



Product Suite New Features

Release 3.10

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/cblidoc.html>

Contents

SELCOPY Product Suite 3.10 New Features.....	1
Documentation Notes.....	1
SELCOPY/i Panel Interface.....	2
SELCOPY/i User INI File.....	3
Segmented Record Browse/Edit.....	4
Segmented Records.....	4
Segment Record-Type Definition and Assignment.....	4
Segment Record-Type Identification Criteria.....	4
Segmented Records SDE Browse/Edit View.....	5
Navigating Segmented Record Display.....	5
Segmented Record Edit.....	6
Data Updates that affect Record Segment Mapping.....	6
Full Edit Functionality.....	6
Record Truncation.....	6
Full Edit Operations.....	7
Compare Files Utility.....	8
Source File Types.....	8
Output Report.....	8
Unformatted Compare.....	9
Formatted Compare.....	9
Hierarchical Compare.....	9
SELCOPY/i Structured Data Editor New Features.....	11
File-AID Migration Utility.....	11
COBOL REPLACE.....	11
IDENTIFY.....	11
LOCATE.....	12
WHERE, MORE & LESS.....	12
ONLY.....	13
DISPLAY RECTYPES.....	13
LAYOUT.....	13
SET/QUERY/EXTRACT Options.....	14
Maintenance Applied.....	15

SELCOPY Product Suite 3.10 New Features

Documentation Notes

Information in this New Feature List details changes introduced to CBL software since SELCOPY Product Suite 3.00.

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.html>.

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

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

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/i Panel Interface

SELCOPY/i Interactive Panels (IPOs) were first introduced in release 3.00 to offer ISPF style panels to a selection of commonly used SELCOPY/i utilities. Backed by customer requests, this panel driven interface has been extended to include a suite of nested menus that provide access to all SELCOPY/i functions and utility panels.

At the top of the panel tree is the SELCOPY/i Primary Option menu which is presented to the user automatically on startup of SELCOPY/i 3.10. This panel is opened instead of the user's HOME (CMX) command centre file which was the default action on startup of previous releases of SELCOPY/i.

Note that the HOME file may still be opened by selecting option 4. from the Primary Options Menu. Also, if the user wants to revert back to opening the HOME file automatically on startup (or open both the Primary Menu and the HOME file) then this may be controlled via the "Settings", "Startup" menu option (=0.1).

```
SELCOPY/i - Primary Option Menu
File SwapList Window Help
Command>
ZZSGPRIM
wS wR
Scroll>
Lines 1-20 of 21
Csr

0 Settings          Set SELCOPY/i options
1 Text Edit        Edit/View small text-type files
2 Data Edit        Edit/Browse potentially large data files
3 List             List Volumes,VTOCs,Datasets,Members etc
4 Home            Edit and execute point-and-shoot commands
5 Copy/Reformat    File Copy with optional copybook reformat
6 Search/Update    File Search/Update/Copy/Reformat
7 Compare          File/Library Compare Utilities
8 Utilities        General utilities
9 Structure        Create structure from copybooks etc
10 Filter          Create record selection filter
11 Alloc/Define    Create new VSAM or Sequential datasets
12 DB2            Work with DB2, browse/edit tables etc
W Window List     Display active windows, select with cursor to switch focus
X Exit            Exit SELCOPY/i

User: NBJ2
Version: 3.1B
Date: 2012/03/09
Time: 10:26:38
OpSys: z/OS 1.11.0
System: ADCD
VM User: ZOS111

Use PF9 to switch between SELCOPY/i display windows.
Use PF21 to switch to other ISPF split-screens.
Use "=" command(+optional fastpath e.g. =3.4) to access this menu/sub-options.
```

Figure 1. Primary Option Menu Panel.

If the user operates within a 3270 terminal of width 80 columns (e.g 3270 Models 2/3/4), then all panel windows, lists and edit window views are displayed in maximised (full screen) format by default.

However, since SELCOPY/i is a windowed environment, there is no requirement to close a panel in order to simply return to the previous options menu and select a different application. Instead, <PF9> may be used to cycle through the opened panels, lists and edit window views until focus is back on the required window. Alternatively, the required window view may be selected from the SELCOPY/i list of opened windows, displayed on executing command "WL" or selecting "Window" and, if in an edit view, "All Windows" from the menu bar.

Returning focus to the Primary Option Menu panel may be achieved simply by entering "=" (equals) at any command prompt. If required, a fast path may be specified immediately following the "=" symbol to directly open sub-panels of the Primary Option menu. (e.g. =0.4.1 for "COBOL Compiler options.")

SELCOPY/i User INI File

A SELCOPY/i User INI file must exist for each user of SELCOPY/i in order to define user specific environment options. This data set has a DSN as defined by the System.UserIniFile option in the SELCOPY/i System INI file.

In previous releases, the first time a user started SELCOPY/i, the FIRSTUSE facility would present the user with a series of prompts to provide preferred environment options and also prompts to allocate the User INI file, the User's HOME command centre file and the User's Text Edit macro library.

