



**Product Suite New Features**

**Release 3.20**

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

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

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

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

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

# Contents

<b>SELCOPY Product Suite 3.20 New Features.....</b>	<b>1</b>
Documentation Notes.....	1
<b>SELCOPY Product Suite Components.....</b>	<b>2</b>
SELCOPY/i.....	2
SELCOPY.....	2
SELCOPY (SLC) C++ Version.....	2
CBLVCAT.....	2
<b>SELCOPY/i Product Suite Documentation.....</b>	<b>3</b>
SELCOPY/i Training Material & Manual.....	3
SELCOPY/i Quick Reference.....	3
<b>SELCOPY/i ISPF Integration.....</b>	<b>4</b>
ISPF KEYS.....	4
ISPF SPLIT.....	4
SELCOPY/i Key List Tables.....	4
ISPF Edit Rexx Macros.....	5
<b>SELCOPY/i General New Features.....</b>	<b>6</b>
Regular Expressions.....	6
Usage.....	6
Syntax.....	7
Description.....	7
Examples.....	9
Syntax Elements and Operators Summary.....	10
XML Generation.....	12
Data Set Information.....	14
Define GDG Base panel.....	15
List VTOC Extents DSN filter.....	15
Allocate Non-VSAM Expiry.....	16
<b>SELCOPY/i Print Utility.....</b>	<b>17</b>
Print Utility Panel.....	17
Data Edit PRINT Primary Command.....	18
<b>SELCOPY/i FSU and FCOPY Utility Updates.....</b>	<b>19</b>
Panel Redesign.....	19
File Search, Update, Copy & Remap (=6).....	19
File Copy/Reformat (=5).....	20
FIND Condense Member list panel.....	21
Primary Command Updates.....	22
<b>SELCOPY/i Text Editor.....</b>	<b>23</b>
Processing Enhancements.....	23
ISPF Edit Rexx Macro support.....	23
Environment Variable %MCAT%.....	23
CMDTEXT Command Execution.....	23
Syntax Colour Highlighting.....	24
Primary Commands.....	24
GO.....	24
NOND.....	24
RUNSLC.....	25
SET/QUERY/EXTRACT Options.....	25
ACTIONCURSOR.....	25
ACTIONDELIM.....	25
COLOUR (COLOR).....	26
HCOLOUR (HCOLOR).....	26
HILIGHT (HILITE).....	27
INIVAR.....	27
<b>SELCOPY/i Structured Data Editor.....</b>	<b>28</b>
Panels.....	28
Data Browse/Edit (=2).....	28
Create STRUCTURE from COBOL/PL1 copybook(s) (=9.1).....	28
Create STRUCTURE from XREF File (=9.2).....	29
Display record layouts defined by an SDO/copybook (=9.3).....	29
Processing Enhancements.....	29
COBOL PICTURE symbol "P".....	29
PL1 PICTURE symbols "T", "I" and "R".....	30
Escaped Quotes.....	30
COBOL Level 88 condition support.....	30
Browse OCCURS DEPENDING array.....	30
Primary Commands.....	31
ARRC.....	31
ARRX.....	31
CHANGE.....	31

# Contents

<b>SELCOPY/i Structured Data Editor</b>	
COPY.....	32
CREATE.....	33
CREATE STRUCTURE.....	33
EXCLUDE.....	34
FIND.....	34
GO.....	35
GRPC.....	35
GRPX.....	35
HEXDUMP.....	36
LEFT.....	36
LOCATE.....	37
NOND.....	37
ONLY.....	37
RIGHT.....	37
REPLACE.....	38
SHIFT.....	38
XREFLIB.....	39
Line Commands.....	39
SET/QUERY/EXTRACT Options.....	39
ASCII.....	39
ARRAYALL.....	40
ARRAYASCHARACTER.....	40
BOUNDS.....	40
CCSID.....	41
COLOUR (COLOR).....	41
GROUPASCHARACTER.....	42
MAPPING.....	42
PREFIX.....	43
SAVEOPTIONS.....	43
SESSION.....	43
SIZE.....	44
WINPOS.....	44
WINSIZE.....	44
ZEROS.....	45
<b>SELCOPY/i DB2 Objects &amp; SQL Execution.....</b>	<b>46</b>
EXECSQL.....	46
Create Table Space.....	46
Create Table.....	47
<b>Maintenance Applied.....</b>	<b>48</b>

# SELCOPY Product Suite 3.20 New Features

---

## Documentation Notes

Information in this New Feature List details changes introduced to CBL software since SELCOPY Product Suite 3.10.

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

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

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

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

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

At the time of publication, this document is believed to be correct. CBL do not warrant that upward compatibility will be maintained for any use made of this program product to perform any operation in a manner not documented within the user manuals.

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

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

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

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

**All** - All MVS, VSE and CMS operating systems.

# SELCOPY Product Suite Components

---

SELCOPY Product Suite 3.20 includes updates to the following product components.

## SELCOPY/i

### z/OS Systems

SELCOPY/i has been updated extensively for z/OS systems. Details of SELCOPY/i 3.20 new features comprise the remainder of this document.

### z/VM and z/VSE Systems

Please be aware that, with the exception of the following, features of SELCOPY/i 3.20 detailed in this document are specific to **z/OS systems only**.

- ◇ **Regular Expressions**

## SELCOPY

The SELCOPY 3.20 (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:

- Default method QSAM used for output to non-VSAM z/OS data sets. Also support CBLNAME option SMVSOOutBSAM=Yes/No to change this default.
- Support VSAM control blocks loaded above the 16MB line. This is particularly pertinent to operation in z/VSE 4.3 systems and later.
- Latest ADABAS ADAUSER module, link edited as AMODE(31), is now supported as the SELCOPAD module used for ADABAS database access.
- SELCOPY DB2 interface for z/OS includes supports for WITH *common\_table\_expression* on READ SQL='full\_select' syntax.
- Correct 0C4 at EOJ following CALL to a Language Environment sub-routine when SELCOPY executed in TSO or under SELCOPYi (SELCOPY Debug.)

## SELCOPY (SLC) C++ Version

The SELCOPY (SLC) 3.20 C++ Version batch program has been updated with new features which are detailed in the [SELCOPY C++ Version \(SLC\) 3.20 New Features](#) document. This publication includes features for SELCOPY (SLC) on all supported platforms including Microsoft Windows and i386 Linux.

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

## CBLVCAT

The CBLVCAT 3.20 batch program has been updated with important fixes that are specific to both z/OS and z/VSE systems. These are as follows:

- A z/VSE sequential file defined with a retention period, no longer displays as \*EXPD\* (expired) following a MOD EXP=PERM action to make it permanent.
- Number of records (NRECS) now reports as 0 (zero) for VSAM data sets that have had records loaded and then all deleted. Previously, the HI Used RBA value would be used to estimate number of records and inserts with CI access assumed.
- LISTVTOC now correctly reports EXTENT information for z/VSE files on BIG and FAT 3390 DASD devices.

# SELCOPY/i Product Suite Documentation

---

SELCOPY/i 3.20 introduces two additional publications and accompanying sample data.

## SELCOPY/i Training Material & Manual

Introduced in SELCOPY/i 3.10 PTF RS00021, the "Tutorial" item on the Primary Options panel (=T) guides new users through initialisation of sample test data, copy books and record selection (filter) data sets. These data sets provide users with working illustrations of SELCOPY/i features demonstrated in the [SELCOPY/i Training Manual](#).

The sample data sets are duplicated for each user selecting "Tutorial" from their primary options panel and so may be freely updated, destroyed and recreated as required.

## SELCOPY/i Quick Reference

Introduced in SELCOPY/i 3.10 PTF RS00018, the Quick Reference summarises SELCOPY/i commands, options and default PFKey settings as arranged by application feature and is intended for any application developer or administrator who may use SELCOPY/i.

In addition to the Adobe PDF format, this document is also available via the product Help system. Primary command "QREF" entered at any SELCOPY/i window command prompt will display the on-line version of this publication.

Information in this document was extracted from the following publications:

- SELCOPY/i Reference and User Guide
- SELCOPY/i Text Editor (CBL) Manual
- SELCOPY/i Structured Data Editor Manual

# SELCOPY/i ISPF Integration

---

SELCOPY/i 3.20 has been overhauled so that the SELCOPY/i ISPF application panel is better integrated with other ISPF/PDF applications.

## ISPF KEYS

SELCOPY/i has only one ISPF panel, the contents of which are dynamic and under the control of the SELCOPY/i application (ISPF applid CBLI). Because SELCOPY/i displays its own panels as windows within this single panel, each SELCOPY/i window potentially having its own PFKey definitions, SELCOPY/i ISPF PFKeys are set to values that are interpreted by the program but are not standard ISPF PFKey functions (e.g. F9 <> SWAP).

In SELCOPY/i 3.20, the application ISPF panel has been updated to use a supplied ISPF table library entry for PFKey definitions which is shared and so cannot be updated by the user. Although the PFKey values are still set as non-standard ISPF PFKey functions, the introduction of SELCOPY/i Key List Tables (described below) allows these standard key assignments to be implemented for each SELCOPY/i window.

The SELCOPY/i panel ISPF key assignments may be displayed by executing "ISPF KEYS" in any SELCOPY/i window and must contain F1=PF1, F2=PF2, ..., F24=PF24.

## ISPF SPLIT

SELCOPY/i windows within the application ISPF panel now resize based on the depth of the display area available to the panel when it is one of the logical sessions involved in an ISPF SPLIT operation.

This change overcomes scrolling issues and potential loss of SELCOPY/i command prompt display when ISPF SPLIT has been actioned at a location within the 3270 display area.

## SELCOPY/i Key List Tables

SELCOPY/i 3.20 introduces function key definitions in SELCOPY/i key list tables which are analogous to ISPF panel key lists. Use of key lists replaces key assignments based on a SELCOPY/i window's class, a method which could not be easily managed by an installation's product administrator.

Each SELCOPY/i window is assigned the name of a key list which exists in the key definitions table SELCOPY/i library member (ZZSKEYS). This member is read from each library in the 3-tier SELCOPY/i table library chain. A window function key definition is established from the named key list which is maintained at the 3 table library levels:

1. PRODUCT - The product supplied key list definitions.
2. SHARED - Key lists Overrides the function key value assigned by the PRODUCT key list of the same name.
3. PRIVATE - Overrides the function key value assigned by the SHARED key list of the same name.

The SELCOPY Product Suite target library (SZZSDIST.TLIB) is the SELCOPY/i table library installed as part of the release 3.20 product package material. This library includes member ZZSKEYS which contains a table of all the **PRODUCT** default function key definition keylists used by SELCOPY/i windows.

During install of SELCOPY Product Suite, an installation specific copy of this library (SITE.TLIB) will be generated so allowing authorised users (administrators) to set installation specific **SHARED** key definitions for all users. Setting the SHARED assignment for a function key will override the equivalent function key assignment in the PRODUCT keylist definition of the same name. This installation specific table library should be protected by a security manager (e.g. RACF) to prevent update by non-authorised users.

When a standard user updates a function key assignment during a SELCOPY/i session (e.g. via the **KEYS** panel), a **PRIVATE** copy of the table library will be generated for the user with a DSN prefix equal to the defined SELCOPY/i user prefix. Setting the PRIVATE assignment for a function key will override the equivalent function key assignment in the SHARED keylist definition of the same name.

Key list assignments may be maintained using Option 1. KeyList of the **Function Keys Settings (=0.8)** panel, or the **KEYLIST** primary command. The **KeyLists Selection List** allows the user to set the key list for the focus window and/or open the KEYS panel for an individual keylist definition. If using the Function Key Settings panel, the function key assignments for each level of table library member may be edited or browsed in a data edit window.

SELCOPY/i 3.20 also supports the **PFSHOW** primary command as documented for ISPF, to show key assignments for the focus window at the bottom of the 3270 display.

## ISPF Edit Rexx Macros

The SELCOPY/i text editor already supports macros written in the Rexx programming language.

Unlike ISPF Edit, SELCOPY/i uses the same primary edit commands in its macros as would be entered at the text edit command prompt (i.e. it does not employ different edit command syntax for use in edit macros.) This is intended to make understanding the supplied macros in ZZSDIST.CBLE and developing new edit macros much easier for users of all abilities.

However, users at many installations already make use of Rexx macros written for the ISPF editor and, since re-writing these macros for SELCOPY/i text edit is not favourable, SELCOPY/i has been updated to, whenever possible, support execution of these same ISPF edit macros in its own text editor with no changes to the macro syntax.

To activate use of ISPF Edit Rexx macros, set ISR Macros to be "YES" in the Text Edit Settings panel (=0.3) and restart SELCOPY/i. When this option is set, any command entered at a text edit command prompt that is not recognised as a SELCOPY/i text edit command and is not identified as a SELCOPY/i text edit macro within the edit macro path, will result in a search of the SYSUEXEC and SYSEXEC library concatenations for a member of the same name. If found, the member will be loaded and executed as an edit macro.

