Compute (Bridgend) Ltd

CBLi 1.30 New Features

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

Contents

CBLi 1.30 New Features

Documentation Notes

Information in this New Feature List reflects differences between CBLi 1.20 and CBLi 1.30.

The **CBLi Reference and User Guide**, **CBLe Editor Manual**, **CBLi Installation Guide** and **CBLi New Features** are available in Adobe Acrobat PDF format at CBL web page http://www.cbl.com/cblidoc.html.

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 manual.

Section 01: Overview

An overview of the major new features included in CBLi release 1.30 follows.

ISPF style editing

The CBLi file editor, CBLe, has an operational interface compatible with IBM's CMS editor, XEDIT, and Mansfield Software's PC adaptation, KEDIT.

This has proven to be an obstacle to MVS users familiar with the TSO/ISPF editor only.

CBLi 1.30 introduces a new operation mode for the CBLe editor; the ISPF interface, which includes support for:

- Commonly used ISPF primary and line (prefix) commands.
- ISPF page scrolling attributes.
- ISPF PFKey/command line concatenation.

The prevailing operation interface (either ISPF or CBLE) dictates the command environment that has priority.

This is a major enhancement to CBLi and envelopes a large proportion of the new CBLe commands and CBLiINI options contained in this document. For a central reference point, see The CBLe ISPF Edit Interface.

MVS Security Enhancements

By default, CBLi provides users with the ability to display certain system information, execute SELCOPY and CBLVCAT interactively and also execute DB2 dynamic SQL statements.

CBLi 1.30 has facilities to allow security administrators to apply the following restrictions to individual users and groups of users:

- Access to CBLi in TSO and ISPF.
- Access to CBLi running as a VTAM application.
- Authority to query system information.
- Authority to execute SELCOPY interactively.
- Authority to execute CBLVCAT interactively.
- Authority to use DB2 dynamic SQL window.

This new feature depends on resources being defined to the MVS RACF system, or equivalent security package.

See MVS ACF Security Enhancements.

CBLe Environment Variables

- CBLe Standard Environment Variables.

  CBLi 1.20 introduced the CBLe EDITV CLI command which allows users to define CBLe environment edit variables which can be referenced in CBLe REXX macros.

  In addition to user defined variables, CBLi 1.30 includes some useful, standard CBLe environment edit variables. e.g. %user% (User name), %date% (Current date) and %fid% (Current fileid).

  Reference to MVS system symbols and CBLiINI variables is also supported.

  See CBLe Environment Variables for a complete list of standard CBLe environment variables.

- Variable Translation in Commands.

  CBLi 1.20 supported use of CBLe environment variables in CBLe REXX macros only.

  In CBLi 1.30, CBLe standard and user defined environment variables may also be referenced in commands executed from the CBLe command line or via the CMDTEXT facility.

  Where SET ENVVAR is on (this is the default), variable names that are enclosed in the delimiter character (default is % - percent) are translated before the command is executed.

  See SET ENVVARS Command and CBLe Environment Variables for details.

SELCOPY Interactive Enhancements

- SYSIN ddname input.

  Prior to CBLi 1.30, control statements were passed to SELCOPY Interactive by supplying the full DSN of the data set in which they were saved. This had the restriction that all the control statements had to reside in the same data set.

  In CBLi 1.30, an up to 8 character filename (ddname), referencing a data set or concatenation of data sets, may be specified instead of a DSN to reference the input control statements.

  See DDNAME Control Statement Input.

- Library Search Chain

  The location of the SELCOPY program executed and any routines called by the SELCOPY operation, CALL, is determined by the standard search chain for the current environment.

  SELCOPY Interactive in CBLi 1.30 provides users with the ability to include additional libraries to the start of the search chain. This gives the SELCOPY Interactive environment an equivalent to the STEPLIB JCL statement, which may occur in SELCOPY batch jobs.

  The included library path may supplied as one or more DSNs or a single, pre-allocated ddname which, itself, may be a concatenation of library DSNs.

  See LIBPATH Library Search Chain.

Section 02: MVS ACF Security Enhancements

On MVS systems, access to the following CBLi features may be restricted by user name using RACF profiles.

Resource Name	CBLi Feature
System	Access to the 'Operating System' information available via the 'System' menu item on the main desktop or via the commands: `SysI`, `SysLPA`, `SysLL`, `SysAPF`, `SysTask`, `SysStor`, `SysPgm` and `CBLNAME`.
UserTSO	Log on to CBLi under TSO and ISPF.
UserVTAM	Log on to CBLi as VTAM application.
SELCOPY	Use of the SELCOPY Interactive application.
CBLVCAT	Use of the CBLVCAT Interactive application.
DB2	Use of the DB2/SQL Interactive application.

Restrictions are implemented via entries in the CBLi SYSTEM (site wide) CBLiINI file, and cannot be overridden by an individual's user CBLiINI file.

Support for the (RACF) section and the following options have been added.

CBLiINI Variable	Description
ResourceCheck=NO|YES	Use RACF for CBLi resource access checking (default NO).
ResourceClass=ClassName	RACF general resource class (default FACILITY).
SuppressWTO=NO|YES	Suppress access failure WTO messages (default NO).
System=ResourceName	System commands resource name (default CBLI.SYSTEM)
UserTSO=ResourceName	TSO user access resource name (default CBLI.USER.TSO)
UserVTAM=ResourceName	VTAM user access resource name (default CBLI.USER.VTAM)
SELCOPY=ResourceName	Interactive SELCOPY resource name (default CBLI.SELCOPY)
CBLVCAT=ResourceName	Interactive CBLVCAT resource name (default CBLI.CBLVCAT)
DB2=ResourceName	Interactive SQL resource name (default CBLI.DB2)

With RACF.ResourceCheck=YES in effect, users without a minimum of READ access to the resource will be rejected.

RACF must be active and there must be a RACF profile which includes the resource name. The resource name may be generic; for example the profile CBLI.** with UACC(READ) would permit all users access to all resources.

Note that MVS system symbols may be used in the resource names These will be resolved when reading the system CBLiINI file. e.g.

```
(RACF)
   DB2=CBLI.&SYSNAME..DB2     * Interactive SQL resource name
```

Please see the distributed CMX file &PREFIX..CBLI.CMX(RACF) for samples of commands useful in administering CBLi RACF resource restrictions.

Section 03: CBLi CLI Commands

COMMANDLINE Command Parameter Added

Syntax:

```
>>--+- COMMANDLINE -+--+--------------+--+--------+--+-------+--->
    |               |  |              |  |        |  |       |
    +- CLN ---------+  +- windowname -+  +- SHOW -+  +- TOP -+
                                        |        |  |       |
                                        +- HIDE -+  +- BOT -+

 >--+------------------+--+-------------+--><
    |                  |  |             |
    +- PROMPT=prompt -+  +- SCROLL ----+
                        |             |
                        +- NOSCROLL --+
```

Description:

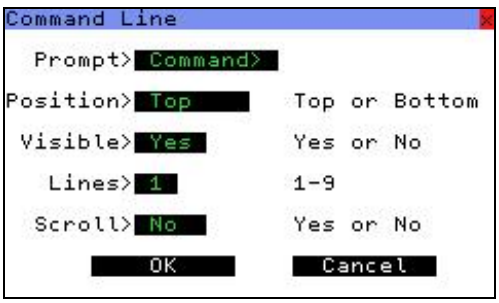
If this command is issued with no parameters or with the window name parameter only, then a dialog box is opened in which the user can define the characteristics of the command line of the specified window.

COMMANDLINE now supports new parameters, SCROLL and NOSCROLL, that take effect in CBLe window, ISPF mode edit views only.

Parameters:

SCROLL NOSCROLL	Show or hide the ISPF Edit style scroll amount entry field. For CBLe INTERFACE=ISPF, the default is SCROLL. For CBLe INTERFACE=CBLE, the default is NOSCROLL.