From SELCOPY/i 3.10 onwards, new users will no longer be prompted for configuration options, nor will they be prompted to allocate the User INI file, HOME file and macro library. Instead, a set of default option values will automatically be configured for the user and, if they do not already exist, the user's data sets will be allocated in the background as SELCOPY/i is started.

Users may configure their INI file environment options via the "Settings" interface panels available on selecting option 0. from the Primary Options Menu.

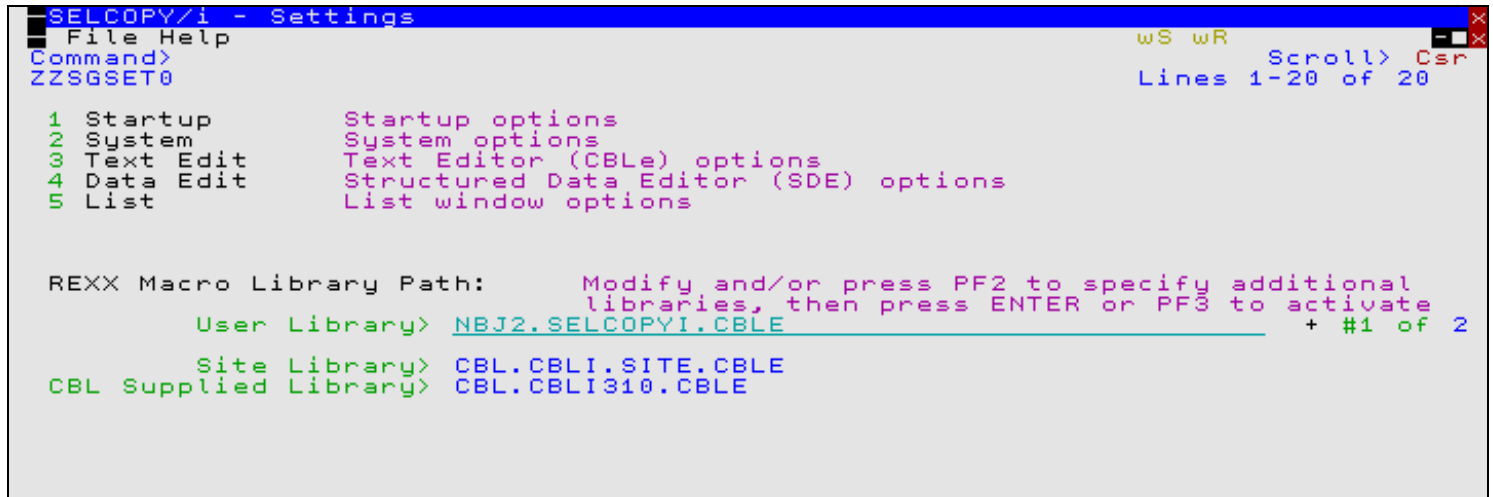


Figure 2. Settings Menu Panel.

Segmented Record Browse/Edit

SELCOPY/i 3.10 introduces support for browse and edit of segmented data records. This includes enhancements to the existing CREATE STRUCTURE command, and introduces new commands, SDE functions and record data display format.

Segmented Records

Records of a file may be organised in such a way that they are split into a number of record segments, each segment being mapped by a unique structure (COBOL group or PL1 major/minor structure).

These segmented records begin with a single primary (base) segment, immediately followed by any number of non-overlapping, secondary segments. A secondary segment may have the same or different segment record-type (RTO) mapping as other secondary segments in the record.

```
Record: 1
+-----+-----+-----+-----+-----+
| Primary_1 | Secondary_1 | Secondary_1 | Secondary_1 | Secondary_1 |
+-----+-----+-----+-----+-----+

Record: 2
+-----+-----+-----+-----+
| Primary_1 | Secondary_1 | Secondary_2 | Secondary_2 |
+-----+-----+-----+-----+

Record: 3
+-----+-----+-----+-----+
| Primary_2 | Secondary_1 | Secondary_4 |
+-----+-----+-----+-----+

Record: 4
+-----+-----+-----+-----+
| Primary_1 | Secondary_1 | Secondary_2 | Secondary_3 |
+-----+-----+-----+-----+
```

The record data must contain ID fields that identify which segment mapping is to be used to format individual segments of the record.

ID fields that identify a primary segment mapping must exist within the primary segment data. ID fields that identify a secondary segment mapping may exist within the secondary segment data, within the data of any previously mapped segment belonging to the same record, or, specifically, within the primary segment data.

Segment Record-Type Definition and Assignment

SELCOPY/i SDE treats each segment of a segmented record as if it was a non-segmented record so that a record-type definition within the SDE structure (SDO) is used to map data in a record segment.

A SELCOPY/i SDE structure may be generated using the **CREATE STRUCTURE** command or **Create Structure (SDO) Menu** panels (=9), in order to format segmented records so that a record type (RTO) mapping definition exists for each primary or secondary record segment in the file.

Each RTO is defined as being either a primary (PRI) or secondary (SEC) segment map, or the default (primary) segment map (DEF). Except for the default segment RTO, which must **not** be defined with segment identification criteria, all primary and secondary segment RTOs must be defined with (USE WHEN) segment identification criteria.