With additional edit features such as UNDO/REDO and multiple edit views of the same data, in most cases the SELCOPY/i text editor may be used in preference to the ISPF Editor.



# SELCOPY/i General New Features

---

## Regular Expressions

Regular expressions introduce another dimension to searching and updating data in SELCOPY/i and may be used in the following utility primary commands:

- Text Edit Interface CBL: ALL, MORE, LESS and LOCATE.
- Text Edit Interface ISPF: FIND, CHANGE, EXCLUDE and ONLY.
- Data Edit: FIND, CHANGE, EXCLUDE and ONLY.
- FSU - File Search/Update/Copy/Remap Utility.

The use of regular expressions for complex pattern matching was first developed in the UNIX environment.

Implementations of regular expression syntax vary in the operations supported and the special characters used to specify these operations. Since the SELCOPY/i text editor includes many command features similar to the PC text editor **KEDIT** produced by **Mansfield Software Inc** we have based our version of regular expression syntax on that of **KEDIT**.

For information on **KEDIT** see web site <http://www.kedit.com>.

## Usage

Regular expressions are used as arguments to commands which search for string patterns in text data. These include the **FIND** and **CHANGE** commands of the text and structured data editors.

The syntax and meaning of a regular expression is the same in all contexts in which it can be used, but the way it is specified depends on the command.

The following table shows how regular expressions are used in different contexts. The term **regexp** represents the regular expression string itself.

Command	Syntax
Text edit KEDIT style LOCATE	locate r / <b>regexp</b> /
Text edit KEDIT style CHANGE	change r / <b>regexp</b> /change_string/
Text edit KEDIT style ALL	all r / <b>regexp</b> /
Text edit ISPF style FIND	find r' <b>regexp</b> '
Text edit ISPF style CHANGE	change r' <b>regexp</b> ' change_string
Text edit ISPF style EXCLUDE	ex r' <b>regexp</b> '
Structured browse or edit FIND	find r' <b>regexp</b> '
Structured edit CHANGE	change r' <b>regexp</b> ' change_string
Structured browse or edit WHERE	where find("r' <b>regexp</b> '")
File Search/Update/Copy/Remap Utility (FSU) FIND	find( r' <b>regexp</b> ' )
File Search/Update/Copy/Remap Utility (FSU) CHANGE	change( r' <b>regexp</b> ' change_string)

Note that in the **CHANGE** commands the *change\_string* is not itself a regular expression but may contain tag references to tagged sub-expressions of the regular expression search pattern.

In what follows only the syntax of the regular expression **regexp** is discussed rather than the syntax of the search command which uses it.



**character\_class**

A character class is a set of one or more characters in the range `\x00-\xff`. It is a text specifier which matches a single character in the target string when that character is one of those in the class.

The class definition is a list of character specifications enclosed within delimiters. It is **inclusive** when delimited by `[ ]` (square brackets) or **exclusive** when delimited by `[~ ]` (a logical not operator `~` immediately following the opening square bracket). When the class definition is **exclusive** the characters in the class are all those in the range `\x00-\xff` **except** those in the definition list.

The class definition list consists of single characters and character ranges.

Single characters can be given in any of the forms described above for single character text specifiers.

Character ranges are defined as a range start character followed by a hyphen followed by a range end character both of which can be given in any of the forms described above for single character text specifiers. Ranges can be specified in either order, so `0-9` and `9-0` both define the set `0123456789` of numeric digits.

The list of characters included in a range depends of the type of the start and end characters:

- ◇ When both ends are lower case letters the range consists only of the lower case letters alphabetically between the range ends. So `h-k` defines the characters `hijk` even though in EBCDIC there are code points between `i` (`\x89`) and `j` (`\x91`) which do not represent letters.
- ◇ Similarly when both ends are upper case letters the range consists only of the upper case letters alphabetically between the range ends.
- ◇ When the range ends are numeric digits the range consists of the numeric digits between the range ends.
- ◇ When the range ends are of different types the range consists of all code points between the range ends considered as hexadecimal values. So `a-Z` defines all characters with hexadecimal code points between `\x81` and `\xE9` inclusive, regardless of whether they represent letters.

**?** The **question mark** character is the wild card character text specifier. It matches any character and so is equivalent to the character class `[ \x00-\xff ]`.

**^** The **caret** character is the start of string text specifier. When this text specifier is encountered in applying a regular expression to a string it is only considered a match if the current position in the target is the start of the string.

**\$** The **dollar sign** character is the end of string text specifier. When this text specifier is encountered in applying a regular expression to a string it is only considered a match if the current position in the target when matching is complete is the end of the string.

*predefined\_expression*

Several commonly used regular expressions have been supported as predefined expressions and can be referred to with a shorthand consisting of the predefined expression operator `:` (colon) followed by a single lower case letter. They can be used as the regular expression itself or as a sub-expression in a more complex regular expression.

**( )** Parentheses delimit a sub-expression.

Parentheses can be used to group terms together to which the logical NOT operator `~` or any of the pattern repetition operators minimal closure `*`, minimal plus `+`, maximal closure `@`, maximal plus `#`, and power `^` apply.

For example:

`ab#` matches an "a" followed by one or more "b"s such as "ab" "abbb"  
`(ab) #` matches one or more "ab"s such as "ab" "ababab"

Parentheses must be used to enclose an **alternation** which is a list of regular expressions separated by logical OR signs `|` (vertical bar). When an alternation is encountered in applying a regular expression to a string each of the sub-expressions is tried in turn at the current position in the target string until one matches or they all fail.

For example:

`(a|b|c)` would match an "a" or a "b" or a "c" at the current position in the target string.

**{ }** Braces delimit a tagged sub-expression. Up to 9 sub-expressions may be tagged.

Tagged sub-expressions are used to enable reference to be made to that part of the target string which was matched by the sub-expression. When the regular expression is parsed from left to right each tagged sub-expression is given a reference number starting with 1. When the regular expression is being matched, if the tagged sub-expression matches part of the target string, the matching part is saved and may be used in a tag reference.

**&n** The tagged expression reference operator **&** (ampersand) is used to refer to the string matched by a previous tagged sub-expression. The reference number **n** must be in the range 1-9.

For example in the expression:

```
{ [a-zA-Z] } [0-9] &1
```

the term **[a-zA-Z]** defines a character class of the lower and upper case alphabet, the term **[0-9]** defines a character class of the numeric digits, and the term **&1** refers back to the characters that were matched by the first tagged expression in braces, **{[a-zA-Z]}**.

This regular expression would match 3-character strings such as **a1a** and **X9X** where the first and last characters are the same alphabetic character and the middle character is a numeric digit.

The following operators are used to specify repetitions of a pattern. They apply to the preceding term or sub-expression of the regular expression.

- \* The **asterisk** is the minimal closure operator. It applies to the term preceding it and specifies that zero or more occurrences be included in the match, but only the minimum number.
- + The **plus sign** is the minimal plus operator. It applies to the term preceding it and specifies that one or more occurrences be included in the match, but only the minimum number.
- @ The **at sign** is the maximal closure operator. It applies to the term preceding it and specifies that the maximum of zero or more occurrences be included in the match.
- # The **hash sign** is the maximal plus operator. It applies to the term preceding it and specifies that the maximum of one or more occurrences be included in the match.
- <sup>n</sup> The **caret** character when followed by an integer is the power operator. It applies to the term preceding it and specifies that exactly **n** occurrences be included in the match. The repetition factor **n** must be in the range 1-999.

## Examples

### Alternation (or) operator

The alternation (or) operator **|** is used to choose between a series of possible matches. The alternative sub-expressions are separated by or signs and enclosed in parentheses.

Expression	String	Match
(u v)	xxxabbbbvyy	No match
(u v)	xxxvccccccyy	v

### Not operator

The not operator **~** applies to the following term in the expression. It specifies that the regular expression fails to match if the negated term matches at the current position in the target string.

Expression	String	Match
a~b	xxxabbbbvyy	No match
a~b	xxxacccccyy	a

### Minimal closure

The minimal closure operator **\*** specifies zero or more repetitions of the term it applies to but a minimal number. As such it is only useful when used for a term which is not the last in the pattern.

Expression	String	Match
ab*	xxxabbbbvyy	a
ab*c	xxxabbbbccyy	abbbbc
ab*c	xxxacccccyy	ac

### Maximal closure

The maximal closure operator **@** specifies zero or more repetitions of the term it applies to but a maximal number.

Expression	String	Match
ab@	xxxabbbbvyy	abbbb
ab@c	xxxacccccyy	ac

### Minimal plus

The minimal plus operator **+** specifies one or more repetitions of the term it applies to but a minimal number.

Expression	String	Match
ab+	xxxabbbbbyyy	ab
ab+c	xxxacccccccyyv	No match
ab+c	xxxabbbbcccyv	abbbbc

**Maximal plus**

The maximal plus operator # specifies one or more repetitions of the term it applies to but a maximal number.

Expression	String	Match
ab#	xxxabbbbbyyy	abbbb
ab#c	xxxacccccccyyv	No match
ab#c	xxxabbbbcccyv	abbbbc

**Syntax Elements and Operators Summary**

The regular expression character string consisting of delimiters, operators and text specifiers. Apart from characters which represent themselves as text specifiers, regular expression syntax uses special characters as delimiters, operators and special sorts of text specifiers.

**Special Characters Used in Regular Expressions:**

The following characters have a special meaning as operators or delimiters in regular expression syntax:

Character	Name	Description
\	Backslash	The escape operator
(	Left Parenthesis	Delimits the start of a sub-expression or alternation
)	Right Parenthesis	Delimits the end of a sub-expression or alternation
[	Left Bracket	Delimits the start of a character class definition
]	Right Bracket	Delimits the end of a character class definition
{	Left Brace	Delimits the start of a tagged sub-expression
}	Right Brace	Delimits the end of a tagged sub-expression
?	Question Mark	The wild card character text specifier
~	Tilde	The NOT operator
^	Caret	The power operator and the start of line text specifier
\$	Dollar Sign	The end of line text specifier
*	Asterisk	The minimal closure operator
@	At Sign	The maximal closure operator
#	Hash	The maximal plus operator
	Verical Bar	The alternation (or) operator
:	Colon	The predefined expression operator
&	Ampersand	The tagged expression reference operator
+	Plus Sign	The minimal plus operator
-	Minus Sign	The character range operator

**Text Specifiers:**

Text specifiers are used to match characters:

Text Specifier	Name	Description
character	Literal character	Any character (including blank) which is not a regular expression <b>special characters</b> .
[...] or [~...]	Character class	A set of characters and/or character ranges enclosed in square brackets.
?	Wild card	The wild card character which matches any character.
\character	Escaped Character	The escape operator followed by any one of the <b>special characters</b> used in regular expression syntax represents that character as a literal.
\letter	Symbolic Escaped Character	A number of control characters or special characters can be specified using a <b>symbolic escaped character</b> for convenience. This consists of the escape operator followed by a single lowercase letter.
\xnn	Hex character	The escape operator followed by x or X and a two digit hexadecimal number defines a literal character by its hexadecimal code value.

<b>^</b>	Match start	The caret character matches the start of the compare string
<b>\$</b>	Match end	The dollar sign matches the end of the compare string

**Operators:**

Operators are used to affect the way text specifiers or sub-expressions are processed or to combine sub-expressions to build more complex regular expressions.

Operator	Name	Description
*	Minimal Closure	Match zero or more occurrences of the preceding expression but only as many as necessary.
+	Minimal Plus	Match one or more occurrences of the preceding expression but only as many as necessary.
@	Maximal Closure	Match zero or more occurrences of the preceding expression matching as many as possible.
#	Maximal Plus	Match one or more occurrences of the preceding expression matching as many as possible.
<b>^n</b>	Power	Match n occurrences of the preceding expression.
~	Not	The not operator applies to the following expression. It allows the specification of a pattern in terms of the negation of a match.
()	Parentheses	Parentheses are used to define sub-expressions.
	Alternation	The alternation (or) operator matches one of a series of expressions enclosed in parentheses.
{ }	Tagged Expressions	Braces are used to identify a tagged sub-expression. When the tagged expression is matched, the matching text can be referred to with a tag reference. Tagged expressions are identified by a sequence number assigned left to right starting at 1.
<b>&amp;n</b>	Tag Reference	A tag reference represents the matched text of the associated tagged expression.
<b>:letter</b>	Predefined Expression	A <b>predefined expression</b> is represented by a colon followed by a lowercase letter. It represents a shorthand form of commonly used regular expressions which can be used by themselves or as sub-expressions.

