures if the field name and the record-type name to which the field belongs is the same in both structures.

This new feature allows the user to connect a record-type in the new file structure with a record-type of a different name in the old file structure. If field names in the connected record-types are different, a field belonging to the new file record-type may be connected with a field of a different name in the old file record-type.

See parameters **MAPDIALOG**, **MAPRECORD [EXPLICIT]** and **MAPFIELD** in the syntax diagram for the COMPFILE command and also use of the **MAP** command in the Compare Files utility panel.

Embedded Blanks

SELCOPYi 3.50 introduces a new facility to allow a NEW and OLD file record pair to match if the only difference is the number of blanks between words in the text. i.e. An equal condition is set when the record pair contains differences but are equal after all blanks have been removed.

Furthermore, if this facility is not used, then a record pair that would be a match but for the number of embedded blank characters is now treated as a CHANGE rather than a DELETE and INSERT.

See parameter **SPACE** in the syntax diagram for the COMPFILE command and also field option "Ignore embedded blanks" in the "Unformatted Re-synchronisation Options" view of the Compare Files utility panel.

Mask Record Text

When comparing unformatted records from 2 data sets, you may want the utility to ignore differences that occur within specific areas of the records based on specified criteria. For example, when comparing 2 COBOL copybooks, you may wish to ignore differences in in the group-field name if the group-field level number starting in position 8 is "01".

SELCOPYi 3.50 introduces support for an **IGNORE** clause on the COMPFILE command which conditionally or unconditionally masks one or more areas of the compared records.

For the purpose of efficiency in read-ahead record synchronisation, the compare operation will overlay mask areas in a record with "-" (hyphon/minus) symbols. If necessary, this overlay character may be changed using parameter **IGNFILL**. If a conditional WHEN clause is specified, then NEW and OLD file records are each tested individually so that masking will only occur in a record if its text satisfies the WHEN clause criteria.

Therefore, if both the NEW and OLD records in the compare record pair contain masked areas, then any differences that would have existed within the original content of these areas will be undetected.

In the following example, 3 areas within a NEW or OLD file record will be masked if the record text contains "3" in position 1 and either "blues" or "tables" at any location. The mask area starting at position 131 for 999 bytes extends beyond the length of the record data and so is equivalent to masking from position 131 to the end of the record.

```
<COMPFILE                                          \
    JGE.SELCTRN.ZZST2DAT      /* New-file */       \
    NBJ.SELCTRN.ZZST2DAT      /* Old-file */       \
      readahead(100)                               \
      ignore( 11:8  21:6  131:999                  \
          when( left(record,1) =  '3'              \
               & (   record << 'blues'   \
                  or record << 'tables'      \
                 )                                 \
              )                                    \
          )
```

Compare Files Utility Panel

The Compare Files Utility panel has been updated to accommodate record-type and field name remapping features and also compare excluding embedded blanks.

From the "Formatted: Old file details and options" view of the "Compare Files" utility panel, enter primary command "MAP" to open the "Remap Record-Types" panel view. Here, you can choose how source and destination file record-types match up. (See "File Copy Utility Panel" for sample displays.)

The "Compare Files (unformatted): Re-synchronisation options" view has also been updated to include the "Ignore embedded blanks" field option.