A primary segment RTO is assigned to a primary (base) segment of the record. However, if ID fields do not satisfy the segment identification criteria for any of the primary segment RTOs, the default segment RTO is used.

Unprocessed record data that follows the primary segment is assigned a secondary segment RTO based on associated identification criteria. The assignment of secondary segment RTOs is repeated for all remaining record data until the end of record is reached or the remaining data does not satisfy any segment identification criteria belonging to the secondary segment RTOs. In the latter case, the remaining record data is displayed as unformatted character data with the record type "UnmappedSeg".

Segment Record-Type Identification Criteria

CREATE STRUCTURE command USE WHEN segment identification criteria is mandatory for each non-default primary and secondary segment RTO definition.

Segment identification criteria associated with each RTO definition is based on an SDE **expression**. This identifies all ID fields within the record data and also values against which those ID fields will be tested in order to determine whether the RTO segment

mapping is suitable to be assigned to an individual record segment.

Within any RTO segment mapping USE WHEN expression, ID fields may be referenced using the following methods:

Unformatted data position and length

Built-in function **SEGPOS()** may be used in the USE WHEN *expression* record type identification criteria to reference a data field in the current segment or the segment immediately preceding the current segment. Similarly, **BASPOS()** may be used to reference a data field in the primary (base) segment only.

e.g. The following identification criterion is based on an ID field of length 3 characters located 10 bytes before the end of the segment immediately preceding the current segment:

```
USE WHEN SegPos(-10,3) = C'X12'
```

If the ID field is a Packed Decimal value then **LEFTDEC()** may be used to obtain the numeric value with or without a scale. e.g. The following ID field is a packed decimal field located at position 12 of the primary segment:

```
USE WHEN LeftDec(BasPos(12)) > 31
```

Formatted data field

A formatted ID field references a field in the current segment or within a previous segment of a specific structure (record type) name. If referencing a field in a previous segment, the record type name which maps that segment must be specified as a high level qualifier to the field name (i.e. *record-type.field_name*). If more than one preceding segment is mapped using *record-type*, then the one closest to the current segment is tested.

Because formatted fields have a specific data type, the USE WHEN expression may involve operators and terms based on the data type of the ID field. e.g. The following identification criterion is based on a floating point ID field XVAL in a preceding segment of record type MAPBASE1:

```
USE WHEN MAPBASE1.XVAL > 27.83
```

Segmented Records SDE Browse/Edit View

As for display of non-segmented records, files containing segmented records may be displayed in multi-record (FORMAT TABLE/CHAR/HEX) or single-record (FORMAT SINGLE) view.

Primary commands **CHAR**, **MAP** (FMT), **VFMT** and **UNFMT** are supported to display the current segment in multi-record unformatted character view, single-record formatted view, multi-record formatted view and single-record unformatted character view respectively.

Data in record segments is displayed in exactly the same way as non-segmented records. Prefix area, record type headers and record information is also displayed as for non-segmented records with the following exceptions:

1. Segmented record header display differs from that of non-segmented records only in that, instead of "**Record Type:**", header line 1 of each group of matching segment types displays either "**Base:**", for primary segments; "**Base(D):**", for default primary segments; or "**Segment:**", for secondary segments.

In order to easily distinguish between primary and secondary segments, the header line 1 of secondary segments is coloured yellow by default. See SET/QUERY/EXTRACT option **COLOUR** (or COLOR) SEGMENT.

2. For secondary segments only, the record information id column, displayed using the SET/QUERY/EXTRACT option **RECINFO**, displays the offset into the record of the start of the segment.

Primary segments display the TTR/Offset or RBA value of the start of the record as for non-segmented records.

3. Prefix area numbers no longer correspond to record numbers within the edited file but to the sequence of segments. Therefore, navigating the file data is performed by segment and not record number. e.g. "LOCATE 32" will scroll to the 35th (primary or secondary) segment, not the 35th record, of the file.

Navigating Segmented Record Display

In both multi-record and single-record views, the display of segments may be navigated using standard SDE CLI (primary) commands: **FIND**, **LOCATE**, **UP**, **DOWN**, **LEFT**, **RIGHT**, **TOP**, **BOTTOM**, **NEXT** and **PREV**.

Note that, in single-record view, LEFT and RIGHT will scroll backwards and forwards respectively through every segment in the file regardless of the type of segment (primary or secondary) or record to which a segment belongs.

Commands NEXT and PREV may be used to restrict scrolling to only secondary segments belonging to the current record. This is the NEXT and PREV default, however, parameters are supported to scroll to the next and previous primary or unmapped (UnmappedSeg) segments.

Segmented Record Edit

SELCOPY/i SDE segmented record edit is supported for all file organisations other than VSAM KSDS.

Full Edit, Auxiliary Edit and Update-in-Place Edit SDE edit techniques are all supported for segmented record edit.

Data Updates that affect Record Segment Mapping

The RTO mapping assigned to a segment is determined by values in ID fields which may occur in the same segment or within the primary and/or previous secondary segments. Therefore, any change to a segment may affect the RTO mapping identification criteria for that segment or any subsequent segment in the record.