**Predefined Expressions:**

The following predefined expressions are supported:

Name	Definition	Description
<b>:a</b>	[a-zA-Z0-9]	Alphanumeric character
<b>:b</b>	([\t\x40]#)	White space (a string of blanks and tabs).
<b>:c</b>	[a-zA-Z]	Alphabetic character
<b>:d</b>	[0-9]	Numeric digit
<b>:g</b>	(("[~""]@) ("[~""]@'))	Quoted string (in single or double quotes).
<b>:w</b>	([a-zA-Z]#)	Word (a string of alphabetic characters).
<b>:z</b>	([0-9]#)	Integer (a string of numeric digits).

**Symbolic Escaped Characters:**

The following symbolic escaped character sequences are supported:

Symbol	Character	Description
<b>\a</b>	'	Apostrophe (single quote)
<b>\g</b>	"	Double quote
<b>\b</b>	X'16'	EBCDIC control BS backspace
<b>\f</b>	X'0C'	EBCDIC control FF formfeed
<b>\r</b>	X'0D'	EBCDIC control CR carriage return
<b>\l</b>	X'15'	EBCDIC control NL new line
<b>\n</b>	X'25'	EBCDIC control LF linefeed
<b>\t</b>	X'05'	EBCDIC control HT horizontal tab



**Description:**

The XMLGEN command is used to produce an exportable copy of a structured dataset as XML text.

XMLGEN reads an input structured dataset and writes an output text dataset consisting of XML tags and tag content. The tag names correspond to the field names of the structure applied to the input dataset and the tag content to the field values expressed in character format.

The hierarchy of record types, group fields and elementary fields in the structure is represented by a corresponding nesting of XML tags, with the output enclosed within `<INPUT>` and `</INPUT>` tags.

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

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

**Examples:**

In the following example the COBOL copybook CBL.COB(XMLEXAMP) is used to map the 3 records in the file CBL.XMLEXAMP

The content of the COBOL copybook CBL.COB(XMLEXAMP) is shown as it would appear in text edit:

```
-CBL.COB(XMLEXAMP)      80 F PDSE      Size=10      Alt=0,0;1
<---+---1---+---2---+---3---+---4---+---5---+---6---+---7---
000000 * * * Top of File * * *
000001 ** CBL.COB(XMLEXAMP) ***      L=001 --- 2013/10/08 14:45:56 (CBL)
000002
000003      01 XMLExample.
000004      05 Employee.
000005          07 FirstName          Pic x(20).
000006          07 LastName           Pic x(20).
000007          05 Age                Pic s9(3) comp.
000008          05 Salary             Pic s9(7)v99 comp-3.
000009
000010
000011 * * * End of File * * *
```

The content of the file CBL.XMLEXAMP is shown as it would appear in structured data edit with HEX ON in effect to show the content of numeric fields:

```
-Edit  CBL.XMLEXAMP using CBL.COB(XMLEXAMP)          47 F SEQ
00000000 *** Top of Data ***
Record type: XMLEXAMPLE      Fixed(47) Offset=0 Data elements=6

      FIRSTNAME          LASTNAME          AGE          SALARY
      #3                 #4                 #5           #6
<---+---1---+---> <---+---1---+---> <---+> <---+---1>
00000001 John          Doe          52          33000.00
D9894444444444444444 C9844444444444444444 03          03000
16850000000000000000 46500000000000000000 04          0300C

00000002 Amy          Johnston     28          41500.00
C9A44444444444444444 D989AA99444444444444 01          04500
14800000000000000000 16852365000000000000 0C          0100C

00000003 Freda         Bloggs      39          27800.00
C9888444444444444444 C9988A44444444444444 02          02800
69541000000000000000 23677200000000000000 07          0700C

00000004 *** End of Data ***
```

The following XMLGEN command then produces the output file CBL.XMLGEN:

```
xmlgen  indsn(cbl.xmllexamp using cobol cbl.cob(xmllexamp))
        outdsn(cbl.xmlgen)
        nocomment indent 3 edit
```

The XMLGEN output file is edited as a result of the EDIT keyword parameter in the command. Note that each record has an associated record type (level 1) tag XMLEXAMPLE with the SEQ\_NUMBER attribute identifying the record number:



```

-CBL.XMLGEN      27990 V SEQ   Size=26   Alt=0,0;0
<---+-----1-----2-----3-----4-----5-----6-----+--
000000 * * * Top of File * * *
000001 <INPUT FILE="CBL.XMLEXAMP" FORMAT="CBL.COB(XMLEXAMP)">
000002   <XMLEXAMPLE SEQ_NUMBER="1">
000003     <EMPLOYEE>
000004       <FIRSTNAME>John</FIRSTNAME>
000005       <LASTNAME>Doe</LASTNAME>
000006     </EMPLOYEE>
000007     <AGE>52</AGE>
000008     <SALARY>33000.00</SALARY>
000009   </XMLEXAMPLE>
000010   <XMLEXAMPLE SEQ_NUMBER="2">
000011     <EMPLOYEE>
000012       <FIRSTNAME>Amy</FIRSTNAME>
000013       <LASTNAME>Johnston</LASTNAME>
000014     </EMPLOYEE>
000015     <AGE>28</AGE>
000016     <SALARY>41500.00</SALARY>
000017   </XMLEXAMPLE>
000018   <XMLEXAMPLE SEQ_NUMBER="3">
000019     <EMPLOYEE>
000020       <FIRSTNAME>Freda</FIRSTNAME>
000021       <LASTNAME>Bloggs</LASTNAME>
000022     </EMPLOYEE>
000023     <AGE>39</AGE>
000024     <SALARY>27800.00</SALARY>
000025   </XMLEXAMPLE>
000026 </INPUT>
000027 * * * End of File * * *

```

## Data Set Information

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

A file's information panel may be displayed using either of the following methods:

1. From a file list, enter prefix command "I" against the required file name.
2. Enter primary command "**DSINFO fileid**" where *fileid* is the name of the required data set or HFS file path.

Depending on the data set type, information may span several panel views. Press <Enter> to display the next panel view and primary command **BACK** (F3 default) to redisplay the previous panel view.

```

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

Data Component Information:
Device Type: 3390
Organisation: KSDS
KSDS Key Length: 6
KSDS Key Offset: 0
Average Record Length: 64
Maximum Record Length: 100

DEFINE Options:
Load Option: SPEED
Share Options: 1 - 3
Buffer Space: 40960
Write Check: NO
Erase on Delete: NO
Imbedded Index: NO
Replicated Index: NO
Reuse Option: NO
Spanned Records: NO
Key Ranges Present: NO
AIX-Unique keys:
AIX-Upgrade:

Allocated Space: Unit Primary Secondary
Data: Tracks 1 1
Index: Tracks 1 1

Dataset Date Information:
Created: 2008/06/11
Expires:
Last Modified: 2008/12/04 15:45:44 GMT

```

Figure 1. Data Set Information Panel - KSDS.

### Menu Bar Items:

The following menu bar items may be displayed depending on the type and organisation of the file.

- File** The File drop-down menu contains the single item "Exit" which simply closes the panel window.
- Report** Display all information obtained for the file in a single formatted report. This report is generated using hyper-text markup language (HTML) and is displayed in SELCOPY/i's HTML browser. The browser includes the menu item "Text" which will

display the report in plain EBCDIC text in a text edit view. This text output may subsequently be saved to DASD and printed.

**IDCAMS**

Opens the "Execute IDCAMS" utility window to generate an LISTCAT report for the file.

**RecordCount**

Executes a foreground pass of the file to provide an accurate account of the number of records, the lengths of the longest and shortest records and the average record length.

## Define GDG Base panel

SELCOPY/i 3.20 introduces a new IDCAMS DEFINE GDG Base panel. This may be opened using any of the following methods:

1. Select option 7. 'GDG' from the Create New Datasets menu panel. (=8.15.7)
2. Select 'Define GDG' from the File menu in the text edit main window menu bar.
3. Enter line command AMSDIALOG GDG (synonym AMSG) on the command line of any window.

```

SELCOPY/i - Define GDG Base
File Help JCL Command
Command>
ZZSGAMSG
WS WR
Scroll> Csr
Lines 1-20 of 21

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

GDG Name> NBJ.DUMPDS.GDG (required)
Model> NBJ.GDG01

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

Catalog> USERCAT.CBLCAT Catalog or Alias.

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

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

```

Figure 2. Define GDG Base.

## List VTOC Extents DSN filter

In addition to the volume id specification, the SELCOPY/i list of VTOC extents now supports specification of a DSN filter.

The DSN filter may contain wild card characters "\*" and "%" representing multiple and single characters as defined in the help for "Data Set Details". The DSN filter may be specified using either of the following methods:

1. As an optional parameter on primary command **LX**. e.g.  

```
LX CBLM08 NBJ.*
```
2. As an entry in the `Filter>` field of the VTOC Extent List panel.

```

SELCOPY/i - VTOC Extent List: cblm08
View Refresh Back Forward FDB Text Help
Command>
Volume> cblm08
Filter> nbj.*

```

-Vol--	-CC--	-HH--	Seq	Dsn	Or
CBLM08	11	1	1	NBJ.CC.INCLUDES	PS
CBLM08	11	8	1	NBJ.CC.PLK.MAP	PS
CBLM08	12	5	3	NBJ.CC.PLK.MAP	PS
CBLM08	13	2	1	NBJ.SQ11749.PDS1	PO
CBLM08	13	5	1	NBJ.SQ11749.PDS2	PO
CBLM08	13	11	1	NBJ.SQ11749.CTL	PS
CBLM08	14	0	1	NBJ.CTL	PO
CBLM08	22	3	1	NBJ.DATA.TEST	PS
CBLM08	23	9	1	NBJ.SELCOPY.NAM.FIXED	PS
CBLM08	23	10	1	NBJ.SELCOPY.NAM.VAR	PS
CBLM08	27	5	1	NBJ.CBLFTPZ.CMD	PS
CBLM08	33	0	1	NBJ.CUNJIMAP.CUNIMG00.ACTIVE	PS
CBLM08	34	0	1	NBJ.FSU.D2012059.T103405.SDO	PS
CBLM08	34	4	1	NBJ.INST.CBL13283.NBJ2.SELCOPYI.CMX	PS
CBLM08	35	5	1	NBJ.CBLINST.CBL12075.AZZSPENG	PO
CBLM08	36	0	1	NBJ.ZZSX.CBLA.D2012031.T1001199.AUD	PS
CBLM08	37	0	1	NBJ.ZZSX.CBLA.D2012030.T1703230.AUD	PS

```

Line 1 of 125 | Col 1 of 113 | Views 1 | select * sort Vol,CC,HH

```

Figure 3. VTOC Extent List.

### Allocate Non-VSAM Expiry

The SELCOPY/i Allocate NonVSAM panel now includes entry fields for data set retention period and expiry date. Use either of these input fields to assign a data set expiry after which it will be eligible for overwrite.

Any attempt to ERASE an unexpired file now produces a warning message which prompts the user to choose whether or not to continue the erase operation with PURGE.

```

SELCOPY/i - Allocate NonVSAM
Define Help
Dataset> NBJ.TEST.SEQ
Model>
Volume> CBLM12
MaxVol> 1
Unit> 3390
Type> SEQ (SEQ|PDS|PDSE)
Retention period> 7 (0-9999)
Expiry date> (yyyyddd)

```

Space Allocation	DCB Information	SMS Classes
Allocation Unit> BLOCKS	Organisation> PS	Data>
Primary> 120	Record Format> VB	Storage> CBLDFLT
Secondary> 50	Record Length> 255	Management> CBLDFLT
Directory Blocks> 0	Blocksize> 27998	

```

Allocate Cancel Help

```

Figure 4. Allocate NonVSAM.

# SELCOPY/i Print Utility

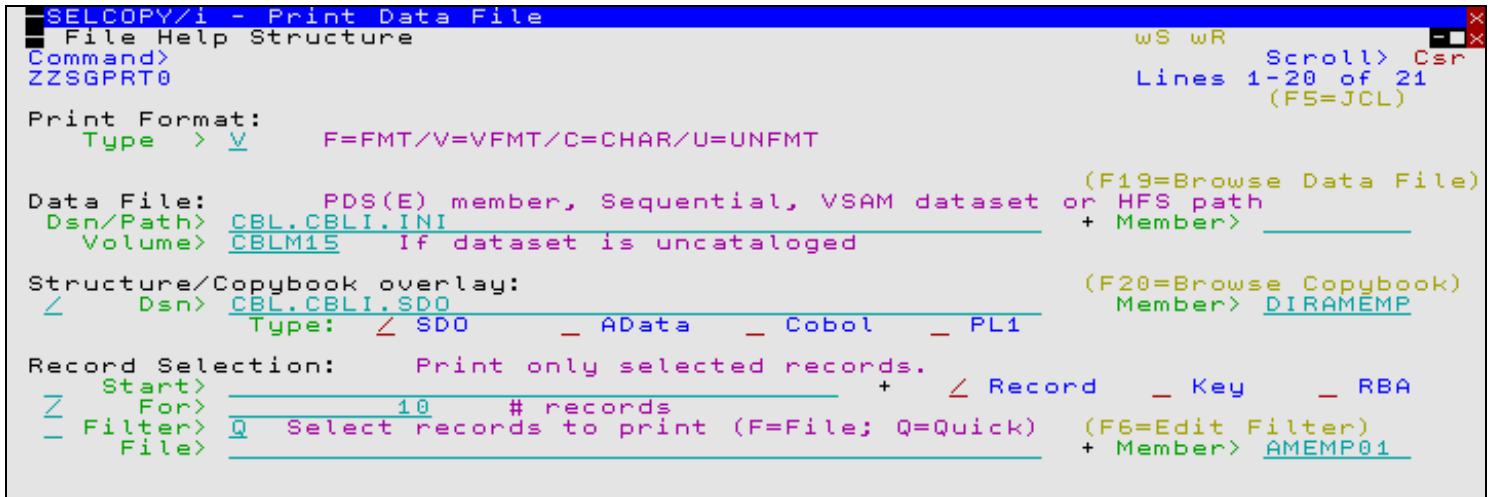
First introduced in PTF RS00019 to SELCOPY/i 3.10, the Print utility has been included to write the contents of a data file to print output. The printed file data may be first formatted using a structure (SDO, COBOL or PL1 copybook).