The Command Line dailog window includes a Scroll field:



ERASE Command Parameter Added

Syntax:

```
>>-- ERAse --+------------+-- fileid ----><
              |            |
              +-- volid: --+
```

Description:

Erase (delete) the specified fileid.

CBLi 1.30 supports erase of MVS data sets that are not cataloged. To accomodate this, the ERASE command supports the new parameter, *volid:* which must be supplied when attempting to erase an uncataloged data set.

Parameters:

`volid:`	For MVS uncataloged data sets, the volume serial number of the volume on which the file resides.
`fileid`	The fully qualified fileid of the file to be erased.

Examples:

```
erase OEM001:cbl.cbli.test.file
```

NEXTMAINWINDOW Command

Syntax:

```
>>-+- NEXTMAINWINDOW -+-><
   |                  |
   +- NMW ------------+
```

Description:

This command sets the focus window to the next main window i.e. one that is an immediate child of the desktop window, e.g. instances of CBLe, SELCOPY Interactive, CBLVCAT Interactive or any list windows/dialogs created directly from the desktop menu bar or command line.

The ring is maintained in creation sequence and wraps round from the first created to the last created.

See Also:

PREVMAINWINDOW
MDIPREV
MDINEXT

PREVMAINWINDOW Command

Syntax:

```
>>-+- PREVMAINWINDOW -+-><
   |                  |
   +- PMW ------------+
```

Description:

This command sets the focus window to the previous main window i.e. one that is an immediate child of the desktop window, e.g. instances of CBLe, SELCOPY Interactive, CBLVCAT Interactive or any list windows/dialogs created directly from the desktop menu bar or command line.

The ring is maintained in creation sequence and wraps round from the first created to the last created.

Because MDI applications such as the CBLe editor and SELCOPY Interactive have many child windows of their own (navigable with MDINext/Prev commands), this command is necessary to switch directly between CBLi applications.

By default, PF09 is set to **MDINext**.
By default, PF21 is set to **PrevMainWindow** (Shift-PF9)

See Also:

NEXTMAINWINDOW
MDIPREV
MDINEXT

SELCOPY Command Parameters Enhanced

Syntax:

```
>>-- Selcopy ------------------------------------------------------><


>>-- Selcopy --- -CTL selcctl --+------------------+-------------->
                                |                  |
                                +-- -LIB libpath --+

 >---+-----------------------------------------------------------+----><
     |                                                           |
     |             +- -DLI -+                                    |
     |             |        |                                    |
     +- -PSB psbn -+--------+-+--------------+-+--------------+
                   |        | |              | |              |
                   +- -BMP -+ +- -IMSID ssn -+ +- -AGN grpn -+
```

Description:

The SELCOPY command is used to start the SELCOPY Interactive application.

The SELCOPY Interactive main window is opened and the specified SELCOPY control statement file (selcctl) is loaded.

CBLi 1.30 contains the following changes to parameters on the SELCOPY command:

Parameters:

`-CTL selcctl` The full fileid of the file containing the SELCOPY control statements.

Alternatively, **selcctl** may now also be a previously defined DD/FILEDEF/DLBL which references a single data set or concatenation of data sets. In this case, the data in the SYSIN/SYSIPT edit view is read-only.

A token of 8 or fewer characters with no embedded dots is automatically treated an allocated filename.

Note that SELCOPY control statement input from members held in CA-LIBRARIAN libraries on an MVS system, is possible by allocating the selcctl ddname using CBLi's ALLOC command with SUBSYS(LAM).

`-LIB libpath` Introduced in CBLi 1.30 for **MVS** SELCOPY only, -LIB specifies a list of load libraries to be included before the current environment's search library chain. This is equivalent to supplying a JCL STEPLIB statement in a batch job and so may be used to control which SELCOPY module is executed and also any routines executed via the SELCOPY CALL statement.

Libpath may be one of the following:

- A DDname which has been pre-allocated to one or more load libraries.
- One or more load library DSNs separated by '`,`' (commas), '`;`' (semi-colons) or '` `' (blanks). Note that if blanks are used, quotes must also be used to delimit the list of DSNs, not the individual DSNs.

Examples:

```
SELCOPY  -CTL  SYSIN
```
Start SELCOPY Interactive using control statements in the currently allocated filename SYSIN.

```
SELCOPY  -CTL  SYS3.CBL.SELCOPY.CTL001   -LIB  "DEV.CBL.LOAD DEV.TEST.RTN001.LOAD"
```
Start SELCOPY Interactive using control statements in data set SYS3.CBL.SELCOPY.CTL001. Search load libraries DEV.CBL.LOAD then DEV.TEST.RTN001.LOAD ahead of the search chain when locating the SELCOPY executable module and any modules called using the SELCOPY CALL operation.

TASK Command

Syntax:

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

Description:

For MVS only, use the TASK command to start a program as a sub-task of CBLi.

TASK commands are generated by the CBLe REXX macro, JCLCMX, to run non-SELCOPY job steps of an MVS batch job in the environment in which CBLi is being executed (i.e. TSO or VTAM).

Parameters:

`pgmname`	The name of the program load module to be executed.
`-LIB libpath`	A list of load libraries to be included before the current environment's search library chain. This is equivalent to supplying a JCL STEPLIB statement in a batch job and so may be used to define the location of the program module to be executed plus any modules called by the program. Libpath may be one of the following: • A DDname which has been pre-allocated to one or more load libraries. • One or more load library DSNs separated by `','` (commas), `';'` (semi-colons) or `' '` (blanks). Note that if blanks are used, quotes must also be used to delimit the list of DSNs, not the individual DSNs.
`-PARM parm`	Parameter string to be passed to the program. This is equivalent to supplying the PARM parameter on an JCL EXEC statement in a batch job. If the parm string contains blanks, then quotes must be used to delimit the parm string.

Examples:

```
TASK  TRSMAIN  PARM='UNPACK'
```
Start program TRSMAIN to unpack a tersed data set.
Relevant INFILE and OUTFILE ddnames must be allocated before executing this command. (See the CBLe command ALLOCATE.)