SELCOPY/i SDE is able to detect changes to ID fields which have been identified via a USE WHEN expression that includes only fields referenced explicitly by name or field reference number. Changes to ID field name "Unmapped" (reference number #1) are detected only when record-type formatting is disabled. Where an ID field change is detected, or where the length of the segment data is altered via CHANGE or RECLN updates, the ID flag is set on for the changed segment.

The ID flag, represented by "m" in the record information flag display and **==ID>** in multi-record view prefix area, is an indication that the record to which this segment belongs may require a remap (re-assignment of segment RTOs) based on new ID field values. This flag may be ignored but may be of importance if the user wants to first verify that integrity between the required segment mapping and its identification criteria is maintained prior to saving the data and ending the edit session.

Remap of a record and reset of the ID flag is achieved using the primary command **IDENTIFY**, or prefix (line) commands **ID<n>** or **IDD**. Note that the remap is **not** performed automatically, so giving the user the opportunity to make further changes to the data before executing IDENTIFY. Note that, if ID field values do not satisfy any secondary segment mapping identification criteria, the remainder of that record is displayed as unformatted data (UnmappedSeg).

Remap of a record may also be required where an update has occurred for an ID field identified in a USE WHEN expression by its unformatted record position and length, or when segments have been inserted, deleted, duplicated, copied or moved. For this reason, the default action of the IDENTIFY command is to not only remap records containing segments with the ID flag, but also records flagged as having changed segments. (Option **IDSCOPE** may be used to force IDENTIFY to remap **only** those records containing a segment with ID flag set on).

Full Edit Functionality

Update-in-place edit only supports changing data in existing segments without changing the the length of a segment. In contrast, **full function** edit supports changing data, changing the length of a segment and also the addition and removal of record segments.

When segments are displayed in multi-record view, individual segments (or blocks of consecutive segments) may be replicated, deleted, inserted, moved or copied using standard SDE edit primary and line commands.

In single record view, segments may only be deleted, duplicated or inserted using primary commands **DELETE**, **DUPLICATE** and **INSERT** respectively.

Edit actions that affect the location of a primary segment within the file, directly affect the location of the record to which it belongs. In contrast, secondary segments may be deleted, moved, copied, duplicated and inserted without affecting the location of a record within the file. For secondary segments, these edit operations may only increase or decrease the length of existing records.

Record Truncation

Beware that full edit operations involving primary segments can potentially split and join together records in the file.

If as a result of adding secondary segments to a record, the length of that record exceeds the maximum defined logical record length of the file (LRECL), then remapping the record or executing a save operation will truncate the excess data. For variable record format files, if data at the end of the truncated record represents an incomplete record segment map, then that data is also truncated.

Following an execution of IDENTIFY in which record truncation occurs, the truncated segments are removed from the display, the truncation error flag (**=TRNC>**) is set on in the primary segment of each truncated record and the following message is returned:

```
ZZSD447I IDENTIFY process caused 1 physical records to be truncated
```

The **UNDO** command may be executed to undo the remap and recover the truncated segments. The user may then correct the cause of the truncation.

Following execution of SAVE, FILE or END in which record truncation occurs, the truncated segments are **not** removed from the display, the truncation error flag (**=TRNC>**) is set on in the primary segment of every truncated record and the following message is returned:

ZZSD406E The physical record associated with a base segment is too long (greater than *nnn*). Truncated to length *mmm*.

Where *nnn* is the maximum allowable record length and *mmm* is the length at which the record was truncated. If truncation occurs on execution of FILE or END, the close of the SDE edit view is temporarily disabled in order to allow the user the opportunity to correct the truncation error.

Although the file has been saved to disk with truncated records, the in-storage copy of the records remains unchanged. Therefore, the user may correct and re-save the file data before closing the edit view.

Full Edit Operations

The effects of full edit operations that add, remove and/or reposition record segments are summarised below:

DELETE

Delete individual segments using command DELETE or prefix (line) command **D<n>** (where *n* is the number of segments to be deleted). Alternatively, delete a block of segments using prefix command **DD** on the first and last segments to be deleted.

On deleting a primary (base) segment, the record to which the primary segment belongs is deleted and its secondary segments are appended to segments belonging to the previous record.

Deleting a secondary segment simply removes that segment from the record, hence reducing the record length. All subsequent segments in the record are shifted left to fill the gap left by the deleted segment.

Note that **SEGTYPE PRIMARY** or **SEGTYPE SECONDARY** may be used to force a segment to be primary or secondary respectively. This causes the record to split or join with the previous record as appropriate.

INSERT

Insert individual segments following the focus segment using command INSERT or prefix (line) command **I<n>** (where *n* is the number of segments of the focus segment type to be inserted).

Inserting a new primary segment inserts a new record in the file. If the inserted primary segment is inserted before a secondary segment (i.e. in the middle of an existing record), then the existing record is split so that all secondary segments following the inserted primary segment belong to the new record.

Inserting a new secondary segment simply adds the segment to the focus record.

COPY and DUPLICATE

Copy individual segments using prefix (line) command **C<n>** (where *n* is the number of segments to be copied). Alternatively, copy a block of segments using prefix command **CC** on the first and last segments to be copied. Prefix commands **A** or **B** must be used to specify whether segments are to be copied After or Before the selected segment.