## Print Utility Panel

The Print Data File panel is used to generate a batch (SDEAMAIN) job to browse a file and execute the PRINT primary command to print the file data. The panel may be opened by selecting option 11. (=11) from the primary options menu.

This panel consists of a sequence of panel views used to identify the following:

- The print type (single or multiple record view, formatted or unformatted).
- The name of the file to be printed.
- The name and type of structure to be used to format the file data.
- A temporary (quick) filter or the name of a filter file to be used to select records for print.
- Options specific to the type of print selected.
- Print output options.



```
SELCOPY/i - Print Data File
File Help Structure
Command>
ZZSGPRT0
Print Format:
Type > V F=FMT/V=VFMT/C=CHAR/U=UNFMT
Data File: PDS(E) member, Sequential, VSAM dataset or HFS path (F19=Browse Data File)
Dsn/Path> CBL.CBLI.INI + Member>
Volume> CBLM15 If dataset is uncataloged
Structure/Copybook overlay: (F20=Browse Copybook)
Dsn> CBL.CBLI.SDO Member> DIRAMEMP
Type: / SDO _ AData _ Cobol _ PL1
Record Selection: Print only selected records.
Start> + / Record _ Key _ RBA
For> 10 # records
Filter> Q Select records to print (F=File; Q=Quick) (F6=Edit Filter)
File> + Member> AMEMP01
```

Figure 5. Print Data File.

## Data Edit PRINT Primary Command

The PRINT primary command may be executed from any Structured Data Edit view of data which includes DB2 table data, SQLEXEC, FSU and COMPFIL utility report output.

### Syntax:

```

      +- Record +-      +- Limit ----- 100 --- Pages ----+
      |         |      |         |                       |
>>- PRINT -----+-----+-----+-----+-----+-----+-----+----->
      |         |      |         |                       |
      +- Page ----+      |         |           +- Pages +- |
      |         |      |         |           |         | |
      +- File ----+      +- Stopafter -+--- n ----+-----+-----+
      |         |      |         |           |         | |
      |         |      +- Limit -----+      +- LINES +- |
      |         |      |         |           |         | |
      |         |      +- Nolimit -----+-----+-----+
      |         |      |         |           |         | |
      +- NOTRUNCate -----+      +- Acrossthendown -+      +- Browse +-
      |         |      |         |           |         | |
>-----+-----+-----+-----+-----+-----+-----+----->
      |         |      +- Downthenacross -+      +- Edit ----+
      |         |      |         |           |         | |
      +- TRUNCate -----+-----+-----+-----+-----+
      |         |      |         |           |         | |
      +- Outdsn SDEPRINT -+      |         |           +- Copies 1 -----+
      |         |      |         |           |         | |
>-----+-----+-----+-----+-----+-----+-----+----->
      |         |      +- Outdsn fileid ----+      +- SYSOut outputclass -+      +- Copies n_copies -+
      |         |      |         |           |         | |
      |         |      |         |           |         | |
      |         |      |         |           |         | |
      (1)
      +- OLD -----+
      |         |
>-----+-----+-----+-----+-----+-----+-----+-----<
      |         |      +- SHr -----+      +- PAGEWidth -+- n_cols -+      +- PAGEDepth -+- n_lines +
      +- SHare --+      +- PW -----+      +- PD -----+
      |         |
      +- MODify -+
      +- APPEND -+

```

Note: 1. Default disposition is SHR if OUTDSN is a PDS/PDSE library member.

### Description:

The PRINT command may be used to print the current line view of file data, all data within the current line in single-record view, or all data within all displayed lines starting at the current line.

PRINT may be executed interactively or in batch as SDEIN input to the SELCOPY/i batch program (SDEAMAIN). In both environments, PRINT may be specified following an EDIT or BROWSE command and any display formatting commands to print the contents of the current SDE window view.

Whether or nor output is to a SYSOUT data set, the dimensions of the printed SDE view are not determined by the actual SDE view display, but correspond to the page width and page depth values.

The print output page depth is defined as the value set by the PAGEDEPTH parameter or else the value set by the **PAGEDEPTH** option (default 60 lines). The page depth value includes the 5 Print header lines so that the number of lines of data printed will be 5 less than the PAGEDEPTH value.

The print output page width is defined as the value set by the PAGEWIDTH parameter. If PAGEWIDTH is not specified then, for output to SYSOUT data sets and HFS files only, the output page width is defined by the value set by the **PAGEWIDTH** option (default 133 columns). Otherwise, the output page width is equal to the file's maximum record length value. Note that, for print of records in a single record view, page width is limited to a maximum of 255 columns.

# SELCOPY/i FSU and FCOPY Utility Updates

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

## Panel Redesign

FSU and FCOPY utility panels have been enhanced in particular to allow the following:

- Selection of input files and/or library members from a supplied fileid mask.
- Optional specification of a temporary (quick) filter or the name of a filter file to be used to select records for processing.
- Browse of input file selected for file copy, with or without an accompanying structure.
- Edit of the input and/or output COBOL or PL1 copy book to be used for file remap.
- Reload of the input and/or output structure generated by an updated COBOL or PL1 copy book.
- 

## File Search, Update, Copy & Remap (=6)

The FSU panel is now a sequence of panel views, the first of which is for a basic file search. Only if the Extended File Search/Update/Copy/Remap Tasks option is set will the utility guide the user through a sequence of panels to perform more advanced tasks.

Primary command SELECT (no parameters) is supported by this panel view to select individual files and/or library members that satisfy the supplied input fileid mask.

Primary command FILTER (no parameters) is supported to define a temporary (quick) filter or to view/update an existing filter file depending on the filter type specified.

Figure 6. Basic File Search Panel.

If the Extended File Search/Update/Copy/Remap Tasks option is set, a menu of FSU utility functions, arranged by task, will be displayed. The subsequent sequence of displayed panel views will depend on which of the tasks has been selected.

```

SELCOPY/i - Extended File Search, Update, Copy & Remap Tasks
File Help                               wS wR
Command>                               Scroll> Csr
ZZSGFSU0                               Lines 1-20 of 21

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

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

```

Figure 7. FSU: Extended File Search, Update, Copy &amp; Remap Tasks Panel.

## File Copy/Reformat (=5)

The FCOPY panel now supports primary commands SELECT and FILTER as described for the FSU panel above. If the input data set is a PDS or PDSE library, then, so that the user is given the opportunity to select required members, the SELECT function is performed automatically prior to performing the copy.

```

SELCOPY/i - File Copy
File Help JCL Command                   wS wR
Command>                               Scroll> Csr
ZZSGFCO0                               Lines 1-20 of 21

Input PDS/PDSE Library, Sequential, VSAM DSN mask or HFS path mask:
  DSN/Path Mask> _____ +
  Member Mask> _____ + (All mask matches will be copied,
  Volume Mask> _____ + press PF5 to view/deselect matches)

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

Record Selection: For each input file, copy only selected records.
- Start> _____ + / Record - Key - RBA
- For> _____ # records
- Filter> Q Select records to copy. (F=File; Q=Quick) PF6=Edit Filter
- File> _____ + Member> _____

Options:
- Reformat using structure/copybook layouts _ Recurse HFS Sub-directories
- Append to existing Output _ Ignore HFS fileid case

```

Figure 8. File Copy Panel.

The FCOPY panel now also has an option "Reformat using structure/copybook layouts" which, if selected, will open a new panel view allowing the user to select a structure file name and type from a filtered list of files for both the source and destination structures. The selected structures will be used to format the input and output record data as it is copied.

```

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

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

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

```

Figure 9. File Reformat Panel View.

### FIND Condense Member list panel

For both the FSU and FCOPY panels, having used SELECT to display a list of PDS or PDSE library members, the primary command FIND (no parameters) may be executed to open the Find/Condense input members sub-panel. This panel may be used to condense the list of displayed members based on their satisfying the generated FIND operation.

On each execution of the FIND panel operation, an FSU FIND operation is performed to identify, and so select, only those members with records containing at least one hit for the search string. New FIND criteria may be entered and executed repeatedly to further condense the list of members.

```

SELCOPY/i - Select Input Members - FIND
File Help
Command>
ZZSGFSU9
Find string ==> c'IQ003727'
Limit ==> 1 Max number of hits per file/member (0=>All)
Scope ==> WORD (CHARS, WORD, PREFIX, SUFFIX)
Start column ==> 0 (0=>All columns)
End column ==> 0 (0=>Start column only)
Start record ==> 1
Number of records ==> 0 (0=>All)
Relational operator ==> EQ (EQ, NE, GT, GE, LT, LE)
View report ==> Yes Display FSU FIND report output
Condense member list ==> Yes Include only members containing hit(s)

```

Figure 10. File Copy Panel.



## Primary Command Updates

Both the FSU and FCOPY primary commands have been updated as follows:

### **REPORT default**

FSU report output is written to a temporary, in-storage data set when run interactively. However, prior to SELCOPY/i 3.20, specification of an existing report output data set was mandatory when executed in batch using SDEAMAIN.

SELCOPY/i 3.20 removes this restriction so that, if no report data set is specified, a new data set and accompanying structure (SDO) will be allocated with a generated DSN based on the current timestamp and the user's defined user prefix. Similarly, if an output report fileid is specified for a data set that does not already exist, then it will be automatically allocated assuming SMS automatic class selection and with a fixed space specification.

### **LIMIT**

FSU now supports parameter LIMIT which specifies a maximum number of successful search string hits that will be reported for an individual file selected by the fileid mask. Once this limit has been reached, then no further data will be read from the input file, the file is closed and processing continues with the next file that satisfies the input fileid mask.

### **KSDSINSERT**

Both FSU and FCOPY primary commands now support parameter KSDSINSERT which allows copying of records that are not in key sequence to a KSDS data set.

### **OVERWRITE**

Both FSU and FCOPY primary commands now support parameter OVERWRITE to allow a file that satisfies the input fileid mask to be overwritten if specified as the output file. This may only occur if the overwritten input file is the first or only file selected for processing.

By default, FSU and FCOPY have been updated to prevent accidental overwrite of any input file.

### **Regular Expressions**

The FSU utility uses the data edit FIND and CHANGE command syntax to search and update record data. Since these commands both now support regular expressions, the FSU command syntax also supports specification of regular expressions as arguments on both the FIND and CHANGE parameters.

### **FIND & CHANGE Operators**

Data edit FIND and CHANGE command syntax also now support specification of an operator so supporting searches for values or character strings that satisfy an inequality. Because FSU syntax utilises data edit FIND and CHANGE functionality, operators are also supported in the arguments of the FSU FIND and CHANGE parameters.

# SELCOPY/i Text Editor

---

SELCOPY/i 3.20 includes a number of enhancements to the SELCOPY/i text editor.

## Processing Enhancements

The following new features have been introduced in SELCOPY/i 3.20 and relate to the display of text and operation of a text edit view.

### ISPF Edit Rexx Macro support

As already reported under heading **ISPF Edit Rexx Macros**, the SELCOPY/i text editor will now attempt to execute ISPF Edit macros that have been written in the Rexx programming language and exist in the SYSUEXEC or SYSEXEC library concatenations.

Support of ISPF Edit Rexx macros first requires activation via the Text Edit Settings panel (=0.3) by setting the ISR Macros option to "YES".

### Environment Variable %MCAT%

SELCOPY/i 3.20 introduces the new text edit standard environment variable MCAT which corresponds to the DSN of the local z/OS host's ICF master catalog. This environment variable may be referenced in any command executed via the ACTION key in a text edit view, or within a SELCOPY/i text edit macro. e.g. To open a catalog list displaying only alias entries...

```
LISTCAT * %MCAT% X
```

### CMDTEXT Command Execution

The CMDTEXT primary command has now been assigned the preferred synonym ACTION. In due course, all specification of CMDTEXT in SELCOPY/i documentation and key lists will be updated to be ACTION instead. This is particularly true in reference to the ACTION PFkey.