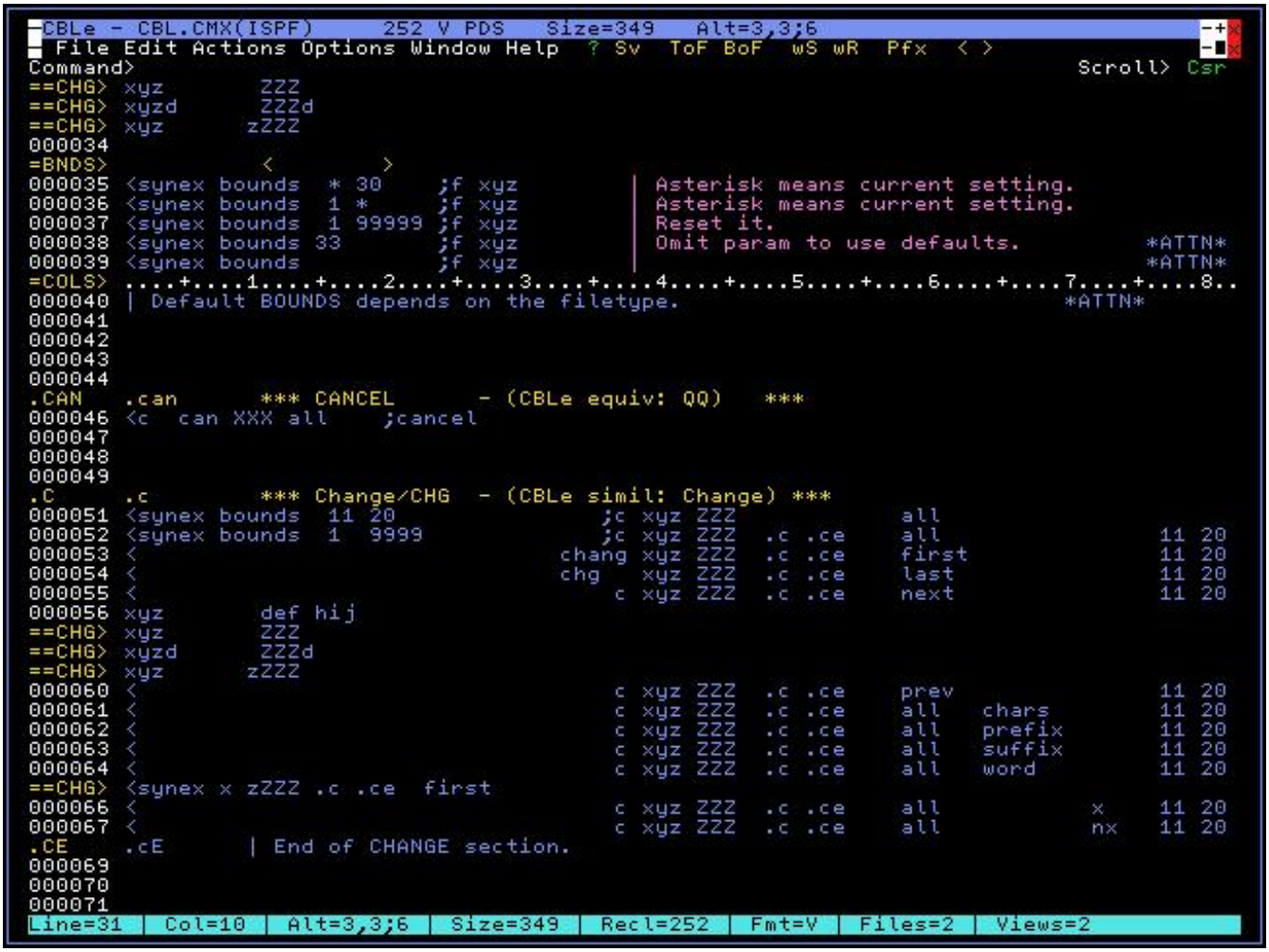
```
TASK  MYPROG  -LIB "SYS7.DEV.MYLIB.LOAD  SYS4.USER.ROUTINES.X01323"
```
Include the specified libraries at the start of the load library search chain then execute program MYPROG.

Section 04: CBLe Text Editor

The CBLe ISPF Edit Interface

The CBLe editor now has two operating interfaces, CBLE and ISPF.

- The CBLe interface is the classic CBLe operational mode that is based on IBM's CMS XEDIT and Mansfield's PC Kedit.
- The ISPF interface is an operational mode based on IBM's MVS ISPF Edit.



ISPF Interface Features

Wherever possible, the CBLe ISPF Interface is intended to mirror functionality included in the ISPF editor.

Although not all features of ISPF Edit are included, the CBLe ISPF interface supports the following:

1. **ISPF and ISPF Edit Primary Commands.**

AUTOSAVE	**COPY**	**HEX**	**RIGHT**
BNDS	**CREATE**	**LEFT**	**SORT**
BOTTOM	**DELETE**	**LOCATE**	**TOP**
BOUNDS	**DOWN**	**MOVE**	**UP**

CANCEL	**END**	**RCHANGE**	**X**
CAPS	**EXCLUDE**	**REPLACE**	
CHANGE	**FIND**	**RESET**	
CHG	**FLIP**	**RFIND**	

See CLI Command Precedence below.

2. **ISPF Edit Line Commands.**

(	**BNDS**	**I**	**R**
((	**BOU**	**L**	**RR**
)	**BOUND**	**LC**	**S**
))	**BOUNDS**	**LCC**	**TF**
<	**C**	**LCLC**	**TS**
<<	**CC**	**M**	**UC**
>	**COL**	**MM**	**UCC**
>>	**COLS**	**MASK**	**UCUC**
A	**D**	**O**	**X**
B	**DD**	**OO**	**XX**
BND	**F**	**R**	

In addition to these standard ISPF line commands, the ISPF interface also supports the following CBLe prefix area commands:

"	/	**MB**	**SCALE**
""	**HEX**	**ML**	**SJ**

3. **ISPF Edit Display Fields.**

   - The ISPF edit enterable SCROLL field with valid entries: 0-9999, CURSOR, DATA, HALF, MAX and PAGE.
   - Line labels.
     Note that the SETPT REXX macro may be used to set multiple labels corresponding to line markers in the file's text.
   - Line command (prefix) area flagged text for changed lines (==CHG>), error lines (==ERR>) and special lines for boundary definition (=BNDS>) and column identification (=COLS>).

4. **ISPF PFKey/command line concatenation.**

   The contents of the edit view command line are concatenated to the PFKey definition. The result is executed as a single command.

   As for ISPF, the caveat exists whereby the PFKey definition is a scroll command and the concatenation of the command line contents results in an invalid scroll command. In this case, if the word following the PFKey scroll command definition begins with a non-numerical character, the contents of the command line are executed prior to the PFKey scroll function.

The functionality provided by each of these features are documented fully in IBM manuals "ISPF: Edit and Edit Macros", "ISPF: User's Guide Vol 1" and "ISPF: User's Guide Vol 2".

Interface Initialisation

The current interface is defined via the following:

1. The CBLiINI variable EDIT.INTERFACE may be set in either the SYSTEM or USER CBLiINI file. e.g.

```
(Edit)
   Interface=ISPF       * ISPF or CBLe.
```

   This method defines the default interface for any new invocation of the CBLe editor.

   If the CBLiINI variable EDIT.INTERFACE is not set, then INTERFACE=ISPF is the default on MVS type systems, whereas INTERFACE=CBLe remains the default for VM/CMS and VSE.

2. The CBLe CLI command SET INTERFACE, allows the user to define the interface type to be used at the specified level. Supported levels are:

VIEW	The current edit view only.
FILE	All new and existing edit views of the current file.
GLOBAL	All new and existing edit views.

   The parameter INI may also be specified to intitialise the edit view(s). This redefines the PFKeys to be the defaults for the specified interface and also updates the appearance of the edit view (e.g. SCROLL field on for ISPF). e.g.

```
SET INTERFACE ISPF VIEW INI
```

CLI Command Precedence

The new set of ISPF interface primary commands are supported in addition to the existing CBLe CLI commands.

Where a command verb is associated with only one of the command interfaces (INTERFACE=CBLE or INTERFACE=ISPF), then it may be used with either interface without conflict. e.g. The CBLe CLI command, ALL, may be executed when in INTERFACE=ISPF.

The following are commands or SET options that exist in both command interfaces.

AUTOSAVE	**CREATE**	**LOCATE**	**TOP**
BOTTOM	**DOWN**	**MOVE**	**UP**
CANCEL	**FIND**	**REPLACE**	**X**
CHANGE	**HEX**	**RESET**	
COPY	**LEFT**	**RIGHT**	

When executing one of these commands, then, by default, the function and syntax of the command will be that associated with the active command interface. e.g. If INTERFACE=ISPF is active, the ISPF primary command, CHANGE, would be executed instead of the CBLe CLI command, CHANGE.

The active command interface may be temporarily overridden by prefixing the command with ICommand or ECommand. ICommand passes the command to the ISPF command interface, ECommand to the CBLe command interface.