Duplicate a segment using command DUPLICATE or prefix (line) command **R<n>** (where *n* is the number times the segment is to be duplicated). Alternatively, duplicate a block of segments a number of times using prefix command **RR<n>** on the first and last segments to be duplicated.

The effect of a copy or duplicate operation on existing primary and secondary segments is the same as for insert.

MOVE

Move individual segments using prefix (line) command **M<n>** (where *n* is the number of segments to be moved). Alternatively, move a block of segments using prefix command **MM** on the first and last segments to be moved. Prefix commands **A** or **B** must be used to specify whether segments are to be moved After or Before the selected segment.

The effect of a move operation on existing primary and secondary segments located at the destination of the move is the same as for insert.

The effect of a move operation on primary and secondary segments immediately preceding and following the original location of the moved segments is the same as for delete.

Compare Files Utility

SELCOPY/i 3.10 and SELCOPY/i 3.00 with PTF RS00014 applied, include major enhancements to the file compare utility, accessible via the a series of "Compare Files" panels (=7.1) or the **COMPFILE** command.

The Compare Files utility (COMPFILE) provides a set of both basic and extended features that allow the user to compare records in **NEW** and **OLD** versions of a file.

Basic features include:

- Specify the start record.
- Restrict the number of records compared.
- Restrict the number of differences to be reported.
- Restrict the comparison to a specific area of the file records.
- Strip trailing characters prior to record compare.

Extended features include:

- All basic feature options but with separate specifications for the NEW and OLD files where sensible.
- Apply a **structure** (copybook) overlay to map records, and optionally restrict the comparison to specified record-types and/or named fields. This is known as a **formatted compare**.
- Control how re-synchronisation of record pairs should occur following detection of an inserted or deleted record.
- For formatted or unformatted compare, specify **key** segments (at the record-type level) that allow the utility to identify **synchronised pairs** of records.
- Formatted compare supports application of different structures to the NEW and OLD files, with comparison restricted to only those fields that exist in both structures. This allows comparison of NEW and OLD file records where corresponding fields are at different locations within the records and maybe of different data-type or length.

Following Compare Files execution, report output is generated in a structured format suitable for presentation to the user in an SDE window view. To generate this report output and in order to perform advanced record selection and field compare, COMPFILE utilises functions and features provided by the structured data environment (SDE). Therefore, the COMPFILE utility is only available to users who have a licensed version of SELCOPY installed and operational on their system.

Source File Types

COMPFILE can process records from any combination of the following file types:

- Cataloged or uncataloged sequential (including multi-volume) datasets.
- Partitioned dataset (PDS/PDSE) members.
- GDG datasets.
- VSAM (KSDS, ESDS, RRDS, VRDS).
- HFS Files.
- (DB2 Tables planned but not yet supported).

Output Report

The report generated by the compare files utility is a **structured data file**. This is designed to be browsed (not printed) from within a SELCOPY/i session using a structure definition file (SDO) which is also generated automatically during execution of the compare.

Following execution of the compare utility, records are flagged as being **matched** or as having been **inserted, deleted or changed**.

Matched

Records that exist in both the NEW and OLD files forming a synchronised record pair for which the compared data is unchanged (matches).

Inserted

Records that previously did not exist in the old file and so have been inserted into the NEW file.

Deleted

Records that no longer exist in the NEW file and so have been deleted from the OLD file.

Changed

Records that exist in both files forming a synchronised record pair in which the compared data has been changed (i.e does not match).

Determination of synchronised record pairs is achieved by the compare file utility using record synchronisation techniques.

In order to improve readability, the report of consecutive records flagged as having been deleted are grouped together, and similarly for records flagged as having been inserted.

Unformatted Compare

Unformatted compare is the most commonly used format for **text** files containing unformatted records.

By definition, unformatted compare operates on records without application of a structure (SDO) or COBOL/PL1 copybook to format record data. i.e. each record is treated as a single character string.

Basic Unformatted Compare

Basic unformatted compare specifically relates to unformatted compare where selected options apply to **both** files involved in the compare operation. These options are:

- ◇ The compare data start position within the record.
- ◇ The compare data length.
- ◇ The trailing character to be stripped before comparing the data.
- ◇ The first record to be compared. (Nominated by record number, key or RBA.)
- ◇ The number of records to be compared.

Furthermore, record synchronisation technique employed is restricted to **1-TO-1** or **read-ahead** with a read-ahead limit of 100 records and read-ahead matching record count of 1.

Extended Unformatted Compare

Extended unformatted compare allows specification of the same options as basic unformatted compare but with potentially different values for each of the two files in the compare operation. In addition to this, extended unformatted compare allows specification of the following:

- ◇ Record synchronisation techniques **Sorted Keyed** and **Unsorted Keyed** which involves specification of key segments.
- ◇ For read-ahead record synchronisation, non-default values for limit and matching record count. Also the option to allow synchronisation on blank records.
- ◇ The option to perform case-insensitive compare.
- ◇ Report output options to exclude display of changed, inserted and/or deleted records. Also allows specification of a non-default report file DSN.
- ◇ Output file DSNs into which to copy records flagged as being matched, changed, inserted and/or deleted. A separate data set name may be specified for NEW and OLD file records that are attributed these flags.