The ACTION facility, when assigned to a PFKey, enables the user to point-and-shoot at command syntax that has been entered within the text of an edited file and so execute the command without having to type it at a command prompt.

Certain characters that may occur within the line of data being processed, have special meaning to the ACTION functionality. These characters are "|" (OR symbol - X'4F'), "<" (less than - X'4C'), ">" (greater than - X'6E'), "\_" (underscore - X'6D'), "'" (reverse apostrophe - X'79') and "\" (backslash - X'E0').

For **new** users only (i.e. users whose SELCOPY/i User INI has not yet been allocated), special processing caused by the following special characters will be disabled by default.

"|" (OR symbol - X'4F')

The separator character used to delimit the end of line text to be processed by the ACTION facility. Any text, including another command or comment data, may follow a "|" character and it will not be processed with the text on which the cursor is positioned.

Therefore, for a line of text that includes "|" as part of the command syntax, the special significance of "|" would have to be suppressed by adding "||" (2 OR symbols) at the start of the line of text but following the first ">" or "<" symbol.

This effect of the "|" character may be re-enabled by setting option ACTION Delim to be "ON" in the Text Edit Settings panel (=0.3). Alternatively, by executing the new text edit primary command SET ACTIONDELIM ON.

"\_" (underscore - X'6D')

Indicates that, when ACTION parameter EDIT or EDITALL is used or if the line of text begins with the ">" symbol, the cursor will be placed at the location of the first "\_" within the text when the text is copied to the command line. Regardless of whether these conditions are true, the first occurrence of an underscore within the text will always be removed from the text.

Therefore, for a line of text that includes "\_" as part of the command syntax, an additional "\_" would have to be introduced before the first occurrence of "\_" in the command.

This effect of the "\_" character may be re-enabled by setting option ACTION Cursor to be "ON" in the Text Edit Settings panel (=0.3). Alternatively, by executing the new text edit primary command SET ACTIONCURSOR ON.

## Syntax Colour Highlighting

In release of SELCOPY/i prior to 3.20, rudimentary syntax colour highlighting was performed for edited files using the SET SCOLOR primary command.

In SELCOPY/i 3.20, colour highlighting of text displayed in a text editor view has been implemented correctly so that syntax highlighting is supported for the following programming languages:

<b>ASSEMBLER</b>	<b>JCL</b>
<b>C</b>	<b>PL1</b>
<b>CMX</b>	<b>REXX</b>
<b>COBOL</b>	<b>SELCOPY</b>
<b>HTML</b>	<b>XML</b>

Syntax highlighting is controlled using the SET HILIGHT (or HILITE) and SET HCOLOUR (or HCOLOR) commands.

## Primary Commands

The following text editor primary commands have been introduced in release 3.20.

---

### GO

---

#### Syntax:

```
>>-- GO -----+-- Browse ----><
|               |               |
+-- Edit -----+               |
|               |               |
+-- SE -----+               |
|               |               |
+-- SU -----+               |
|               |               |
+-- View -----+               |
```

#### Description:

Closes the current view(s) of the file data and redisplay it in either a text edit or data edit view. The options are as follow:

GO SE	Switches to Data-Edit (Full-Edit)
GO SU	Switches to Data-Edit (Upd-in-Place)
GO Browse	Switches to Data-Edit (Browse)
GO Edit	Switches to Text-Edit
GO View	Switches to Text-Edit (View)

---

### NOND

---

#### Syntax:

```
>>-- NOND -----><
```

#### Description:

NOND is supplied as an edit macro. NOND exploits the new NONDISPLAY colour setting to toggle the display of underscores on printable text in lines of data which include unprintable characters.

---

## RUNSLC

---

**Syntax:**

```
>>-- RUNSLC -----><
```

**Description:**

RUNSLC allows execution of the program SLC, the C++ version of SELCOPY, in the foreground using control statements in the focus edit view as the SYSIN input. SYSPRINT output is captured and displayed in a separate edit view.

---

## SET/QUERY/EXTRACT Options

The following text editor SET, QUERY and EXTRACT options have been introduced in release 3.20.

---

## ACTIONCURSOR

---

**Syntax:**

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

```
>>--- Query ----- ACTIONCURSOR -----><
```

```
>>--- EXtract --- /ACTIONCURSOR/ -----><
```

**Description:**

Enable and disable the significance of the first occurrence of special character "\_" (underscore - X'6D') in a line of text processed by the ACTION facility. SET ACTIONCURSOR takes effect at the Global level.

---

## ACTIONDELIM

---

**Syntax:**

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

```
>>--- Query ----- ACTIONDELIM -----><
```

```
>>--- EXtract --- /ACTIONDELIM/ -----><
```

**Description:**

Enable and disable the significance of special character "|" (OR symbol - X'4F') in a line of text processed by the ACTION facility. SET ACTIONDELIM takes effect at the Global level.

## COLOUR (COLOR)

### Syntax:

```

>>+-----+---+ COLOr  ---+--- NONDisplay  ---+ Blue  -----+---+-----+---><
    |         |   |         |         |         |         |         |         |         |
+- SET  -+  +- COLOur  -+         +- Red  -----+--- BLInk  -----+
    |         |         |         |         |         |         |         |         |
+- Pink  -----+--- REVerse  ---+
    |         |         |         |         |         |         |         |         |
+- Green  -----+--- UScore  ---+
    |         |         |         |         |         |         |         |         |
+- Turquoise  -+
    |         |         |         |         |         |         |         |         |
+- Yellow  -----+
    |         |         |         |         |         |         |         |         |
+- White  -----+
    |         |         |         |         |         |         |         |         |
+- Default  ---+
  
```

### Description:

NONDISPLAY option has been included as a parameter to the SET COLOUR (COLOR) option to control the appearance of printable characters in a line of text containing unprintable characters. By default, these characters will be displayed as Turquoise Uscore.

QUERY and EXTRACT COLOUR (COLOR) include an entry for the current NONDISPLAY setting.

## HCOLOUR (HCOLOR)

### Syntax:

```

>>+-----+---+ HCOLOr  ---+--- DEFault  -----+--- Blue  -----+---+-----+---><
    |         |   |         |         |         |         |         |         |         |
+- SET  -+  +- HCOLour  -+  +- COMment  -----+--- Red  -----+--- BLInk  -----+
    |         |         |         |         |         |         |         |         |
+- ERRor  -----+--- Pink  -----+--- REVvideo  +
    |         |         |         |         |         |         |         |         |
+- KEYword  -----+--- Green  -----+--- UScore  ---+
    |         |         |         |         |         |         |         |         |
+- NUMber  -----+--- Turquoise  +
    |         |         |         |         |         |         |         |         |
+- OPERator  -----+--- Yellow  -----+
    |         |         |         |         |         |         |         |         |
+- PARentthesis  -+--- White  -----+
    |         |         |         |         |         |         |         |         |
+- STRing  -----+--- Default  ---+
    |         |         |         |         |         |         |         |         |
+- VARiable  -----+

>>--- Query  ----- HCOLour  -----><

>>--- EXtract  --- /HCOLour/  -----><
  
```

### Description:

Assign colour to individual syntax elements when syntax colouring is set on by the SET HILITE option. SET HCOLOUR takes effect at the File level.

---

**HIGHLIGHT (HILITE)**


---

**Syntax:**

```
>>+-----+----- Hilite --+--+ OFF -----><
   |         |         |         |         |
   +- SET -+ -+ HIligh -+ -+ ON ---+ ASSEmbl er -+--+
                                   |
                                   +- C -----+
                                   |
                                   +- CMX -----+
                                   |
                                   +- COBo1 -----+
                                   |
                                   +- HTM1 -----+
                                   |
                                   +- JCL -----+
                                   |
                                   +- PL1 -----+
                                   +- PLI -----+
                                   |
                                   +- REXx -----+
                                   |
                                   +- SELCopy ----+
                                   |
                                   +- XML -----+

>>--- Query ----- HIlite -----><

>>--- EXTRACT --- /HIlite/ -----><
```

**Description:**

Controls text edit syntax colour highlighting which is sensitive to the language or format of the edited source. Selected colours of recognised syntax elements may be altered using **SET HCOLOUR**. SET HILITE takes effect at the File level.

---

**INIVAR**


---

**Syntax:**

```
>>-- UNSET ----- INIVar ----- var_name -----><
```

**Description:**

SET INIVAR adds or updates a User INI file variable in storage. This may be one of SELCOPY/i's standard INI variables or one that is user defined. SET INIVAR *var\_name* set the variable to Null. UNSET INIVAR has been introduced to remove a User INI file variable from storage altogether.

# SELCOPY/i Structured Data Editor

SELCOPY/i 3.20 includes a number of enhancements to the SELCOPY/i structured data editor.

## Panels

The following Structured Data Edit panels have been introduced or enhanced in SELCOPY/i 3.20.

### Data Browse/Edit (=2)

The Structured Data Browse/Edit utility panel has been updated to select the type of edit to be performed on the specified file and also to include optional specification of a temporary (quick) filter or the name of a filter file to be used to select records for the browse/edit operation. File object name fields also support wild card entries allowing file selection from a filtered list of entries that satisfy the fileid mask.

An additional options view has also been included to select HFS file format and to specify the use and name of a profile macro to be executed on initial display of the data.

```
SELCOPY/i - Structured Data Browse/Edit
File Command Structure Replace Help
Command> ZZSGSDE0
PDS/PDSE member, Sequential, VSAM or HFS path:
  Name> NBJ.SELCTRN.ZZST1DAT
  Volume> _____ If dataset is uncataloged.
Action:
  Browse Data.
  Edit Full. (Insert/Update/Delete)
  Edit In-Place. (Update only)
Structure/Copybook overlay:
  Dsn> NBJ.SELCTRN.SAM1
  Type: SDO
Record Selection:
  Start> _____
  For> _____ # records
  Filter> Q Filter selected records. (F=File; Q=Quick)
  File> _____
Additional Options: _____ (Enter "/" to display HFS and Profile options.)
```

Figure 11. Structured Data Browse/Edit Panel.

### Create STRUCTURE from COBOL/PL1 copybook(s) (=9.1)

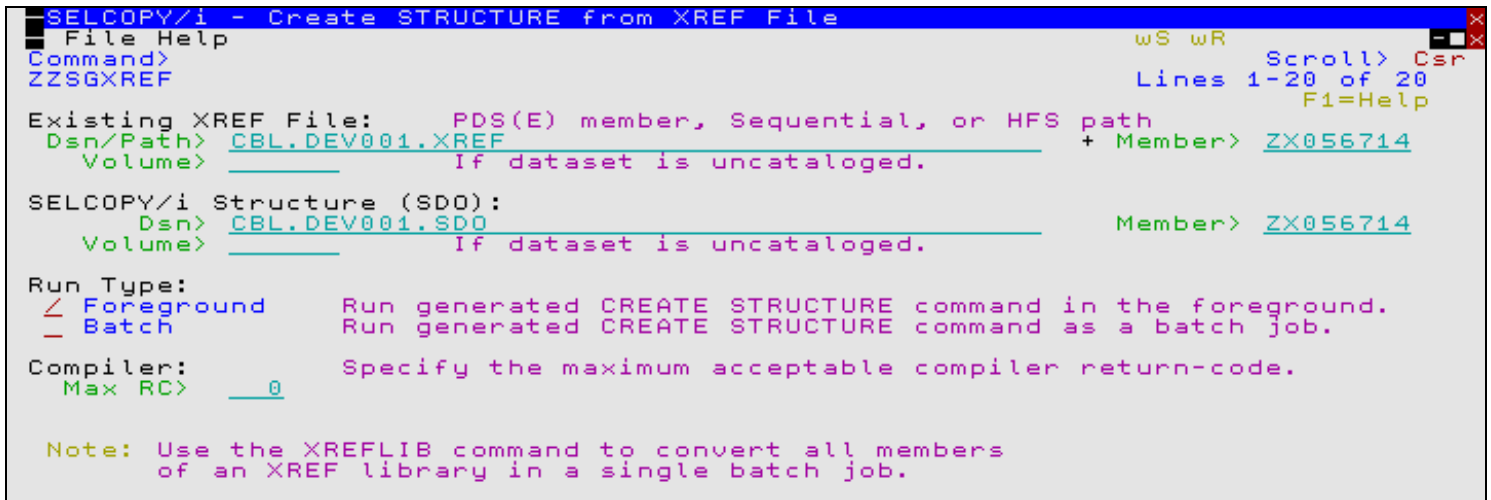
The Create STRUCTURE from COBOL/PL1 copybook(s) utility panel has been introduced to assist users with generation of SELCOPY/i structures from one or more COBOL or PL1 copy book, COBOL or PL1 ADATA file or SELCOPY/i SDO structure.

```
SELCOPY/i - Create STRUCTURE from COBOL/PL1 copybook(s)
File Help
Command> ZZSGSD01
1 Library Specify source copybook libraries
2 Record-type Add/Delete record-types from COBOL/PL1 copybooks
3 Replace COBOL Replacing options
4 Create Create Structure (SDO) in the foreground
5 Batch Create Batch Job
Structure File to Create/Edit:
  Dsn> NBJ.NEW.SDO
  Member> CBLDIST2
Title > Jameson Ltd Accounts
Description> Sales and Purchase Ledger Formatting for Jameson Ford Trading Limited
```

Figure 12. Create STRUCTURE from COBOL/PL1 copybook(s) Panel.