The following command will use the CBLe version of the CHANGE command when INTERFACE=ISPF is active. (Upper case characters in the keywords indicate minimum abbreviation.)

```
ECommand Change /ABC/DEF/ * *
```

CBLe Environment Variables

Types of Variables

CBLe environment variables may be one of the following four types:

1. User defined environment variables set via the CBLe CLI EDITV command. Use the REXX macro, EQU, for a simple method of setting user environment variables.
2. Standard environment variables, as follow:

VarName	Description
user	User name as reported by 'Query USERname'
datetime **dtme**	Current date and time in format 'yyyy/mm/dd hh.mm'. e.g. 2006/08/09 23.59
timestmp	Current date and time in format 'yyyymmddhhmnss'. e.g. 20060809235942
date	Current date in format 'yyyy/mm/dd'. e.g. 2006/08/09
yyyy	Current 4 digit year. e.g. 2006
yy	Current 2 digit year. e.g. 06
mm	Current 2 digit month. e.g. 08
dd	Current 2 digit day of month. e.g. 09
ddd	Current 3 digit day of year. e.g. 221
time	Current time in format 'hh:mm:ss'. e.g. 23:59:42
tme	Current time in format 'hh.mm'. e.g. 23:59
hh	Current 2 digit hour of day. e.g. 23
mn	Current 2 digit minute of hour. e.g. 59
ss	Current 2 digit second of minute. e.g. 42
fi **fid**	Current file's fileid as reported by 'Query FIleid'
fm	Current file's filemode as reported by 'Query FMode'
fp	Current file's filepath as reported by 'Query FPath'
fn	Current file's filename as reported by 'Query FName'
ft	Current file's filetype as reported by 'Query FType'
ds **dsn**	Current file's DSname as reported by 'Query DSn'

3. MVS system symbols. e.g. &SYSNAME.
4. INI variables that have been explicitly set in the SYSTEM or USER CBLiINI files.

   These types of variables have the form "SYSTEM.section.varname" and "USER.section.varname" where:

SYSTEM **USER**	Variable as defined in the SYSTEM or USER CBLiINI file.

section	Section within the CBLiINI file. e.g. SYSTEM, EDIT, SELCOPY, RACF, etc.
varname	Name of the variable applicable to the relevant section of the CBLiINI file. e.g. InitialSize, CmdText, ProgramName, etc.

e.g. SYSTEM.CBLVCAT.SVC, USER.EDIT.SizeWarning, SYSTEM.HELP.DefaultPath

Variable Substitution

Prior to CBLi 1.30, variables could only be referenced in CBLe REXX macros.

CBLi 1.30 supports use, and subsequent translation, of CBLe environment variables specified within any command string that is executed from a CBLe edit view. This includes:

1. Commands in an edited file (typically a CMX file) that are executed using the CMDTEXT facility.
2. Commands executed from the CBLe edit view command prompt.

The CBLe CLI command, SET ENVVAR, has been introduced to switch CBLe variable substitution ON or OFF, and to define the variable delimiter character. By default, variable substitution is switched on with '%' (percent - X'6C') as the variable delimiter character.

By default, the CBLe edit variables, **user** and **system.edit.macropath**, will be translated in the following command.

```
set MACROPath   %user%.CBLE.MACROS   %system.edit.macropath%
```

The following example of concatenated command strings could be saved in your command (CMX) file for execution via CMDTEXT (PF4), to output a lists of library members updated by your userid today. Note that ';' (semi-colon - X'5E') is the command separator character.

```
<LL %user%.cbli.cble ; WHERE USER = %user% & LASTMOD => '%date%'   \
;ll %user%.cbli.cmx  ; WHERE USER = %user% & LASTMOD => '%date%'
```

The prevailing SET ENVVAR status (ON/OFF) may be temporarily overridden by prefixing the command with VIgnore or VRespect. VIgnore bypasses variable translation and VRespect performs variable translation regardless of the SET ENVVAR status.

The following command could be issued from a CBLe REXX macro to temporarily bypass variable translation when ENVVAR is ON. (Upper case characters in the keywords indicate minimum abbreviation.)

```
VIgnore Input '<ld %&SYSNAME%.Z16.**   | List system data sets.'
```

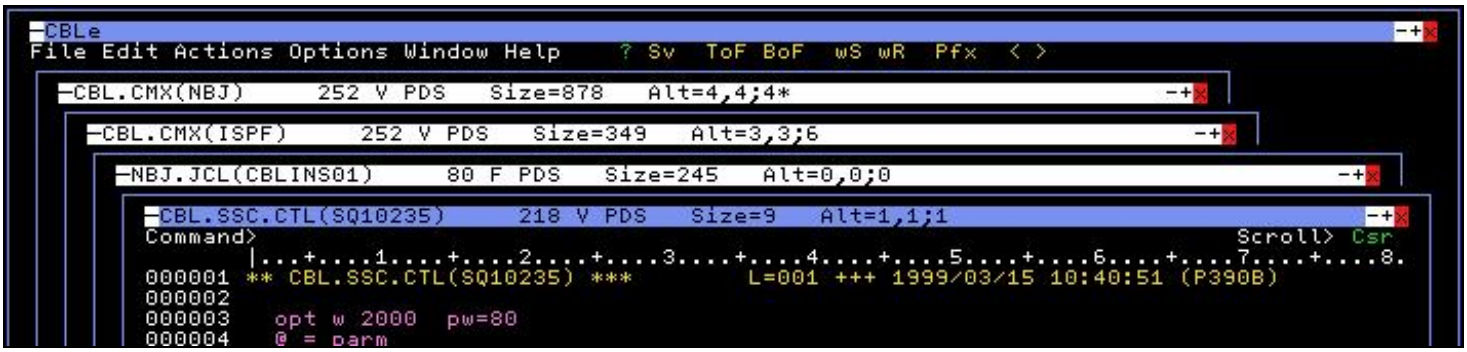
The LISTDATASET command string, beginning '<ld ', will be inserted following the focus line of the edited data with %&SYSNAME% unchanged.

Window Title Bar File Size

Edit views now display the file size (number of records) in the title bar. This is in addition to displaying the size of the file in the focus edit view window, in the information bar.

This has the benefit that, if the title bar of a non-maximised edit view is visible, the file size of an edited file may be determined without having to make it the focus window.

Introduced in CBLi 1.20, the alteration count information is already displayed in the title bar.



Section 05: CBLe CLI Commands

CANCEL Command

Syntax:

```
>>-- CANCEL --------------><
```

Description:

CANCEL causes CBLe to issue the QUIT command for all edited files in the edit ring.

If a file has been changed and not yet saved to disk (i.e. ALT is set to a value >0) then CBLe opens a dialog window asking whether you want to save the changes to the file before it is removed from memory.

Parameters:

None.

See Also:

QUIT

ECOMMAND Command

Syntax:

```
>>-- ECommand -- command -------><
```

Description:

Where a command exists in both the CBLE and ISPF edit command sets and INTERFACE=ISPF is active, ECOMMAND may be used as a prefix in order to force the use of the CBLe version of the command.

If INTERFACE=CBLE is active, the ECOMMAND prefix has no effect.

Parameters:

`command`	Any CBLe command followed by its parameters.

Examples:

```
ECommand Change /ABC/DEF/ * *
```
The CBLe version of the CHANGE command is used regardless of INTERFACE=ISPF.

See Also:

ICOMMAND

EXTRACT Command Options Added

Syntax:

```
>>-- EXTract --+-- option --+--><
               ^            |
               |            |
               +------------+
```

Description:

The EXTRACT command allows the user to extract information about the current edit environment for use in CBLe macros. The values for each EXTRACT option are assigned to REXX stem variables.

Where the definition of an option may be set by the user, a more detailed description of that option may be found under the relevant SET command.

Parameters:

`option`	Each EXTRACT option must be separated with a delimiter which may be any non-alphanumeric character, including blank.

The following EXTRACT options have been introduced in CBLi 1.30.