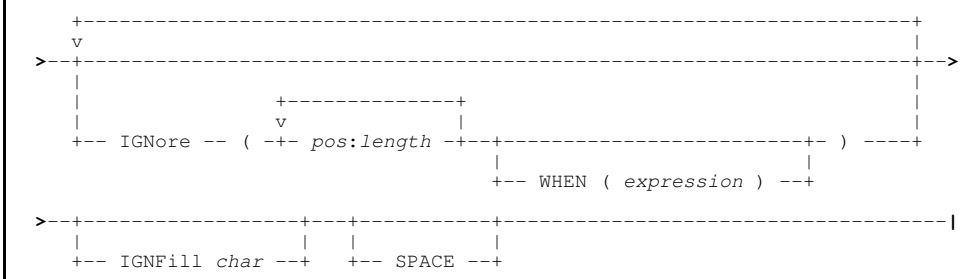
```
SELCOPY/i - Compare Files (unformatted): Re-synchronisation options
 File Help                                                    wS wR
Command>                                                          Scroll> Csr
ZZSGCF00                                                    Lines 1-21 of 21

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

Char Text Options:                                    _ Show Context
 _ Perform case-insensitive compare                     10 Lines Top/Bottom
 / Ignore embedded blanks                                1 Gap Lines

Report File:  (default is 'userid.SELCOPYI.COMPFILE.REPORT')
_      Dsn> ____________________________________________     Member> ________
   Volume> _______   If dataset is uncataloged.

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

Figure 17. SELCOPYi Compare Files (unformatted): Re-synchronisation options panel view.

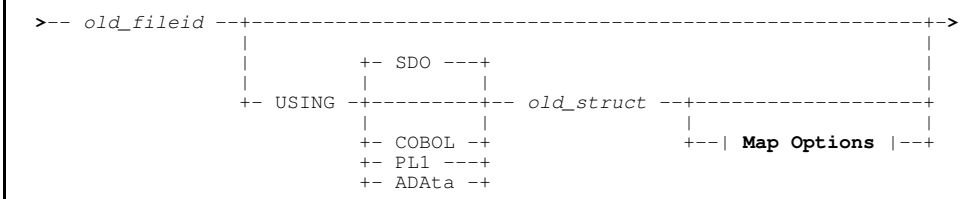
COMPFILE Command

Syntax:

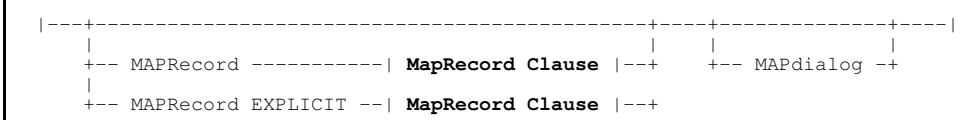
New Unformatted Compare Options:



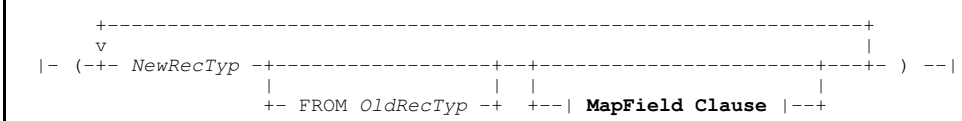
Old File Specification:



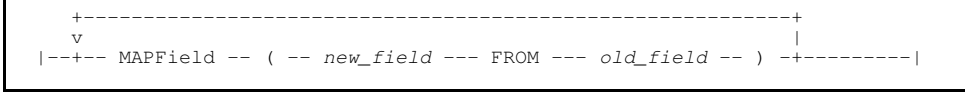
Map Options:



MAPRecord Clause:



MAPField Clause:



Description:

Updated in SELCOPYi 3.50, the COMPFILE primary command supports conditional and unconditional IGNORE clauses to mask areas of unformatted record text, and SPACE keyword to perform an unformatted compare with embedded blanks removed.

For formatted record data compare, it includes support for a MAPRECORD clause for record-type and field name mapping options between NEW and OLD file input records. These MAP options must be entered immediately following the OLD file mapping structure (*old_struct*) specification. See FCOPY Command for example of the MAPRECORD clause.

SELCOPYi Other Utilities

Other SELCOPYi utilities have either been introduced in SELCOPYi 3.50 or undergone minor changes.

XML and JSON Generation

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

This action is controlled by the TAGUPPER or NOTAGUPPER option specified on the XMLGEN or JSONGEN primary command, or by the "Tag Name Case>" option field in the "Output Text File" view of the utility panels.