## Create STRUCTURE from XREF File (=9.2)

The Create STRUCTURE from XREF File panel has been introduced to provide a migration tool to convert FileAID XREF library members to SELCOPY/i SDO structure format.



```

SELCOPY/i - Create STRUCTURE from XREF File
File Help
Command>
ZZSGXREF
Existing XREF File: PDS(E) member, Sequential, or HFS path
Dsn/Path> CBL.DEV001.XREF + Member> ZX056714
Volume> _____ If dataset is uncataloged.
SELCOPY/i Structure (SDO):
Dsn> CBL.DEV001.SDO Member> ZX056714
Volume> _____ If dataset is uncataloged.
Run Type:
 / Foreground Run generated CREATE STRUCTURE command in the foreground.
 - Batch Run generated CREATE STRUCTURE command as a batch job.
Compiler:
Max RC> 0 Specify the maximum acceptable compiler return-code.
Note: Use the XREFLIB command to convert all members
of an XREF library in a single batch job.
  
```

Figure 13. Create STRUCTURE from XREF File Panel.

## Display record layouts defined by an SDO/copybook (=9.3)

The Display record layouts defined by an SDO/copybook panel has been introduced to allow display the layout of record mappings defined in a SELCOPY/i SDO structure, a COBOL or PL1 copy book or in a COBOL or PL1 ADATA file.

Figure 14. Display record layouts Panel.

## Processing Enhancements

The following new features have been introduced in SELCOPY/i 3.20 and relate to the interpretation and display of data, and operation of a data edit view.

### COBOL PICTURE symbol "P"

SELCOPY/i now correctly interprets and displays a COBOL copy book formatted field that is defined by a PICTURE string which includes the "P" character to represent an assumed decimal scaling position. Specifically, the "P" character is used to specify the location of an assumed decimal point when the point is not within the number that appears in the data item.



In previous releases, error ZZSD022E "Construction of data element *filename* of data type DECIMAL" would be returned when SELCOPY/i attempted to generate the SDO structure.

## PL1 PICTURE symbols "T", "I" and "R"

SELCOPY/i now correctly interprets a PL1 copy book formatted field that is defined by a PICTURE string which includes one of the picture string characters "T", "I" or "R" in the junior byte position so representing an overpunch digit and sign (i.e. fixed zoned decimal value).

In previous releases, error ZZSD323E "Fixed decimal picture *picture* has one or more characters for overpunch (T I R), zero replacement (Y) or scaling factor (F) which are not supported in this release of structured edit." would be returned when SELCOPY/i attempted to generate the SDO structure.

## Escaped Quotes

Character literal strings may be specified in the following contexts:

- As a search string in a FIND, CHANGE, EXCLUDE and ONLY command.
- As a replace string in a CHANGE command.
- As a string in an SDE expression. Commands that use SDE expressions are WHERE, LOCATE, MORE, LESS, USE WHEN and CREATE STRUCTURE.

Character literal strings must be enclosed by a delimiter character which may be either a quote (") or apostrophe ('). Where a character literal string includes a quote (") or apostrophe (') as part of the string data, then the alternative delimiter character may be used to enclose the string. However, if the string contains both characters as part of the data, then, in previous releases, this could not be represented.

SELCOPY/i 3.20 supports escaping of the delimiter character being used as data within a literal string enclosed by the same delimiting character. The escape character is the same character as the string delimiter. e.g.

```
FIND C'O'Brien'
```

## COBOL Level 88 condition support

COBOL copy books may contain level-88 data items to define a condition-name entry that is associated with a particular value for a conditional variable.

SELCOPY/i 3.20 supports use of a level-88 condition name in an SDE expression to test the associated condition. e.g. A COBOL copy book, processed by SELCOPY/i to format data, may contain the following:

```
03 RECORD-TYP
05 RT          PIC X(001).
   88 RT-ARTIST VALUE '1'.
   88 RT-ALBUM  VALUE '2'.
   88 RT-TRACK  VALUE '3'.
   88 RT-ANY    VALUE '1' THRU '3'.
   88 RT-NONE   VALUE '4' THRU '9'.
```

Data formatted using this copy book may then use the condition names specified by the 88 levels to test the contents of the preceding, lower level RT field. e.g. To display only records which have RT = '2', the following primary command may be used:

```
WHERE RT-ALBUM
```

## Browse OCCURS DEPENDING array

As a default, browse of a record in formatted, single record view (MAP) no longer displays null entries of an array (COBOL OCCURS DEPENDING).

SELCOPY/i 3.20 introduces support for the SET/QUERY/EXTRACT ARRAYALL option to control whether or not all entries of an array are displayed when browses in formatted, single record view.

## Primary Commands

The following structured data editor primary commands have been enhanced or introduced in release 3.20.

### ARRC

**Syntax:**

```
>>-- ARRC -----><
      |           |
      +- field_col -+
```

**Description:**

ARRC collapses the specified field array of single characters to display it as a single, variable length character (VARCHAR) field. ARRC is an abbreviation of SET ARRAYASCHAR ON. If *field\_col* is not specified, then the focus field (at the cursor position) is assumed.

### ARRX

**Syntax:**

```
>>-- ARRX -----><
      |           |
      +- field_col -+
```

**Description:**

Where a field array of single characters is displayed as a variable length character (VARCHAR) field, ARRXX expands the specified field to redisplay it as an array of single characters. ARRXX is an abbreviation of SET ARRAYASCHAR OFF. If *field\_col* is not specified, then the focus field (at the cursor position) is assumed.

### CHANGE

**Syntax:**

```
>>--- Change  +- EQ +-
              |   |
              +- op +-
              +- VALID -----+
              +- INVALID -----+

>----- | CHANGE parameters | -----><
```

**Description:**

CHANGE primary command has been enhanced to accomodate the following:

1. Support for regular expressions on both the search and replace strings as described in [Regular Expressions](#) above.
2. Support for a relational operator *op* in the compare operation. This determines the relationship that the data must have with the search string in order for it to be identified as a successful match.

If a character string compare is performed, the EBCDIC values assigned to characters in the search and data strings determine the relationship (equal to, greater than or less than) between the two strings.

Valid values are:

Operator	Description
EQ	Data must be equal to <i>string1</i> . (Default)
NE	Data must be not equal to <i>string1</i> .
GT	Data must be greater than <i>string1</i> .



## CREATE

### Syntax:

```
>>-- CREate -----+-----+-----+-----+----->>
      |               |               |               |
      +- fileid -+    +- .name1 -- .name2 --+
                        (1)
```

### Note:

1. If a group of lines are not specified using line label names, then a group of lines must be specified using one of the following line (prefix area) commands:

- ◆ **C** or **CC** (Copy), if the group of lines in the current file is to be preserved following successful execution of CREATE.
- ◆ **M** or **MM** (Move), if the group of lines in the current file is to be deleted following successful execution of CREATE.

### Description:

Create a new sequential or VSAM data set, PDS/PDSE library member or HFS file, containing a group of lines extracted from the current SDE edit or browse view.

If the target file already exists, then the following message is returned:

```
ZZSD229E File fileid already exists.
```

If output is to a new member of an existing PDS/PDSE library or to an HFS file, the target file will be created automatically created. When CREATE defines a new HFS file, the permission bits are set to 740 (rwx-----).

If output is to a non-existent data set or to a member of a non-existent PDS/PDSE library, the **Allocate Non-VSAM** panel is displayed in order to allocate the new data set.

Beware that the specified *fileid* character string must not be "LIST" (minimum abbreviation LI), "STRUCTURE" (minimum abbreviation "STRUCT") or "VIEW" (minimum abbreviation VI) which conflict with SDE command verbs CREATE LIST, CREATE STRUCTURE and CREATE VIEW.

Specify CREATE with no parameters to open the SDE CREATE/REPLACE file panel.

## CREATE STRUCTURE

### Syntax:

```
>>-- CREATE STRUCTure --- struct_name ----->>
>+-----+-----+-----+-----+----->
  |               |               |               |
  +- TITLE short_title -+ +- DESCRIPTION long_description ---+
>+-----+-----+-----+-----+----->
  |               | XRef Definition |               |
  +-----+-----+-----+-----+-----+
  |               | CopyBook Definition |               |
  +-----+-----+-----+-----+-----+
  |               | DB2 Definition |               |
  +-----+-----+-----+-----+-----+
  |               | Direct Definition |               |
  +-----+-----+-----+-----+-----+
  +- NOREplace -+                +- CASE (IGNORE) --+
  |               |               |               |
  +-----+-----+-----+-----+----->>
  |               |               |               |
  +--- REPlace ---+ +--- TEMPorary ---+ +--- CASE (RESPECT) -+
  |               |               |               |
```

### XRef Definition:

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

**RecordType Definition:**

```

>-- NAME --- record_type ----->
      + COBo1 +
>-- SOURCE +++-----+-----+-----+-----+----- member_name ----->
      |   |   |   |   |   |
      | + PL1 --+ +- copybook_library(member_name) -+ |
      |   |   |   |   |
      | +-- ADAta -----+-----+-----+-----+-----+
      | +-- SDO -----+-----+-----+-----+-----+
      | +-- STRUcture (-- | Record Definition | --) ----+
      |   |   |   |
      | +-- PRImary ----+
      |   |   |   |
      +--+-----+-----+-----+-----+-----+
      | +-- SEcOndary ---+ +- USE -+ +- IF ----+ |
      |   |   |   |   |   |
      +-----+-----+-----+-----+
      |   |   |   |   |   |
      | +-- OFFset -+-----+-----+-----+-----+
      |   |   |   |   |   |
      | +-- (plus) -+
      |   |   |   |   |   |
      | +-- (minus) +

```

**Description:**

The CREATE STRUCTURE has been updated to support new syntax that closely resembles XREF definition syntax in FileAID.

This is now the preferred command syntax for SELCOPY/i SDO structure generation and is the basis of the Create STRUCTURE from COBOL/PL1 copybook(s) panel (=9.1). The new syntax supports creation of an SDO from a number source dataset of potentially different source formats including COBOL copy books, PL1 copy books, SELCOPY/i SDO structures and COBOL and PL1 ADATA files.

**EXCLUDE****Syntax:**

```

>>--+- EXclude -+-----+-----+-----+-----+----->
      |   |   |   |   |   |
      | + X -----+ | + op -+ | + NEXT --+ | + CHARs --+ |
      |   |   |   |   |   |
      | + VALID -----+ | + ALL ----+ | + PRefix -+ |
      | + INVALID -----+ | + FIRST -+ | + SUFFix -+ |
      |   |   |   |   |   |
      | +-----+-----+ | + LAST --+ | + WORD ---+ |
      |   |   |   |   |   |
      |   |   |   |   |   |
      | +-----+-----+ | + PREV --+ |

```

>----- | EXCLUDE parameters | -----><

**Description:**

EXCLUDE primary command has been enhanced to accomodate the same features as described for **CHANGE** above.

**FIND****Syntax:**

```

>>--+- Find -+-----+-----+-----+-----+----->
      |   |   |   |   |   |
      | + / -----+ | + op -+ | + NEXT --+ | + CHARs --+ |
      |   |   |   |   |   |
      | + VALID -----+ | + ALL ----+ | + PRefix -+ | + EX -+ |
      | + INVALID -----+ | + FIRST -+ | + SUFFix -+ | + NX -+ |
      |   |   |   |   |   |
      | +-----+-----+ | + LAST --+ | + WORD ---+ | + X --+ |
      |   |   |   |   |   |
      |   |   |   |   |   |
      | +-----+-----+ | + PREV --+ |

```

>----- | FIND parameters | -----><

**Description:**

FIND primary command has been enhanced to accomodate the same features as described for CHANGE above.