CMDText	`cmdtext.0`	3
	`cmdtext.1`	The text of the command that would be either executed or placed on the command line had the CMDTEXT command been issued with respect to the current focus column and line.
	`cmdtext.2`	'<' indicating that the command should be executed immediately, or '>' indicating that the command should be placed on the command-line.
	`cmdtext.3`	The placement position of the cursor within the resulting command had it been put on the command line, as defined by the presence of the 1st underscore character (_) in the original source field. Note that the first underscore is removed from the resulting command. If no underscore is present then cmdtext.3 will be 0.
COLOr	`color.0`	Number of items reflecting the SET COLOUR commands in effect for the current file.
	`color.i`	Each SET COLOUR item, field name in uppercase, followed by its colour and extended highlighting code. See SET COLOUR.
COLOur	`colour.n`	Same as COLOR.
ENVVars	`envvars.0`	2
	`envvars.1`	ON|OFF
	`envvars.2`	ENVVAR delimiter character.
INIVARS	`inivars.n`	Same as INIVAR
INTERFace	`interface.0`	1
	`interface.1`	CBLe|ISPF
LISTFILEACTion	`listfileaction.0`	1
	`listfileaction.1`	BROWSE|ERASE|NONE
Point	`point.0`	0 if focus line is not a named line; otherwise, 1.
	`point.1`	Line number and names of the focus line, if focus line is named.
Point*	`point.0`	Number of named lines.
	`point.i`	Line number and names of the ith named line.
RESERved	`reserved.0`	0 if no reserved lines; otherwise 1.
	`reserved.1`	List of reserved line numbers, if any.

ICOMMAND Command

Syntax:

```
>>-- ICommand -- command -------><
```

Description:

Where a command exists in both the CBLE and ISPF edit command sets and INTERFACE=CBLE is active, ICOMMAND may be used as a prefix in order to force the use of the ISPF version of the command.

If INTERFACE=ISPF is active, the ICOMMAND prefix has no effect.

Parameters:

`command` Any CBLe command followed by its parameters.

Examples:

```
ICommand Change abc 'DEF' all
```
The ISPF version of the CHANGE command is used regardless of INTERFACE=CBLE.

See Also:

ECOMMAND

LIST Command Options Added

Syntax:

```
>>-- LIst -- listtype -- /listparms/ --+------------------+--+---------+-->
                                       |                  |  |         |
                                       +- STEM stemname -+  +- STRIP -+
                                       |                  |
                                       +- FILE filename -+
    +- Lines ---+
    |           |
 >--+-----------+----+-------------------------+-----><
    |           |    |                         |
    +- Columns -+    +- SUBset /selectclause/ -+
```

Description:

The LIST command allows the user to extract rows of data returned from various CBLi LIST type commands and either place the output in a temporary edit view or assign the fields to REXX stem variables for use in a CBLe REXX macro.

`listtype` The CBLi list type function to extract.
The following listtype options have been introduced in CBLi 1.30.

`FS` / `FSEARCH` Although documented in CBLi 1.20 new features, FS was supported but FSEARCH was not.

`RETRIEVE` The list of commands available via the RETRIEVE command, which, by default, is assigned to PF12.

QUERY Command Options Added

Syntax:

```
>>-- Query ------- option -----><
```

The QUERY command allows the user to query information about the current edit environment. The information is displayed on the message line.

Where the definition of an option may be set by the user, a more detailed description of that option may be found under the relevant SET command.

Parameters:

`option`	The option for which information is returned.

The following QUERY options have been introduced in CBLi 1.30.

CMDText	Displays the text of the command that would be either executed or placed on the command line had the CMDTEXT command been issued with respect to the current focus column and line.
COLOr **COLOur**	Displays all the current colour settings, as defined by SET COLOUR, for fields in the focus edit view.
ENVVars	Displays the current setting for ENVVAR (ON or OFF) and the current environment variable delimiter character.
INIVARs	Displays the current system and user CBLIINI file variables and their values.
INTERFace	Displays the current setting for INTERFACE (CBLe or ISPF).
LISTFILEACTion	Displays the current setting for LISTFILEACTION (BROWSE, EDIT or NONE).
RESERved	Displays the line numbers of all reserved lines in the current edit view.

TFIND Command

Syntax:

```
>>-- TFind --+-----------------+--><
             |                 |
             +- string-target -+
```

Description:

Locate the first line from the focus line to contain the specified line-target in the 1st position of the current ZONE (BOUNDS) and make this line the new focus line.

If WRAP is set ON and the line-target is not found before bottom-of-file or top-of-file is encountered, then the scan continues from the opposite extreme of the file up to and including the focus line.

Where string-target is not specified, CBLe repeats the last TFIND command issued.

Parameters:

`string-target`	Line-target condition.

Examples:

`TF "//} DD"`
Find next DD card in MVS JCL, where 'ARBchar ON }' and default 'ZONE 1 *' are in effect.

`TFIND ~/  /`
Find next line not beginning with 3 blanks.

`TF /++/ | /###/`
Find next line beginning with either 2 plus or 3 hash signs.

See Also:

LOCATE
SET ARBCHAR
SET WRAP
SET ZONE

VIGNORE Command

Syntax:

```
>>-- VIgnore -- command -------><
```

Description:

When ENVVARS is set ON, VIGNORE may be used as a prefix to a command string to temporarily set ENVVARS OFF for the duration of the command execution.

Note that if the command executed is a CBLe REXX macro, then variable translation is suspended for all commands within the macro unless they are prefixed with VRESPECT or ENVVARS is explicitly set back ON.

If ENVVARS is set OFF, the VIGNORE prefix has no effect.

Parameters:

`command`	Any command followed by its parameters.

Examples:

```
VIgnore   Change   /%user%/%system%/ 1 *
```
All occurances of the literal string "%user%" are changed to the string "%system" on the focus line.

```
VIgnore   EQU    LOADLIB   %PREFIX%.LOAD
```
Macro EQU sets user environment variable "LOADLIB" to the literal string "%PREFIX%.LOAD".

```
VRespect   LL    %LOADLIB%
```
List library members belonging to "%PREFIX%.LOAD" where "%PREFIX" may be set to different values for separate invocations of this LL command.

See Also:

EDITV
VRESPECT
SET ENVVARS

VRESPECT Command

Syntax:

```
>>-- VRespect -- command -------><
```

Description:

When ENVVARS is set OFF, VRESPECT may be used as a prefix to a command string to temporarily set ENVVARS ON for the duration of the command execution.

Note that if the command executed is a CBLe REXX macro, then variable translation will be obeyed for all commands within the macro unless they are prefixed with VIGNORE or ENVVARS is explicitly set back OFF.

If ENVVARS is set ON, the VRESPECT prefix has no effect.

Parameters:

`command`	Any command followed by its parameters.

Examples:

```
VRespect  Change   /%user%/%system%/ 1 *
```
Both variables "user" and "system" are substituted and all occurances on the focus line of the value substituted for "user" are changed to the the value substituted for "system".

```
VRespect  EQU    LOADLIB   %PREFIX%.LOAD
```
The current value of variable "prefix" is replaces "%PREFIX%" in "%PREFIX%.LOAD" and macro EQU sets user environment variable "LOADLIB" to the resolved string.

See Also:

EDITV
VIGNORE
SET ENVVARS

WINDOWCOMMAND Command

Syntax:

```
>>-+- WINDOWCOMMAND -+-- command ---><
   |                 |
   +- WINCMD --------+
```

Description:

Execute a CBLi windows management command.

The WINDOWCOMMAND command has been introduced to enable window manipluation commands to be issued from CBLe REXX macros.

Examples of CBLi windows commands are MOVEWINDOW, SIZEWINDOW and the SQL style SELECT. See the supplied macro LDX for an example of its use

Parameters:

`command`	Any CBLi command followed by its parameters.

Examples:

```
WinCmd mw x=3 y=26
WinCmd szw w=72 d=31
WinCmd select entry,org,lrecl,recfm,blksz,referenced,vol,pdse,created
```

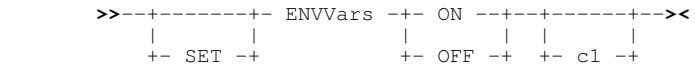
See Also:

CBLI
SYSCOMMAND

Section 06: CBLe SET Commands

SET ENVVARS Command

Syntax:



Description:

ENVVARS (Environment Variables) determines whether translation of CBLe environment variables occurs in commands issued from CBLe edit view windows. This includes commands executed from the command prompt or from within an edited file (typically a CMX file) via the CMDTEXT facility.