The default action to either upper case or preserve tag alpha characters, is determined by the value of the NAMECASE Data Editor option.

SELCOPY Debug

To expand support of SELCOPY debug for a SELCOPY exceution running in an IMS DLI region, the following options have been added to the SELCOPY primary command syntax:

-DBRC [Y|N]
Indicates whether or not IMS Database Recovery Control is to be used in the DLI batch region. "-DBRC" is equivalent to "-DBRC Y".

-IRLM [Y|N]
Indicates whether or not IMS Resource Lock Manager (with default name IRLM) is to be used in the DLI batch region. "-IRLM" is equivalent to "-IRLM Y".

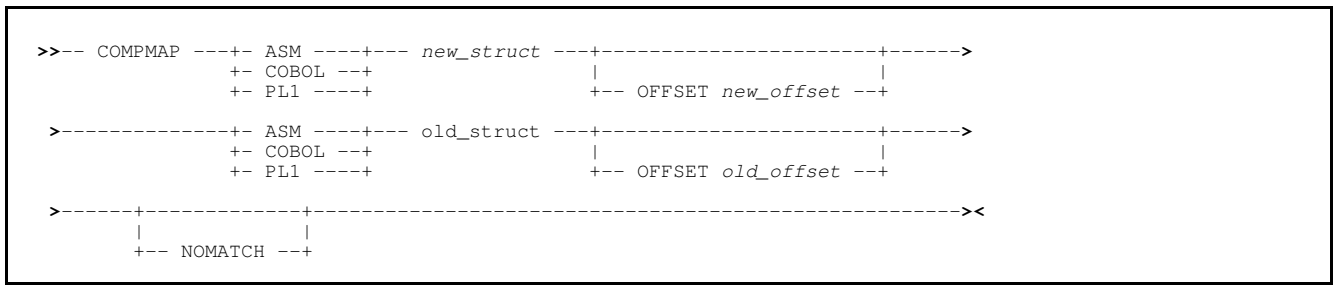
DB2 Execute SQL

The SELCOPYi EXECSQL facility to execute DB2 SQL statements, supports SQL statement input passed from a data set or library member.

In order to match IBM SPUFI operation, EXECSQL has been updated so that records in the SQL source files which contain "--" as the first non-blank characters, are treated as comment records. Note that SPUFI input members may be used as input to EXECSQL.

Compare Record Maps

Syntax:



Description:

New Compare Record Mapping utility (COMPMAP) introduced to compare the record layout mappings generated by 2 different copybooks and used to map the same record data.

The utility is designed to detect and highlight any differences that exist in the name, offset, length or datatype assigned to the same field definition obtained from the 2 copybooks. This is useful when highlighting changes that have occurred between NEW and OLD versions of the same copybook, or to ensure correlation between structures of different source types that map the same record data.

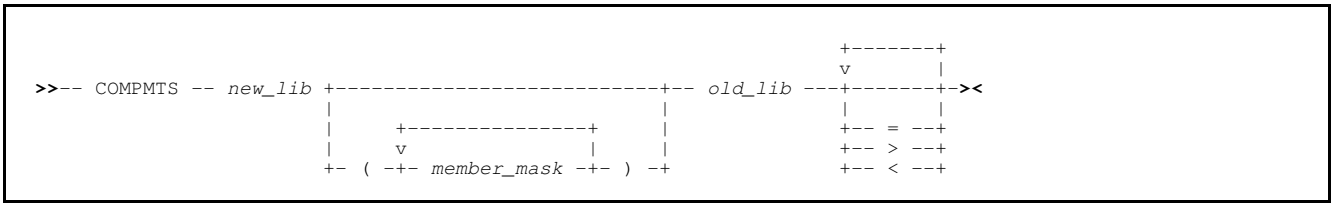
OFFSET values are necessary where record mapping begins at an offset into or before the start of the record data and parameter **NOMATCH** will suppress report of matching fields so that only mismatching fields are reported.

For example, suppose 2 copy books are maintained to map database records, one is in COBOL format, the other PL1. The COMPMAP utility may be executed against the 2 copybooks to ensure equivalence. Any mismatching fields are reported in the COMPMAP output.