**GO**

**Syntax:**

```
>>-- GO -----+-- Browse ----+-----><
                |               |
                +-- Edit ----+
                |               |
                +-- SE -----+
                |               |
                +-- SU -----+
                |               |
                +-- View ----+
                |               |
```

**Description:**

Closes the current view(s) of the file data and redisplay it in either a text edit or data edit view. The options are as follow:

GO SE	Switches to Data-Edit (Full-Edit)
GO SU	Switches to Data-Edit (Upd-in-Place)
GO Browse	Switches to Data-Edit (Browse)
GO Edit	Switches to Text-Edit
GO View	Switches to Text-Edit (View)

**GRPC**

**Syntax:**

```
>>-- GRPC -----+-----><
                |               |
                +- field_col -+
```

**Description:**

GRPC collapses the specified group (structure) field to display it as a single, fixed length character (CHAR) field. GRPC is an abbreviation of SET GROUPASCHAR ON. If *field\_col* is not specified, then the focus field (at the cursor position) is assumed.

**GRPX**

**Syntax:**

```
>>-- GRPX -----+-----><
                |               |
                +- field_col -+
```

**Description:**

Where a group field is displayed as a fixed length character (CHAR) field, GRPX expands the specified field to redisplay it as individual group member fields. GRPX is an abbreviation of SET GROUPASCHAR OFF. If *field\_col* is not specified, then the focus field (at the cursor position) is assumed.

## HEXDUMP

### Syntax:

```
>>-- HEXDump -----><
      |-----|
      +--- New ----+
```

### Description:

Display the focus record or record segment in single record, unformatted hex dump view. HEXDUMP may also be executed for an individual record/record segment using prefix command HEX or HEXD.

The hex dump view is a storage display window which displays a number of fullwords of data in both long hexadecimal and character representation. The hex dump display supports the storage window options popup menu, opened by executing primary command SPM (SHOWPOPUPMENU).

```
Record type: REC-CARD   Fixed(66) Offset=0 Data elements=13
Record> 00000008   Flags: f           Length:      66
     1 00000000 00000008 0000061C 0001C3C4      . . . .CD
    17 C5D3E340 40400675 48756764 74656CD4  ELT .İçİÄÄÄÄ%M
    33 D940C340 E2D3C1C9 D5404040 40404040  R C SLAIN
    49 40404040 40404040 0200409C 0200501C      . æ. &.
    65 719C                                     Éæ
```

Unlike other storage display windows, altering the number of fullwords displayed per line and/or including display of the data's address in storage will not attempt to resize the window display.

For EDIT, both the hexadecimal and character representations of the data may be updated whereby changes to one representation of the data are automatically reflected in the other.

Beware, however, that overtyping the character representation of the data will update **all** fullwords in the updated line so they reflect the character data. In the character representation, a null character (x'00') is represented by blank (EBCDIC x'40') and other non-printable characters are represented by dots (EBCDIC x'4B'). Therefore, updating the character representation will result in these characters replacing the original hexadecimal values.

If parameter NEW is specified, the hex dump display is opened in a new window view allowing the user to preserve the current data display format in the original window view.

## LEFT

### Syntax:

```
>>-- Left -----><
      |-----|
      +-----+--- Cursor -----+
      |                               |
      |                               +--- CSR -----+
      |                               |
      +-----+--- Data -----+
      |                               |
      +--- ALL -----+             +--- Half -----+
      |                               |
      |                               +--- Max -----+
      |                               |
      |                               +--- Page -----+
      |                               |
      |                               +--- n_cols -----+
      |                               |
```

### Description:

Primary command LEFT has been enhanced to support parameter ALL on horizontal scrolling of records in a multi-record view when scrolling by CURSOR is not used. ALL indicates that scrolling is to apply to **all** records in the display and not only those records that are assigned the default record type.

## LOCATE

**Syntax:**

```

+- Locate +-+           +- : +-+ +- RECOrd +-+
| | | | | | | | | | | | | | | | | | | | | |
>>+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----><

```

**Description:**

Primary command LOCATE has been enhanced as follows:  
 LOCATE *record\_num* primary command has been enhanced for segmented-record display and the current setting of PREFIX option PHYSICAL/LOGICAL. If PHYSICAL is in effect, *record\_num* corresponds to a record number whereas, if LOGICAL is in effect, *record\_num* corresponds to a record segment number within the file.

## NOND

**Syntax:**

```

>>-- NOND -----><

```

**Description:**

NOND is supplied as an edit macro. NOND exploits the new NONDISPLAY colour setting to toggle the display of underscores on printable text in character data fields which include unprintable characters.

## ONLY

**Syntax:**

```

+- EQ +-+           +- CHARs +-+
| | | | | | | | | | | | | | | | | |
>>-- Only -----+----- string -----+-----+-----+-----+-----+----->
| | | | | | | | | | | | | | | | | | | | | |
| +- op +-+           +- PREFIX +-+ +- EX +-+
| | | | | | | | | | | | | | | | | | | | | |
+- VALID -----+ +- SUFFIX +-+ +- NX +-+
| | | | | | | | | | | | | | | | | | | | | |
+- INVALID -----+ +- WORD -----+ +- X -----+

>----- | ONLY parameters | -----><

```

**Description:**

ONLY primary command has been enhanced to accommodate the same features as described for CHANGE above.

## RIGHT

**Syntax:**

```

>>- Right -----+-----><
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
+- ALL -----+ +- Cursor -----+
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
+- n_cols -----+

```



**Description:**

Primary command RIGHT has been enhanced to support parameter ALL on horizontal scrolling of records in a multi-record view when scrolling by CURSOR is not used. ALL indicates that scrolling is to apply to **all** records in the display and not only those records that are assigned the default record type.

---

**REPLACE**

---

**Syntax:**

```
>>-- REPlace -----><
      |             |             |             |
      +- fileid -+   +- .name1 -- .name2 ---+
                        (1)
```

**Note:**

1. If a group of lines are not specified using line label names, then a group of lines must be specified using one of the following line (prefix area) commands:

- ◆ **C** or **CC** (Copy), if the group of lines in the current file is to be preserved following successful execution of REPLACE.
- ◆ **M** or **MM** (Move), if the group of lines in the current file is to be deleted following successful execution of REPLACE.

**Description:**

Replace the contents of a sequential or VSAM data set, PDS/PDSE library member or HFS file, with a group of lines extracted from the current SDE edit or browse view. The target sequential or library data set may be uncataloged.

If the target file does not already exist, then one of the following will occur:

1. If output is to a member of an existing PDS/PDSE library or to an HFS file, the target file will be created automatically created.
2. If output is to a non-existent data set or to a member of a non-existent PDS/PDSE library, the **Allocate Non-VSAM** panel is displayed in order to allocate the new data set.

Specify REPLACE with no parameters to open the SDE CREATE/REPLACE file panel.

Note that the existing REPLACE primary command, used to replace individual field values in edited records, has been renamed as **REPLACELINE** (minimum abbreviation REPLACEL).

---

**SHIFT**

---

**Syntax:**

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

      +--- 1 -----+             +- .ZCSR --+ +- .ZLAST -+
      |             |             |             |
      +- n_lines -+   +- EX -+   +- .name1 -+ +- .name2 -+
      |             |             |             |
      +- ALL -----+   +- NX -+
                        |             |
                        +- X ---+
```

**Description:**

Supported for unformatted record view only, SHIFT moves record data within the left and right bounds columns, left or right without altering relative spacing. (See SDE **BOUNDS** option.)

Record data which is shifted left beyond the left bounds column or right beyond the right bounds column gets truncated.

Shift left and shift right may also be executed selected lines using the "("(" and ")"")" line (prefix area) commands respectively.

---

## XREFLIB

---

### Syntax:

```
>>-- XREFLIB -- xref_library+-----+-----><
      |                               |
      +- (member_mask) -+
```

### Description:

XREFLIB is supplied as an edit macro. XREFLIB creates a batch job to convert multiple XREF library members to corresponding SELCOPY/i SDOs. The PDS or PDSE library containing the XREF members must be specified with an optional member mask.

## Line Commands

The following structured data editor line (prefix area) commands have been introduced in release 3.20.

- ([n] )([n]      For unformatted record display only, **SHIFT** a line or block of lines *n* columns to the left. Characters shifted past the current BOUNDS setting are truncated.
- )([n] )([n]      For unformatted record display only, **SHIFT** a line or block of lines *n* columns to the right. Characters shifted past the current BOUNDS setting are truncated.

## SET/QUERY/EXTRACT Options

The following structured data editor SET, QUERY and EXTRACT options have been enhanced or introduced in release 3.20.

---

## ASCII

---

### Syntax:

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

```
>>--- Query ----- ASCii -----><
```

```
>>--- EXtract --- /ASCii/ -----><
```

### Description:

This option causes data in all character (AN) fields to be interpreted in ASCII format. SET ASCII takes effect at the View level and its setting is saved if **SAVEOPTIONS ON** is in effect.

---

## ARRAYALL

---

**Syntax:**

```
>>+-----+--- ARRAYALL -----+--- ON ---+-----><
|         |         |         |         |         |
+- SET ----+         +- OFF ---+
>>--- Query ----- ARRAYALL -----><
>>--- EXtract --- /ARRAYALL/ -----><
```

**Description:**

SELCOPY/i data edit of formatted data containing variable length array (OCCURS DEPENDING) elements will, by default, display the maximum possible number of array elements that may exist within the array.

This option determines whether null elements of the array that are not included by the "depending on" field, are displayed in formatted, single-record (MAP) view.

SET ARRAYALL takes effect at the View level and its setting is saved if **SAVEOPTIONS ON** is in effect.

---

## ARRAYASCHARACTER

---

**Syntax:**

```
>>+-----+--- ARRAYAscharacter -----+----->
|         |         |         |         |         |
+- SET ----+         +- ON ---+
|         |         |         |         |         |
|         |         |         |         |         |
+- OFF ---+         +- field_col ---+
>+-----+-----><
|         |         |         |         |         |
+- FOR ---+-----+ record_type -----+-----+
|         |         |         |         |         |
+- REcord --+         +- IN ---+-----+ struct_name ---+
|         |         |         |         |         |
|         |         |         |         |         |
+- STRUCTure --+
|         |         |         |         |         |
```

**Description:**

Applicable only to formatted, single dimension array (OCCURS) fields comprising single character (PIC X) elements with the number of array elements defined by a value in a separate numeric field (DEPENDING). ARRAYASCHARACTER option determines whether an array field of this type is displayed as a single, updateable variable length character field (XVARCHAR) or as a number of single character array element fields. **ARRC** and **ARRX** are synonyms for SET ARRAYASCHARACTER ON and SET ARRAYASCHARACTER OFF respectively.

SET ARRAYASCHARACTER takes effect at the View level.

---

## BOUNDS

---

**Syntax:**

```
>>+-----+-----+ BOUNds -----+-----><
|         |         |         |         |         |
+- SET ----+         +- BNds -----+ +- left_col -- right_col ---+
>>--- Query ----- BOUNds -----><
|         |         |         |         |         |
|         |         |         |         |         |
+- BNds -----+
>>--- EXtract -----+ /BOUNds/ -----><
|         |         |         |         |         |
+- /BNds/ -----+
```

**Description:**

Applicable to formatted and unformatted, multi and single record views but invalid for DB2 table edit, BOUNDS defines the leftmost and rightmost columns between which the **CHANGE**, **EXCLUDE**, **FIND**, **ONLY** and **SHIFT** commands will operate.

By default, the left bound is set to 1 and the right bound is set to the maximum defined record length plus 1 (LRECL+1) for the file being edited. Note that an error occurs if an attempt is made to set either bound to a value greater than the maximum record length.

Execute SET BOUNDS with no parameters to reset the left and right bounds columns to these defaults.

For variable length records, BOUNDS allows the record length of an individual variable length record to either be preserved or increased (up to the defined maximum) following execution of the SHIFT primary command or the ")"))" or "((((" line (prefix area) commands. When the right bound is set at a column number which is less than or equal to a record's current record length, that record's length is preserved following a shift operation. When the right bound is set at a column number greater than a record's current record length, shifting data left or right will alter the length of that record.

SET BOUNDS takes effect at the View level.

**CCSID**

**Syntax:**

```
>>+-----+----- CCSID ----- ccsid ----->>
  |         |         |
+- SET -----+