ENVVARS also defines the character used to delimit variable names.

CBLe environment variables include standard system determined variables, MVS System symbolics, EDITV variables and CBLiINI vaiables.

SET ENVVARS takes effect at the View level.

Parameters:

ON OFF	ON and OFF switches translation on and off respecively. If ENVVARS is OFF, command string will be passed without translating variables. Default is ON.
c1	The delimiter character. If c1 is not specified, the delimiter character last defined in the current CBLe view, is unchanged. Default is "%" (percent - X'6C'.)

Examples:

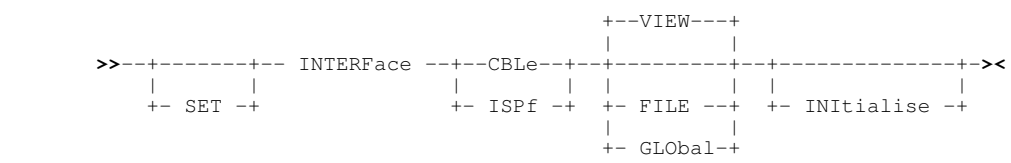
```
envv on  #
```
Set the variable delimiter character to "#" (hash).

See Also:

EDITV
VIGNORE
VRESPECT

SET INTERFACE Command

Syntax:



Description:

Sets the CBLe file editor interface to either CBLE or ISPF mode.

The CBLE interface is compatible with IBM's CMS editor XEDIT and Mansfield Software's PC adaptation KEDIT.

The ISPF interface is compatible with IBM's TSO ISPF editor and includes support for the most commonly used primary and line (prefix) commands, scrolling and PF key/command line concatenation.

INTERFACE=ISPF is the default on MVS type systems.
INTERFACE=CBLe is the default for VM/CMS and VSE.

The interface can be set globally, at the file or view level, allowing different interfaces for different files and even different views of the same file.

The edit interface can also be set in the SYSTEM and/or USER CBLiINI files, using:

```
(Edit)
   Interface=ISPF       * ISPF or CBLe.
```

Parameters:

CBLE ISPF	Set the current edit interface to CBLE (XEDIT/KEDIT style) or ISPF.
VIEW FILE GLOBAL	Set for the level at which the SET INTERFACE command will take effect.

VIEW	The current edit view only.
FILE	All edit views of the current file.
GLOBAL	All current and any new edit views.

INITIALISE	Initialise the default display area fields that correspond to the particular edit environment. This uses system defined defaults and and any CBLiINI option overrides. e.g. For ISPF, displays a SCROLL field in the command line. For CBLe, displays a reserved scale line as the first line of the edit display area.

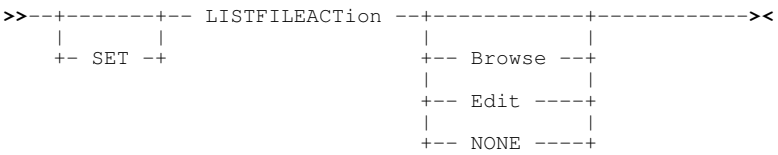
Examples:

`INTERFace CBLe`
Set the CBLe edit mode for the current file edit view.

`INTERFACE ISPF FILE INI`
Set the ISPF edit mode for all edit views of the current file and initialise their edit display areas to defaults associated with the ISPF interface.

SET LISTFILEACTION Command

Syntax:



Description:

Sets the default action when <Enter> is hit on an entry in a List window.

SET LISTFILEACTION overrides the default set in the SYSTEM and/or USER CBLiINI file by the (System) ListFileAction option.

Although a CBLe command, SET LISTFILEACTION not only affects list windows opened as child windows of the current CBLe session, but also list windows opened from anywhere within CBLi.

SET LISTFILEACTION takes effect at the Global level.

Parameters:

Browse	If the entry is a CMS fileid, VSE LIBR member name, MVS DSN of a sequential or VSAM data set, or MVS PDS(E) member name, then hitting <Enter> will edit the entry Read-Only.
Edit	If the entry is a CMS fileid, VSE LIBR member name, MVS DSN of a sequential or VSAM data set, or MVS PDS(E) member name, then hitting <Enter> will edit the entry Read-Write.
NONE	No action will be taken on hitting <Enter> on any list window entry.

Section 07: CBLe Edit Macros

The following macros have been updated in CBLi 1.30 to provide compatiblilty with INTERFACE=ISPF and/or to fix minor bugs:

BOXSEQ, BOXTOT, CMXAMS, DELALL, DELNOT, EM, ERA, HD, JOBCARD, LDIFF, LM, SDBPOPUP, SDBTRACK, SDBWINX, SETPT, SV, TSOC, TrB, WINX,

The following new or updated CBLe REXX macros are also included in CBLi 1.30.

The detailed description for each of the following macros is documented in the macro member itself. Because macros are subject to change by users, only a brief description of macros supplied by CBL may be found in the CBLi Help documentation.

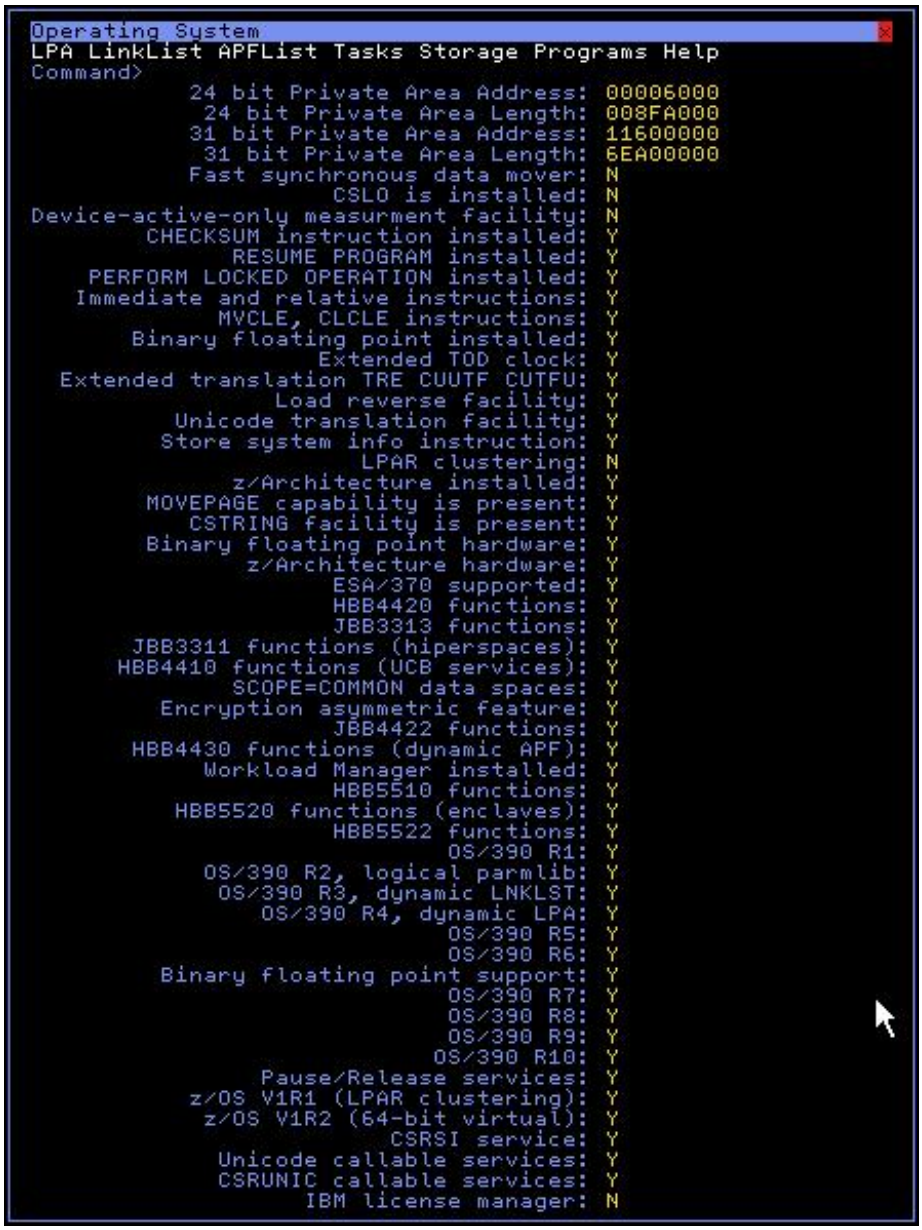
Macro Name	Status	Description
CBLIZAPL	New	Report all zaps that are applied to the CBLi module being executed.
CMDFUNC	New	CBLe utility functions for dataset point-and-shoot from: 1. MVS JCL DD, IDCAMS DEFINE and TSO/CBLe ALLOC statements. 2. Any file where the cursor is positioned on a DSN. Format: `CMDFUNC X | K | R` Where: **X** Edit a file. (Executed immediately) **K** Erase a file. (Command put on command line) **R** Rename a file. (Command put on command line) Ideally, where supported by the 3270 emulator, keyboard macros set on Ctrl-X/K/R should be created to issue the releveant CMDFUNC command.
CMEN	New	Generate a temporary CMX file containing all retrievable commands.
COLSET	Updated	Support for CMX, JCL, REXX, etc. keyword parmaters, in order to apply appropriate colouring schemes.
EINI	New	Edit the SYSTEM (Site) or USER CBLiINI file.
EQU	New	Set a CBLe user environment variable using EDITV.
ERASEALL	New	Create a CMX file containing an ERASE command for all files matching a supplied mask.
JCLCMX	Updated	JCL STEPLIB, JOBLIB and SET statements supported and bug fixes.
PROFIRST	Updated	Defines OP and OQ in storage macros for MVS access to SDSF Operator LOG and Output Queue respectively.
PROSITE	New	Provided as a template for Site wide overrides to default CBLE settings. Called by PROFILE.
PROUSER	New	Provided as a template for user overrides to default CBLE and PROSITE macro settings. Called by PROFILE.
RINGL	New	Display the current ring of files as a popup menu. A file may be selected from the menu and made the current file. C|U option displays changed files only.