```
<COMPMAP   COBOL   CBL.COPYBOOK.COB(XXSTRUCT)            \
           PL1     CBL.COPYBOOK.PL1(XXSTRUCT)            \
           NOMATCH
```

Compare Member Timestamps

Syntax:



Description:

New Compare Member Timestamps utility (COMPMTS) introduced to compare the timestamps of members belonging to different library data sets. The timestamp of a member in the NEW library will be compared against the timestamp of a member of the same name in the OLD library.

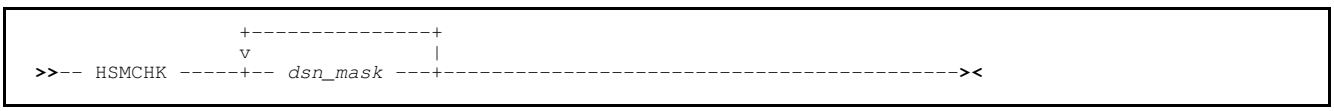
On completion, COMPMTS will open a Text Editor window displaying a report of all members found in each library and whether their timestamps match or mismatch, or they exist in one library but not the other.

Operands "=", ">" and/or "<" may be specified to restrict report entries to instances where timestamps are equal, NEW member timestamp is high and/or NEW member timestamp is low respecively.

This utility is particularly useful when managing backup copies of libraries.

Recall HSM Migrated Data Sets

Syntax:



Description:

New Recall HSM Migrated Data sets utility (HSMCHK) introduced. HSMCHK will generate a batch job to recall all data sets matching the DSN masks supplied. For example:

```
HSMCHK    USER001.**.JCL   USER001.**.TXT
```

SELCOPYi List Windows

SELCOPYi displays lists of data sets, library members, DB2 objects, etc. in list class windows. These types of windows share common attributes and support a common set of commands and operations.

SELCOPYi 3.50 includes a number of changes and enhancements to the SELCOPYi list class windows.

Delete & Rename

Data set, member and HFS/ZFS files may be deleted or renamed via their entry in a list window using line commands "D" and "R" respectively.

In previous releases of SELCOPYi, the list would be automatically refreshed following delete and refresh operations. This meant that deleted entries were removed from the list and renamed entries would be displayed with their new names (provided the new name satisfied any name mask filter used to create the list).

In SELCOPYi 3.50, this behaviour has been changed so that, on completion of a delete or rename operation, the list of entries is unchanged. Instead, the columns displaying additional information for these list entries are overlayed with text highlighting the action performed on the file object.

For a deleted item, the text is "**--- Deleted ---**" and is red in colour. For a renamed item, the text is "**--- Renamed ---**" and is green in colour.

The REFRESH command (or "Refresh" menu bar item) may be used to refresh the contents of the list.

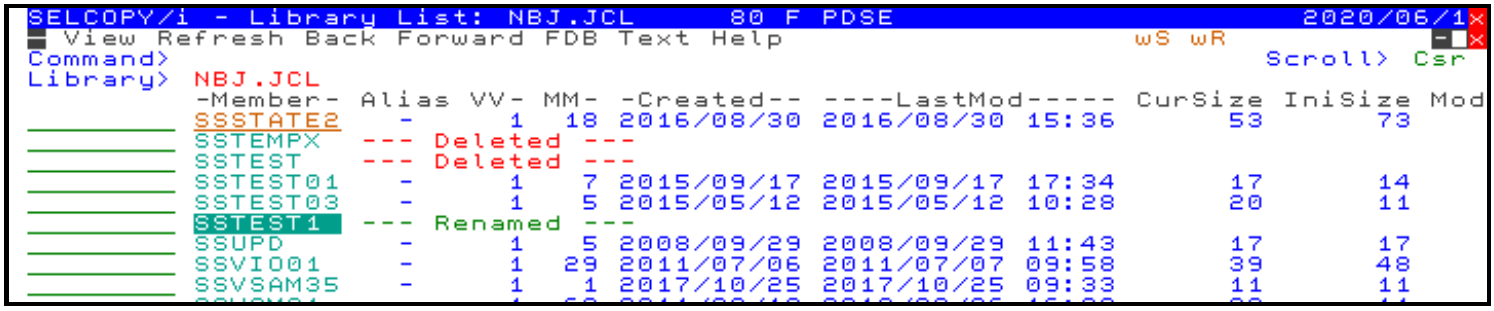


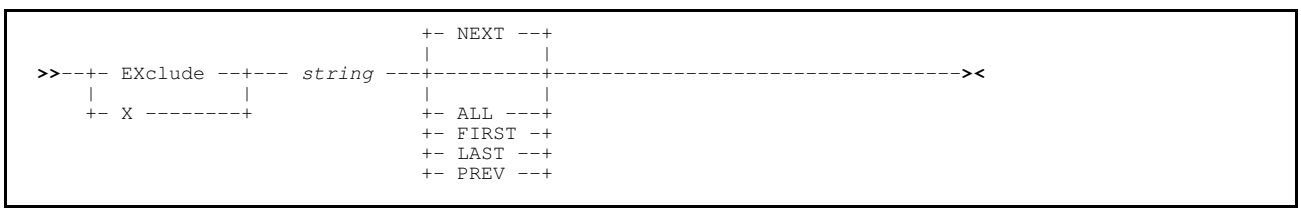
Figure 18. SELCOPYi Librray List - Deleted and Renamed Members.

List Primary Commands

The following List window primary commands have been updated or included in SELCOPYi 3.50.

EXCLUDE

Syntax:



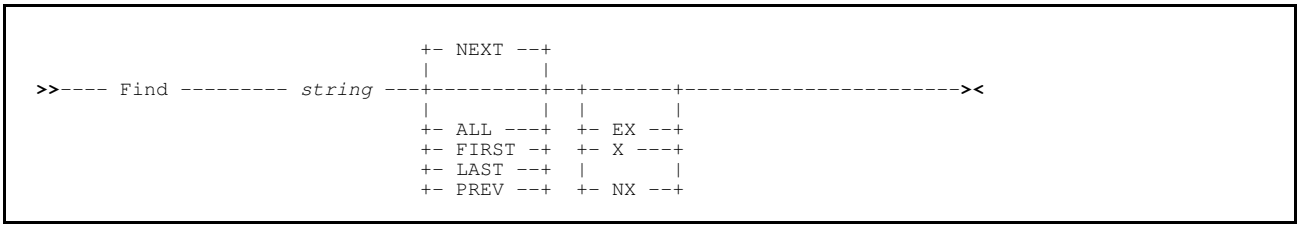
Description:

New in SELCOPYi 3.50, EXCLUDE will exclude lines that contain a character match for a specified search string within the first list column. Lines that are already excluded remain excluded following execution of EXCLUDE. Therefore, execution of successive EXCLUDE commands has a cumulative effect.