Formatted Compare

More advanced than unformatted compare, formatted compare is invoked where an SDE structure (SDO), COBOL or PL1 copybook overlay is specified to map record data fields for use in the compare files operation.

Records are treated as comprising a number of data fields of pre-determined lengths and of various data types. Each field within the record may be referenced independently (by field name or field reference number) allowing the user to be more discriminate when selecting records, and fields to be compared.

If a COBOL copybook, PL1 include file or an ADATA file generated from a COBOL or PL1 compilation is specified, then this file will be used to generate a temporary SDO before proceeding with record formatting. Note that a non-temporary SDO may be generated from the COBOL/PL1/ADATA file using the SDE command, **CREATE STRUCTURE**.

Each input record is assigned a record type (RTO) defined in the specified or generated SDO and the field definitions defined by that RTO are used to map the data within the record. SDE determines the record type to be assigned to each record based on any USE WHEN conditions saved in the SDO and the individual record's length.

Formatted compare may be selected via the Compare File utility panel by first selecting **Extended options** from the Compare Files Basic Options view.

Hierarchical Compare

Hierarchical compare is not selected explicitly but is implied when both of the following conditions are true:

1. Formatted compare is used incorporating records assigned to different record types in the SDO.
2. KEY synchronisation is performed with key segments specified as formatted record field names or field reference numbers.

The compare files command, COMPFILE, generated by the dialog panel or entered manually by the user, specifies synchronisation key fields for one or more record types in the specified SDO. The order in which these record types occur in the COMPFILE command also define the levels of record type hierarchy. i.e. The record type synchronisation key definition occurring first identifies the level-1 (highest level) record type, the second definition identifies the level-2 (level-1 child) record type, etc.

Record types with no synchronisation key are the lowest level in the record type hierarchy, i.e. rated lower than any record type that has been defined with a synchronisation key.

Hierarchical compare is sensitive to the level of record type assigned to a record. All records that immediately follow the current record which are assigned record types lower in the record type hierarchy than that of the current record, are treated as being descendants of the current record. These records are grouped with the current record so that record synchronisation does not exceed the bounds of the current hierarchical record group.

This type of compare ensures that only record pairs that belong to the same hierarchical parent record pair can be synchronised.

SELCOPY/i Structured Data Editor New Features

File-AID Migration Utility

In order to assist users with migration from the File-AID product, the ZZSXREF utility has been included in SELCOPY/i 3.10 to perform a direct conversion of File-AID XREF members to SELCOPY/i SDE structure definition files (SDO).

The "Create Structure from XREF File" panel (ZZSGXREF) is opened on selection of option 2. in the "Create Structure (SDO) Menu" and provides an interface to the ZZSXREF utility.

Simply enter the fileid of the XREF library member and output SDO file, then select whether to perform the conversion in the SELCOPY/i foreground or generate JCL for submission to batch.

The resultant SDO may then be used to edit and format data in structured files.

COBOL REPLACE

When SELCOPY/i generates an SDE structure (SDO) from a COBOL copybook to map structured data set records, a COBOL compile is performed.

If the copybook field definitions include *pseudo-text* which is intended to be substituted with values specified via a COBOL REPLACE command, then compiler error messages are returned and completes with RC=12.

SELCOPY/i 3.10 improves on previous releases by supporting System and User INI variables that define REPLACE source and replacement *pseudo-text* strings that apply to all COBOL copybooks compilations.

The "COBOL Compiler Options" panel (ZZSGSETC) is opened on selection of option 1. in the Structured Data Edit (SDE) Settings panel and allows the user to define 12 pairs of ('From:' and 'to:') fields that together generate a COBOL REPLACE statement.

The 'From:' field specifies a *pseudo-text* source string to be replaced. The corresponding 'to:' field specifies a *pseudo-text* replacement string.

IDENTIFY

SELCOPY/i SDE is sensitive to changes made to ID fields that have been identified explicitly by field name or field reference number in the USE WHEN expression. It is also sensitive to data changes when record-type formatting is disabled (i.e. field name "Unmapped" or field reference number #1).

From SELCOPY/i 3.10 onwards, SDE no longer attempts to automatically re-assign a record-type definition, potentially reformatting the display of the data, if updates occur to any of the record ID fields. This allows the user the opportunity to complete any changes being made before re-formatting the record data display.

If the **IDWARNING** option is set on (the default), then where one of these ID field changes is detected or where the length of the record/segment data is altered (via CHANGE or RECLen updates), the ID flag (**==ID>**) is set on for the changed record/segment. This is intended to notify the user that the record/segment data may no longer satisfy the USE WHEN criteria for its assigned record type.

The IDENTIFY CLI (primary) command and prefix (line) commands IDn or IDD/IDD may be executed to force record type re-assignment for selected records in the current SDE edit view. For **segmented record** edit, if any segment within a record is selected by the IDENTIFY operation, then all segments within the record will be re-assigned.