Section 08: MVS Operating System Window

The Operating System window may be opened via the following:

- Select 'Operating System' from the System menu in the CBLi Main Menu
- Enter the CBLi command SYSInfo on the command line of any window.

In CBLi 1.30, the Operating System window for MVS systems also report the following SCCB configuration bits. (Page Down the Operating System window to display these settings.)



Section 09: SELCOPY Interactive

DDNAME Control Statement Input

The SELCOPY Interactive application window is started via the following:

- Select 'Execute SELCOPY' from the File menu in the CBLi Main Menu.
- Enter the CBLi command SELCOPY on the command line of any window.

A control file containing the SELCOPY source statements must be provided as SYSIN/SYSIPT input to the SELCOPY Interactive window either as a parameter on the SELCOPY command or via the "Run SELCOPY" dialog window.

For either of these methods, the control file may be specified as an explicit fileid (MVS DSN, CMS fileid or VSE LIBR memberid).

New in CBLi 1.30, the control statements may alternatively be supplied via an up to 8 character, previously allocated filename (DD/FILEDEF/DLBL) which may be a concatenation of data sets.

In the following example, the CBLe ALLOCATE command is first used to allocate a concatenation of two DSNs to DDname, INCTL. These commands would be entered in a CBLe edited CMX file and executed using CMDTEXT (PF4).

```
<alloc  dd(INCTL)  dsn('SYS3.NBJ.EQU001' 'SYS3.TEST.SELCOPY.CTL(SQ10249)')  shr
<selcopy   -ctl INCTL
```

SELCOPY Interactive opens a CBLe edit view for the control statement SYSIN/SYSIPT input which, if supplied as an explicit fileid (DSN) which is not locked (ENQ'd) by another process and the user has sufficient authority, is opened read-write.

Where control statement SYSIN/SYSIPT input refers to a previously allocated filename (DD/FILEDEF/DLBL), then the data is edited read-only.

LIBPATH Library Search Chain

The location of the SELCOPY program executed and any routines called by the SELCOPY operation, CALL, is determined by the standard search chain for the current environment.

Note: The SELCOPY CALL operation is used to pass control to an external Assembler, COBOL routine or any MVS program module developed using Language Environment.

For MVS systems only, SELCOPY Interactive in CBLi 1.30 provides users with the ability to include additional libraries to the start of the search chain. This gives the SELCOPY Interactive environment an equivalent to the STEPLIB JCL statement, which may occur in SELCOPY batch jobs.

The included library path may be entered in the "Run SELCOPY" dialog window or via the -LIB parameter on the CBLi CLI SELCOPY command, as one of the following:

- A DDname which has been pre-allocated to one or more load libraries.
- One or more load library DSNs separated by ',' (commas), ';' (semi-colons) or ' ' (blanks).

Note: If the DSNs are separated by blanks, quotes must also be used to delimit the list of DSNs, not the individual DSNs.

The following SELCOPY line command and "Run SELCOPY" dialog examples illustrate use of LibPath.

```
<selcopy   -ctl INCTL   -lib "SYS3.TEST.LOADLIB SYS3.NBJ.TEST.EXE"
```

This feature is exploited by the JCLCMX CBLe REXX macro which translates JCL statements from a batch job into the equivalent CBLi commands, thus setting up an environment suitable for interactive execution of a SELCOPY batch job.

Where STEPLIB JCL statements are found in EXEC PGM=SELCOPY steps, the library names are included on a -LIB parameter in the generated, equivalent SELCOPY command.

Section 10: PFKey Defaults

CBLe CLASS PFKeys

CBLi 1.30 support for INTERFACE=ISPF in the CBLe text editor has meant that default definitions for the CBLe CLASS group of PFKeys have been updated.

To avoid conflict with ISPF commands of the same name the ECOMMAND prefix has been added as follows:

PF17	ECommand copy block
PF18	ECommand move block
PF19	ECommand delete block
PF24	ECommand reset block

The following PFKeys have been defined as the defaults for the CBLe editor when INTERFACE=ISPF is in effect:

PF03	end
PF05	rfind
PF06	rchange
PF07	up
PF08	down

When INTERFACE=CBLe is in effect, these default PFKeys are defined as before:

PF03	quit
PF05	macro block up major
PF06	macro block down major
PF07	backward
PF08	forward

All other default PFkeys are common to both interfaces and are unchanged from the previous release of CBLi:

PF01	sos lineadd
PF02	duplicate
PF04	cmdtext
PF09	MDINext
PF10	left
PF11	right
PF12	retrieve -
PF13	sos linedel
PF14	spltjoin
PF15	mark box
PF16	mark line
PF20	overlaybox
PF21	PrevMainWindow
PF22	undo
PF23	redo

CBLi DEFAULT PFKeys

For all CBLi windows the DEFAULT group of PFKeys has been updated as follows:

PF09	NextMainWindow
PF21	PrevMainWindow

Section 11: CBLiINI Options

Support for the following SYSTEM and USER CBLiINI file options have been added:

```
(SYSTEM)
  CommandDelimiter=char
```

The command delimiter character used to separate multiple commands on any single, CBLi command line. This character also applies to CBLe edit views unless overridden by the SET LINEND command.

Default is the character defined by the CBLNAME module Separator parameter. Refer to the CBLNAME assembler source member in the &PREFIX..SRC install library.

```
CommandDelimiter=;
```

```
(SYSTEM)
  CmdText=fileid
  UserINIFile=user.cbliini.file
```

CmdText specifies the default command (CMX) file to be edited on startup of CBLi, whereas UserINIFile specifies the fileid of the USER CBLiINI file.