Parameter keywords (NEXT, ALL, etc.) have the same operation as for the EXCLUDE Text Editor primary command. RESET EXCLUDED will redisplay excluded lines.

FIND

Syntax:

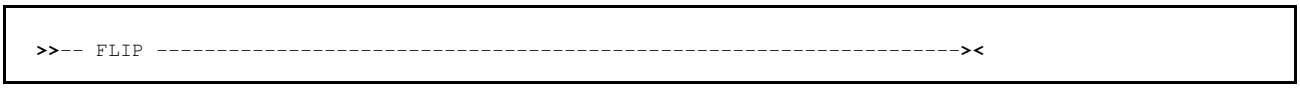


Description:

Updated in SELCOPYi 3.50, the FIND command now supports parameters **ALL**, **FIRST**, **LAST**, **NEXT** and **PREV**, and also **EX** (**X**) and **NX**. These parameter keywords have the same operation as for FIND in the Text Editor.

FLIP

Syntax:

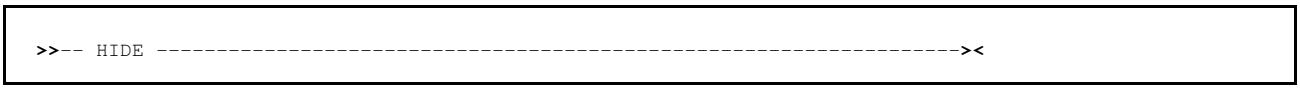


Description:

New in SELCOPYi 3.50, FLIP will reverse the status of excluded and non-excluded lines in the list window so that excluded lines are displayed and non-excluded lines are excluded.

HIDE

Syntax:



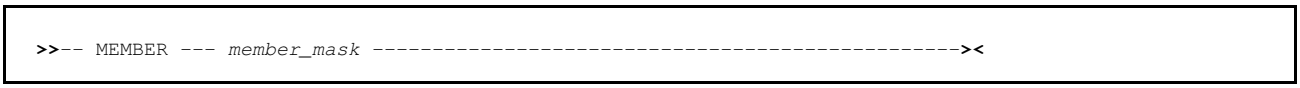
Description:

New in SELCOPYi 3.50, HIDE will hide all shadow lines that represent groups of one or more excluded lines in the list window. HIDE is equivalent to command SHADOW OFF.

RESET HIDE will redisplay all shadow lines.

MEMBER

Syntax:

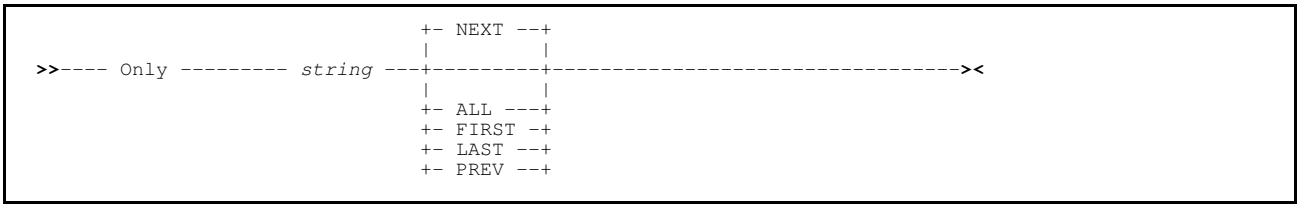


Description:

New in SELCOPYi 3.50, MEMBER is applicable only to lists with key columns containing data set names. MEMBER will exclude all list entries for data sets that are not libraries or do not contain at least one member whose name matches the specified member mask.

ONLY

Syntax:



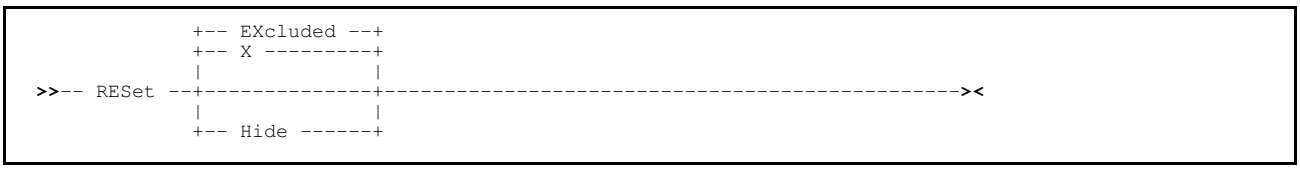
Description:

New in SELCOPYi 3.50, ONLY will first exclude all lines and then include all lines that contain a character match for a specified search string within the first list column.

Parameter keywords (NEXT, ALL, etc.) have the same operation as for the ONLY Text Editor primary command.

RESET

Syntax:

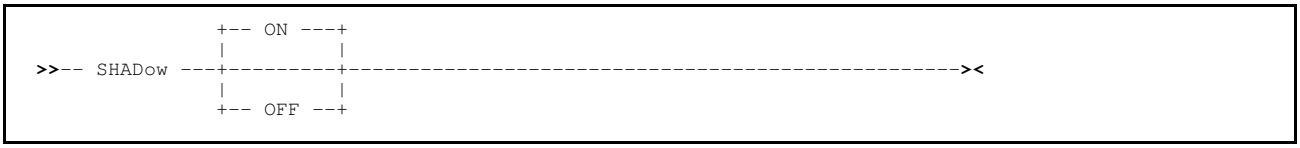