By default, IDENTIFY will not attempt to re-assign record types to records that have not been altered since the file was loaded for edit.

The records selected for re-assignment by an IDENTIFY operation depends on the current value of the **IDSCOPE** option. By default, IDSCOPE is set to CHANGED indicating that all records flagged as having been changed may be selected for record type re-assignment. Alternatively, IDSCOPE FLAGGED indicates that only records with the ID flag set on may be selected by the IDENTIFY operation.

The remap of records, performed on execution of IDENTIFY, may be individually undone and subsequently redone using the UNDO (<PF22>) and REDO (<PF23>) commands.

LOCATE

SELCOPY/i 3.10 enhances the SDE LOCATE command to support the following:

- Scroll the display to a specified field name or field reference number in the focus record.

If field name is specified then it may be fully qualified, partially qualified or unqualified and may include a subscript array element reference.

Regardless of whether the **ABBREVIATION** option has been set on, the group name or field item designators that constitute a field name may be abbreviated, starting with the first letter of the designator. If the specified field name identifies more than one field, then only the first occurrence of a field that matches this name will be located.

Examples

```
LOCATE  COMPUNIT
LOCATE  GROUPX.IX1
LOCATE  ADDR(4)
LOCATE  #22
```

- Scroll the display to a record or record segment assigned the default record type with the specified flag bit (*line_flag*) set on.

Each valid *line_flag* keyword detailed below corresponds to a built-in function. A description of each *line_flag* keyword may be found in its equivalent function description.

<i>line_flag</i> Keyword	Built-in Function
ERRor	CHANGEERROR()
CHAnge Chg	CHANGEOK()
ALTered UPDated	CHANGED()
DATAerror	DATATYPEERROR()
DUPkey	DUPLICATEKEY()
EMPTY	EMPTYSLLOT()
EXcluded X	EXCLUDED()
LABel	HASPOINT()
COMmand CMd	HASPREFIXCMD()
IDentify IDrequired	IDREQUIRED()
NEW	INSERTED()
KEYChanged KEYChg	KEYCHANGED()
LENgtherror	LENGTHERROR()
EOL NOEOL	NOEOL()
SAVE	SAVEREQUIRED()
TRNC TRUNCated	TRUNCATED()
VALERRor	VALUEERROR()

WHERE, MORE & LESS

Like the LOCATE command, SELCOPY/i 3.10 enhances support of the WHERE, MORE and LESS commands to include or exclude only records assigned the default record type that are flagged with the specified *line_flag*. (See LOCATE above for valid *line_flag* values.)

ONLY

Syntax:

```

                                +- CHARs ---+
>>--- Only ----- string -----+-----+----->
                                +- PREFIX +-  +- EX +-
                                +- SUFFIX +-  +- NX +-
                                +- WORD  ---+  +- X  ---+

+-- #ALL -----+-- .ZFIRST ---- .ZLAST -+
>+-----+-----+-----+-----+-----+><
+-- pos1 -----+-- .name1 -----+
|               |               |
|               +- pos2 ---+    +- .name2 ---+
|               |               |
|               +-----+-----+
|               |               |
|               +- , ----+
|               |               |
+-- ( +- field_col -----+ ) +-
|               |               |
+-- field_col1:field_col2 -----+

```

Description:

New command introduced in SELCOPY/i 3.10 and based on the EXCLUDE and FIND ALL commands, ONLY displays all records or record segments assigned the default record type that satisfy the specified search *string* criteria. All records or segments of the default record type that do **not** satisfy the search *string* criteria are excluded.

See the "SELCOPY/i Structured Data (SDE) Manual" for a complete description.

DISPLAY RECTYPES

The DISPLAY RECTYPES (synonym "LR") command may be used to open a SELCOPY/i **list window** and display all record-type (RTO) definitions in the specified (or current) SDE structure (SDO).