These options were supported in previous releases of CBLi but have been enhanced in CBLi 1.30 to allow use of embedded variables. These may be MVS system symbols and/or the special variable &USER or %USER% which gets substituted with the user's RACF userid. e.g.

```
CmdText=&SYSNAME..%USER%.CBL.CBLI.CMX
UserINIFile=&SYSNAME..DEV.&USER.CBLI.INI
```

```
(SYSTEM)
  ListFileAction=Browse|Edit|NONE
```

Set the default action on hitting <Enter> on an entry in a List window.

If the entry is an editable object (CMS fileid, VSE LIBR member name, MVS DSN or PDS(E) member name) then ListFileAction=Browse will cause the object to be edited read/only whereas ListFileAction=Edit will cause the object to be edited read/write.

If ListFileAction=NONE, then no action is taken when <Enter> is hit on any entry of a List window.

Default value is Edit.

```
(EDIT)
  INTERFACE=ISPF|CBLe
```

Defines the default CBLe edit interface. For MVS systems the program default is ISPF, for VM and VSE it is the original XEDIT/KEDIT compatible CBLe interface.

```
(ISPFEdit)
  PFnn=cmd
```

Define PFKey defaults for CBLe editor INTERFACE=ISPF editing. Setting PFKeys here will override the program defaults listed under "10.1 CBLe CLASS PFKeys".

Compare with PFnn settings under section (Edit) which apply to CBLe editor INTERFACE=CBLE editing.

```
(RACF)
  ResourceCheck=NO|YES
```

For MVS only, use RACF, or equivalent security package, for CBLi resource access checking. Default is NO.

```
(RACF)
  SuppressWTO=NO|YES
```

For MVS CBLi resource access checking, SuppressWTO controls whether access failure messages written to the console are to be suppressed. Default is NO.

```
(RACF)
  ResourceClass=ClassName
```

For MVS CBLi resource access checking, ResourceClass specifies the MVS RACF, or equivalent, general resource class. Default is FACILITY.

```
(RACF)
  System=ResourceName           * Default is CBLI.SYSTEM
  UserTSO=ResourceName          * Default is CBLI.USER.TSO
  UserVTAM=ResourceName         * Default is CBLI.USER.VTAM
  SELCOPY=ResourceName          * Default is CBLI.SELCOPY
  CBLVCAT=ResourceName          * Default is CBLI.CBLVCAT
  DB2=ResourceName              * Default is CBLI.DB2
```

For CBLi resource access checking, each of the above options specify the CBLi resource to which the MVS RACF, or equivalent, resource name is to be applied. The resource name must first be defined to the resource class specified in the ResourceClass option.

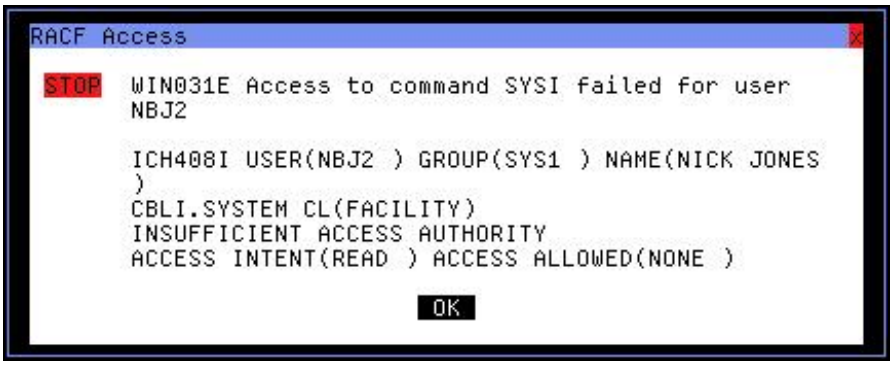
The CBLi resources are as follow:

System	CBLi System windows and commands.
UserTSO	CBLi user access under TSO.
UserVTAM	CBLi user access under VTAM.
SELCOPY	Interactive SELCOPY.
CBLVCAT	Interactive CBLVCAT.
DB2	Interactive SQL.

Section 12: Error Messages

CBLi Messages

The following popup message has been included in line with CBLi 1.30 support for MVS RACF, or equivalent, control over user access to CBLi and its facilities:



CBLe Messages

For the CBLe text editor only, when a command is issued to the host system (TSO or CMS) that gives a non-zero return code, then the following CBLe message is returned:

```
EDT043E RC=<nn> from: <command>
```

This replaces the usual CBLi popup message box:

Particularly for commands issued by CBLe macros, this has the advantage that a user response (i.e. hitting <Enter>) is not required.

The following CBLe text editor massages have been added:

```
EDT099E I/O error for file fileid.
EDT100E Line label "xxx" not found.
EDT101E Inconsistent keywords "xxx" and "yyy" in command zzz.
EDT102E Syntax is ambiguous. Use string delimiters (single or double quotes).
EDT103E Invalid hexadecimal string "X'nn'" in command xxx.
EDT104E A label is missing in the xxx command.
EDT105E No find string (string-1) given on the xxx command.
EDT106E No change string (string-2) given on the xxx command.
EDT107E Left bound (nn) + find string length(mm) is greater than LRECL(zz).
EDT108E No lines were found for the xxx command to search.
EDT109E Bottom of the range reached for xxx.
EDT110E Top of the range reached for xxx.
EDT111E No previous xxx command exists to be repeated.
EDT112E Left bound (nn) + find string length(mm) is greater than right bound(zz)
EDT113E Left bound (nn) + change string length(mm) is greater than right bound(zz).
EDT114E Change string is a picture string so must be same length as find string.
EDT115E Invalid special character x in change picture string.
EDT116E Duplicate keywords "xxx" and "yyy" in command zzz.
EDT117E Characters "xxx" found.
EDT118E Characters "xxx" changed to "yyy"
EDT119E Generic locate requires a line type (CHANGE, COMMAND, ERROR, EXCLUDED, LABEL or SPECIAL).
EDT120E Locate failed for "xxx".
```

EDT121E Invalid left bound value "*nn*".
EDT122E Invalid right bound value "*mm*".
EDT123E Left and right bounds cannot be equal.
EDT124E Left bound (*nn*) higher than right bound (*mm*).
EDT125E Left bound not specified.
EDT126E Right bound not specified.
EDT127E File name missing for the *xxx* command.
EDT128E Line pointer required for the *xxx* command.
EDT129E Invalid line pointer *xxx*.
EDT130E *xxx* command requires a target. Use the A or B prefix commands or specify a line pointer.
EDT131E *xxx* command requires a range. Use the C or M prefix commands or specify a line pointer range.
EDT132E The overlay target range overlaps the overlay source range.
EDT133E The target of a MOVE or COPY prefix command is in the source range.
EDT134E No type (CBLe or ISPF) specified for the SET INTERFACE command.
EDT135E CAPS mode changed from *OFF/ON* to *ON/OFF* as the file has *upper/mixed* case data.
EDT136E The *xxx* string cannot be the null string.
EDT137E No parameters specified for the *xxx* command.
EDT138E Invalid line number *nn*.
EDT139E No lines deleted by the command *xxx*.
EDT140E No lines sorted by the command *xxx*.
EDT141E Sort column *zz* is outside of the bounds (*nn*, *mm*).
EDT142E Too many sort fields. A maximum of *nn* is supported.
EDT143E Sort field (*xx*, *yy*) overlaps field (*nn*, *mm*).
EDT144E Stem name too long (>256 characters).
EDT145E Stem name *xxx.i* has invalid characters in it (not A-Z, a-Z, 0-9, £@#.!?).
EDT146E Stem name *xxx.i* begins with an invalid character (0-9 or .).
EDT147E Variable *xxx* contains a (nested) reference to itself in variable *yyy*.
EDT148E Edit *fileid* is read-only. *userid* is editing it.