Description:

New in SELCOPYi 3.50, RESET or RESET EXCLUDED will include currently excluded lines. RESET HIDE is equivalent to SHADOW ON and will re-display shadow lines that have been hidden by a HIDE or SHADOW OFF command.

SHADOW

Syntax:

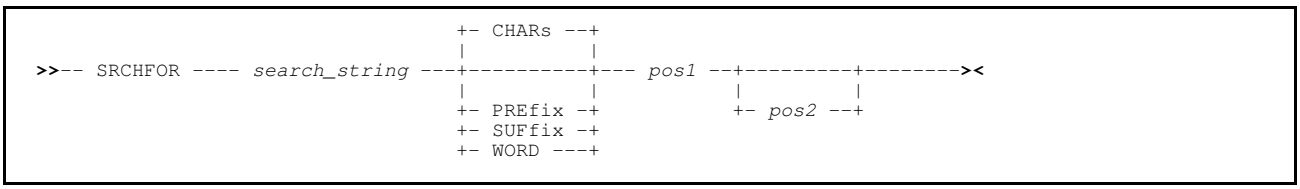


Description:

New in SELCOPYi 3.50, SHADOW will control display of shadow lines that represent groups of consecutive, excluded lines in the list window. SHADOW OFF is equivalent to command HIDE, SHADOW ON is equivalent to command RESET HIDE.

SRCHFOR

Syntax:



Description:

New in SELCOPYi 3.50, SRCHFOR is applicable only to lists of data sets, library members or HFS/ZFS files. SRCHFOR will execute the File, Search & Update utility to perform a search for the specified *search_string* in all files identified by non-excluded entries in the list.

List Line Commands

The following List window line commands have been included for all types of list windows in SELCOPYi 3.50.

NX / NXF / NXL

New in SELCOPYi 3.50, the "**NX**", "**NXF**" and "**NXL**" line commands are used to include previously excluded lines in the list. They are applicable only when entered against a shadow line in the list and so excluded shadow line display must be set on in order to utilise these line commands.

NX will include all excluded lines represented by a single shadow line.
NXF will include the **first** line in a group of excluded lines represented by a single shadow line.
NXL will include the **last** line in a group of excluded lines represented by a single shadow line.

X (//X //)

New in SELCOPYi 3.50, the "**X**" line command is used to exclude a line from the list display. "//**X**" may be entered against one list line and "//" entered against another to mark the first and last lines in a group of list lines to be excluded.

LISTCATALOG / LISTDATASET