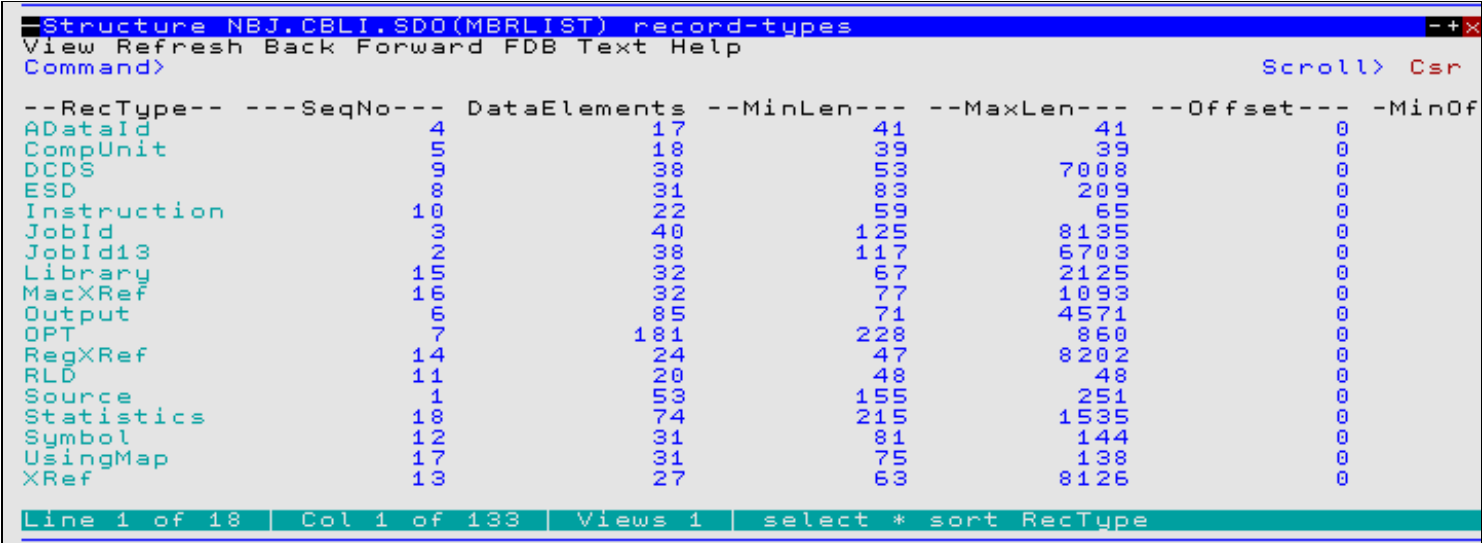


Figure 3. DISPLAY RECTYPES List Window.

LAYOUT

The LAYOUT command may be used to display the record structure **layout list window** detailing field definitions in all record-types in the specified structure file (SDE structure (SDO), COBOL/PL1 copy book or COBOL/PL1 ADATA file).

If executed from an SDE browse/edit view without specifying the name of a structure file, the layout window is displayed for all record-types in the structure used to map records in the current display.

If executed from any other window with no parameters the **Display Record Layout** panel (=9.3) is opened.

The screenshot shows a window titled "SELCOPY/i - Layout from nbj.cbli.sdo(sales)". The window has a menu bar with "View", "Refresh", "Back", "Forward", "FDB", "Text", and "Help". Below the menu bar is a "Command" prompt. The main area displays a table of field information:

	Name	Picture	RefNo	Start	End	Length
1	REC-CUST	Group	1	1	204	204
2	FILEREF	9(5)	2	1	4	4
2	RECNO	9(5)	3	5	8	4
2	CUST-ID	9(5)	4	9	12	4
2	PASS	X(15)	5	13	27	15
2	LASTNAME	X(15)	6	28	42	15
2	FIRSTNAME	X(15)	7	43	57	15
2	COUNTRY	X(2)	8	58	59	2
2	POSTCODE	X(12)	9	60	71	12
2	CITY	X(15)	10	72	86	15
2	HOUSE	9(5)	11	87	90	4
2	STREET	X(25)	12	91	115	25
2	EMAIL	X(35)	13	116	150	35
2	PHONE	X(25)	14	151	175	25
2	MOBILE	X(25)	15	176	200	25
2	BALANCE	S9(5)V99	16	201	204	4
1	REC-CARD	Group	1	1	66	66
2	FILEREF	9(5)	2	1	4	4

At the bottom of the window, a status bar shows "Line 1 of 65", "Col 1 of 52", "Views 1", and "select *".

Figure 4. Layout List Window.

SET/QUERY/EXTRACT Options

The following enhancements to SDE structured data editor options have been introduced in SELCOPY/i release 3.10:

ABBREVIATION

Set, query or extract the current setting (ON or OFF) of the ABBREVIATION option. ON indicates that a field may be referenced using abbreviated field name descriptors.

CAPS

Set, query or extract the current setting (ON or OFF) of the CAPS option. On indicates that automatic capitalisation (upper casing) of text will occur in updated character fields.

DESCRIPTION

Set, query or extract the description of the current SDE structure (SDO).

IDSCOPE

Set, query or extract the current setting (FLAGGED or CHANGED) of the IDENTIFY command. FLAGGED indicates that only records for which the ID flag has been set on may be selected for an IDENTIFY operation, whereas CHANGED (the default) indicates records with either the ID flag or CHANGED flag may be selected.

IDWARNING

Set, query or extract the current setting (ON or OFF) of the ID warning flag option. ON (the default) indicates that the ID flag will be set on for records containing updated ID field data. (ID fields are identified by USE WHEN criteria in the record-type definition.)

TITLE

Set, query or extract the title of the current SDE structure (SDO).

Maintenance Applied

The following table identifies maintenance to SELCOPY Product Suite 3.00 that has been applied at source to SELCOPY Product Suite 3.10.

Details of each SYSMOD may be found at the CBL web page entitled "[SELCOPY Product Suite 3.00 Technical Support](#)".

Service Package Id	SYSMODs
-	QI00012 QI00013 QV00001 QV00002
X0000012	RS00014 RI00003
X0000011	RS00013
X0000010	RS00012
X0000009	RS00011
X0000008	RI00002
X0000007	RS00010
X0000006	RS00009
X0000005	RS00008
X0000004	RS00007
X0000003	RS00006
X0000002	RS00005 RS00004 RS00003 RS00002
X0000001	RS00001 RI00001