>>--- Query ----- CCSID ----->>

>>--- EXTRACT ----- /CCSID/ ----->>
```

**Description:**

SELCOPY/i uses CCSID encoding to translate characters between the program CCSID and the local CCSID for facilities that include XML generation (XMLGEN), regular expressions and display of help pages.

This option controls the CCSID encoding to be used as the local CCSID for SELCOPY/i batch execution (SDEAMAIN) only. Note that, SELCOPY/i interactive execution always interprets the local CCSID encoding scheme to be that of the user's 3270 terminal.

The specified *ccsid* must be the number of an EBCDIC SBCS (Single Byte Character Set) CCSID otherwise an error is returned.

QUERY and EXTRACT operations return 4 values as follow:

1. The user's 3270 terminal defined CCSID.
2. The user's 3270 terminal defined Character Set.
3. The user's 3270 terminal defined Code Page. (In most cases, this will be the same as the CCSID.)
4. The CCSID used by SELCOPY/i batch processing (SDEAMAIN).

By default, the CCSID value for batch processing is set to be the same as the user's local 3270 terminal and is set the first time the user executes a structured data edit command in SELCOPY/i 3.20.

**COLOUR (COLOR)**

**Syntax:**

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

**Description:**

The following options have been included as parameters to the SET COLOUR (COLOR) option:

1. **NONDISPLAY** controls the appearance of printable characters in a character field containing unprintable characters. By default, these characters will be displayed as Turquoise Uscore.
2. **PRIMARY** controls appearance of untagged, unmarked primary record segment text within the display area. By default, this text will be displayed as Turquoise NONE.

QUERY and EXTRACT COLOUR (COLOR) include an entry for the current NONDISPLAY and PRIMARY settings.

**GROUPASCHARACTER**

**Syntax:**

```

                +- ON  +-
                |      |
>>+-----+----- GROUPAscharacter -----+-----+----->
    |      |      |      |      |      |      |
    +- SET  +-----+ +- OFF  +-+ +- field_col +-+
    |      |      |      |      |      |      |
>+-----+-----+-----+-----+-----+-----+-----<<
    |      |      |      |      |      |      |
    +- FOR  +-----+ +- record_type -----+-----+-----|
    |      |      |      |      |      |      |
    +- RECOrd +-+ +- IN  +-----+ +- struct_name +-+
    |      |      |      |      |      |      |
    +- STRUCture +-+
    
```

**Description:**

Applicable only to formatted data, this option determines whether a specified group (structure) field, defined within a specified record type mapping, is to be displayed as a fixed length character field or as a formatted group of its component fields.

When a group field which comprises variable length field elements is displayed as a single character field, the group is first expanded to its maximum length. As a safety mechanism and so avoiding destruction of the group field structure, the character display of this variable length, expanded group field is non-enterable and cannot be altered by a CHANGE operation.

If the specified field name *field\_col* is not a group field, then the GROUPASCHARACTER option is applied to the next higher level group field within which the field is defined.

Note that the record-type is itself a group field, so specifying a field name whose next higher level group is the record-type group will display as character all data in the record/segment mapped by its assigned record-type. This is not the same as displaying the record data in unmapped (CHAR or UNFMT) format.

SET GROUPASCHARACTER takes effect at the View level.

**MAPPING**

**Syntax:**

```

                +- ON  +-
                |      |
>>+-----+----- MAPPING -----+-----+----->>
    |      |      |      |      |      |
    +- SET  +-----+ +- OFF  +-+
    |      |      |      |      |      |
>>--- Query ----- MAPPING ----->>
>>--- EXTRACT --- /MAPPING/ ----->>
    
```

**Description:**

This option controls whether formatted data is displayed in its mapped or unmapped format.

With MAPPING OFF, the record type assigned to formatted records or record segments persists but is not used to format the display of the record data. The record data is displayed in its unformatted (unmapped) format, instead.

With MAPPING ON, the record type assigned to formatted records or record segments is re-applied to the display of the record data.

SET MAPPING does not affect unformatted record data. i.e. SDE EDIT/BROWSE of a data set no USING structure (SDO) specified.

SET MAPPING takes effect at the View level and its setting is saved if **SAVEOPTIONS ON** is in effect.

## PREFIX

### Syntax:

```

>>+-----+--- PREFIX ---+ ON ---+ Left ---+ 6 ---+ LOGical ---+<<
|         |             |   |         |         |         |
+- SET ----+         +- Off -+ + Right -+ + n_bytes -+ + Physical -+
    
```

### Description:

The following LOGICAL and PHYSICAL parameters have been included as parameters to the SET PREFIX option. These are applicable to segmented record display only and determine whether the prefix area contains the physical record numbers or the segment numbers within the file. The default option is LOGICAL.

## SAVEOPTIONS

### Syntax:

```

>>+-----+--- SAVEOPTions ---+ ON ---+-----<<
|         |             |   |         |
+- SET ----+         +- OFF ---+

>>--- Query ----- SAVEOPTions -----<<

>>--- EXtract --- /SAVEOPTions/ -----<<
    
```

### Description:

This option controls whether or not the following SDE options are saved for use in subsequent SDE views opened in the current and subsequent SELCOPY/i sessions:

ABBREVIATION	IDWARNING	QSEPARATOR
ASCII	LEVEL	RECINFO
AUTOSAVE	LOADWARNING	REFERENCE
AUXDSNPREFIX	MAPPING	SCALE
CAPS	MSGLINE	TYPE
COLATTRIBUTES	MULTIPOINT	UNDOING
COLOUR, COLOR	OFFSET	UNNAMED
GROUP	PAD	WRAP
IDSCOPE	PREFIX	ZEROS

Option SAVEOPTIONS takes effect at the Global level.

## SESSION

### Syntax:

```

>>--- Query ----- SESSion -----<<

>>--- EXtract --- /SESSion/ -----<<
    
```

**Description:**

QUERY and EXTRACT SESSION obtain the following general information about the current Structured Data (SDE) Edit/Browse window view.

1. The SDE window view type. (Browse, Edit Reuse, Edit Update or Edit Auxiliary)
2. The SDE record management technique. (InMemory, InPlace, InPlaceInMemory, Auxiliary or KSDS)

**SIZE**

**Syntax:**

```
>>--- Query ----- SIZE -----<<
>>--- EXTRACT --- /SIZE/ -----<<
```

**Description:**

Obtain the number of segments and records in the current file or DB2 table. For DB2 table rows and non-segmented records, the number of segments and records are the same.

Size also returns ">" (greater than) to indicate that the number of records reported does not reflect the whole file size, or "=" (equals) to indicate that the number of records reported does reflect the whole file size.

">" may be displayed for BROWSE or UPDATE-in-place EDIT, where sequential processing of a file is suspended pending a request to display more records. "=" is displayed when all records have been processed.

QUERY and EXTRACT SIZE obtain the following information about the size of the data displayed in the current Structured Data (SDE) Edit/Browse window view:

1. The number of record segments in the file.
2. The number of records/rows in the file or DB2 table.
3. Indicator of whether or not the size value reflects the file size. (">" or "=")

**WINPOS**

**Syntax:**

```

          +- View --+
          |         |
>>+-----+ WINPOS +-----+ row -- col -----<<
    |     |         |         |
    +- SET +-      +- FRAMe +-
>>--- Query ----- WINPOS -----<<
>>--- EXTRACT --- /WINPOS/ -----<<
```

**Description:**

As supported for the text editor, WINPOS will position the current edit view at the specified row and column within the frame (MDI parent) client area, or position the current frame (MDI parent) window at the specified row and column of the screen.

**WINSIZE**

**Syntax:**

```

          +- View --+
          |         |
>>+-----+ WINSIZE +-----+ rows -- cols -----<<
    |     |         |         |
    +- SET +-      +- FRAMe +-
>>--- Query ----- WINSIZE -----<<
>>--- EXTRACT --- /WINSIZE/ -----<<
```

**Description:**

As supported for the text editor, WINSIZE will resize the current edit view or frame (MDI parent) window to the specified number of rows and columns. The position of the window (i.e. top left corner) is unchanged.

---

**ZEROS**


---

**Syntax:**

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

```
>>--- Query ----- Zeros -----><
```

```
>>--- EXtract --- /Zeros/ -----><
```

**Description:**

This option causes values in all numeric fields to be prefixed by zeros up to the width of the field display.

SET ZEROS takes effect at the View level and its setting is saved if **SAVEOPTIONS ON** is in effect.



# SELCOPY/i DB2 Objects & SQL Execution

SELCOPY/i 3.20 introduces the following DB2 features:

## EXECSQL

### Syntax:

```
>>--- EXECSQL -----><
```

### Description:

Primary command EXECSQL may be issued to execute all SQL statements saved in the focus text edit view window. Note that EXECSQL is not yet supported in SELCOPY/i batch (SDEAMAIN).

Output from the SQL execution is displayed in a temporary structured report on completion of the EXECSQL execution.

## Create Table Space

SELCOPY/i DB2 CREATE TABLESPACE panels consisting of a sequence of panel views is now included amongst the create DB2 objects menu panel. (=12.5 or DB2 5)

A Work File Table Space or a LOB, Segmented, Partitioned, Universal Partition-by-Growth or Universal Partition-by-Range table space may be created and options exist to model the new table space from a filtered list of existing table spaces in the current server.

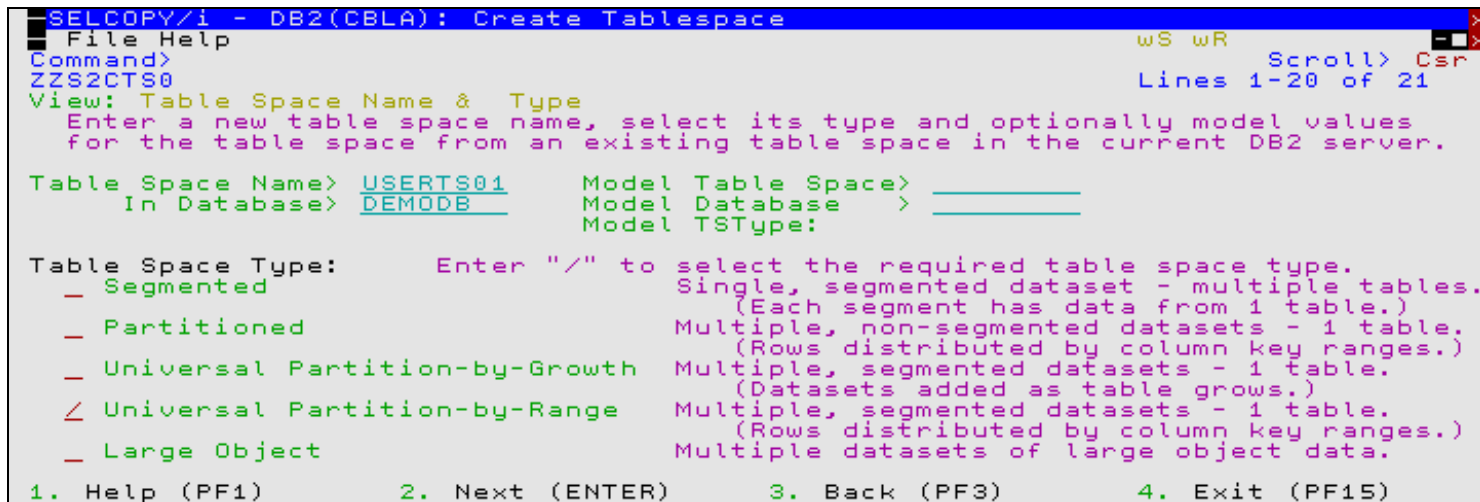


Figure 15. DB2: Create Table Space Panel.

## Create Table

SELCOPY/i DB2 CREATE TABLES panels consisting of a sequence of panel views is now also included amongst the create DB2 objects menu panel. (=12.5 or DB2 5)

Menu options include creation of a Base or Materialized Query table. The sequence of panels displayed vary depending on options selected. Panel options exist to model attributes of the new table on one or more tables in the current server. Table space, database, procedure names, column and constraint definitions may all be modelled from existing tables which may replace or add to existing definitions entered by the user and/or generated by a previous modelling task.

```

SELCOPY/i - DB2(CBLA): Create Table
File Help                               wS wR
Command>                               Scroll> Csr
ZZS2CT00                               Lines 1-20 of 21
View: Table Name & Location             Action: Create Base Table
Enter a new table name and optionally a table space at the current server.

A blank TableSpace value will create the table in an implicitly defined
tablespace in the named (or a default) database.
A blank Database value with a non-blank Tablespace will use that explicitly
defined tablespace in the default database DSNDB04. If an explicitly named
tablespace does not already exist then a TableSpace error is returned.

Use wildcard character "%" in the database and/or tablespace field values
to select from a list of filtered, eligible entries.

Owner> _____ + (optional)
Name> _____ +

DataBase> _____ (optional)   TableSpace> _____ (optional)
TSType:  Implicitly defined. (Segmented or Universal Partitioned)

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

```

Figure 16. DB2: Create Table Panel.

# Maintenance Applied

---

The following table identifies maintenance to SELCOPY Product Suite 3.10 that has been applied at source to SELCOPY Product Suite 3.20.

Details of each SYSMOD may be found at the CBL web page entitled "[SELCOPY Product Suite 3.10 Maintenance Summary](#)".

<b>Service Package Id</b>	<b>SYSMOD</b>
X0000014	RS00016 RI00004
X0000015	RS00017
X0000016	RS00018
X0000018	RS00020 RI00005 RI00007
X0000019	RS00021 RI00008 RI00009
X0000020	RS00022
X0000021	RS00023 RS00024
X0000022	RS00025
X0000023	RI00010
S0000027	QS00038