In order to display information more familiar to users of the ISPF Data Set List Utility (option 3.4), the following amendments have been made to Catalog Lists and Dataset Lists in SELCOPYi 3.50.

VSAM DATA and INDEX

The list windows for cataloged entries and datasets have been updated to include a new input field "**VSAM Data+Ix>**". This field has a value of "Y" or "N" to indicate whether a VSAM cluster's DATA and INDEX components are to be included or omitted from the list, respectively. The VSAM CLUSTER entry is always displayed.

In previous releases, DATA and INDEX portions were always included in these types of list.

Volx Column

A new list column "**Volx**" has been introduced for display in place of the current "**Vol**" column. The "VolX" column has a display width of 7 and may contain one of the following:

volid	The volume id of the only volume on which the data set is saved.
*volid***+**	The volume id of the first volume on which the multi-volume data set is saved.
***ALIAS**	The name is an ALIAS of another data set.
MIGRAT1	The data set is HSM level 1 migrated (to DASD).
MIGRAT2	The data set is HSM level 2 migrated (to TAPE).
***PATH**	The name is a VSAM PATH defining a path between a VSAM alternate index (AIX) and its base CLUSTER.
***VSAM**	The name is a CLUSTER entry for a VSAM data set.

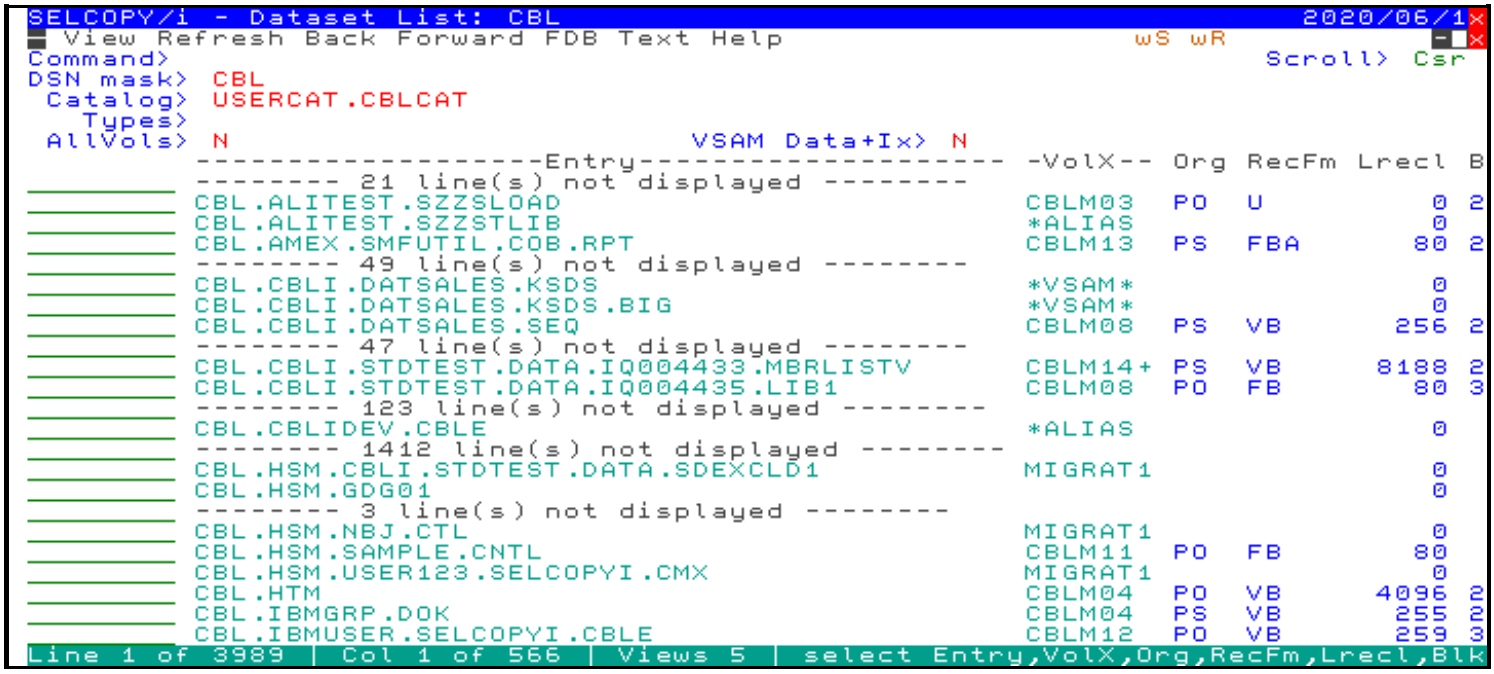


Figure 19. SELCOPYi Data Set List.

Maintenance Applied

The following table identifies maintenance to SELCOPY Product Suite 3.40 that has been applied at source to SELCOPY Product Suite 3.50.

Details of each SYSMOD may be found at the CBL web page entitled "SELCOPY Product Suite 3.40 Maintenance Summary". Individual descriptions are referenced by links to this web page in the following table.

Service Package Id	SYSMOD
X2018019	RI34001, RS34001, RS34002
X2018269	RS34003
X2019037	RS34004
X2019123	RI34002, RS34005
X2019189	RI34003, RS34006
X2019339	RS34007