



**SELCOPY 2.02 New Features  
for 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/selcdoc.html>**

# Contents

<b><u>SELCOPY 2.02 New Features</u></b> .....	<b>1</b>
<u>Documentation Notes</u> .....	1
<b><u>Section 01: Important Changes</u></b> .....	<b>2</b>
<u>CBL Interactive Environment for MVS, VSE and CMS</u> .....	2
<u>CBL Interactive Environment Features</u> .....	3
<u>Windowed Environment within the 3270 Display</u> .....	3
<u>Disaster Recovery Aid</u> .....	4
<u>Interactive SELCOPY execution</u> .....	4
<u>Interactive CBLVCAT execution</u> .....	4
<u>CBLc Text Editor</u> .....	5
<u>List Windows</u> .....	6
<u>System Information List Windows (MVS Systems only)</u> .....	7
<u>Dynamic SQL Interface to DB2 (MVS Systems only)</u> .....	8
<u>Dynamic AMS IDCAMS Execution</u> .....	9
<u>General Utilities</u> .....	10
<u>On-line Help System</u> .....	11
<b><u>Section 02: New Facilities</u></b> .....	<b>12</b>
<u>Interactive SELCOPY Debugger and Development Environment</u> .....	12
<u>SELCOPY SYSIN/SYSIPT Control Statements Window</u> .....	13
<u>Point-and-Shoot Options Popup Menu</u> .....	14
<u>SELCOPY SYSPRINT Output Window</u> .....	14
<u>Work Area and POS Storage Windows</u> .....	15
<u>TRACE Window</u> .....	17
<u>SQL Log Window</u> .....	17
<u>WTO Log Window</u> .....	18
<u>EQUates and @ Pointer Windows</u> .....	19
<u>Non-standard 3270 Terminal Displays</u> .....	19
<b><u>Section 03: Other Changes</u></b> .....	<b>21</b>
<u>SELCOPY 2.01 Zaps applied</u> .....	21
<u>SELCOPY 2.00 Zaps applied</u> .....	21
<u>SELCOPY Archive</u> .....	21

# SELCOPY 2.02 New Features

---

## Documentation Notes

Information in this New Feature List reflects differences between SELCOPY 2.01 and SELCOPY 2.02.

The **CBL Products Installation Guide** and **New Features** documents are available in Adobe Acrobat PDF format at CBL web page <http://www.cbl.com/cblvdoc.html>.

Copyright in the whole and every part of this document and of the SELCOPY 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 system and programs.

No reproduction of the whole or any part of the SELCOPY 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 manual.

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.

# Section 01: Important Changes

## CBL Interactive Environment for MVS, VSE and CMS

CBL software products, SELCOPY and CBLVCAT, are now packaged with an interactive environment (CBLi) that includes additional productivity tools and is a vehicle for interactive execution of each product. (See [SELCOPY Interactive Debugger and Development Environment](#) in **Section 2: New Facilities.**)

Use of CBLi is included, at no additional cost, within the SELCOPY and CBLVCAT licence agreements.

CBLi was first made GA at release 1.10 on 2004/12/06, and made available to all SELCOPY and CBLVCAT users to download as a separately installable product bundle from the [CBL web site](#).

CBLi is now integral to the SELCOPY and CBLVCAT products and is installed automatically as part of the latest SELCOPY and CBLVCAT install procedures.

OpSys	Environment	Startup Command
MVS	TSO/E	Enter the command <b>CBLi</b> at the READY prompt.
	ISPF	Enter <b>TSO CBLii</b> (double "i") on the ISPF command line.
	VTAM	Enter <b>LOGON APPLID(CBLiVTAM)</b> on a VTAM USS screen.
VSE	VTAM	Enter <b>LOGON APPLID(CBLiVTAM)</b> on a VTAM USS screen.
VM	CMS	Enter the command <b>CBLi</b> on the CMS command line.

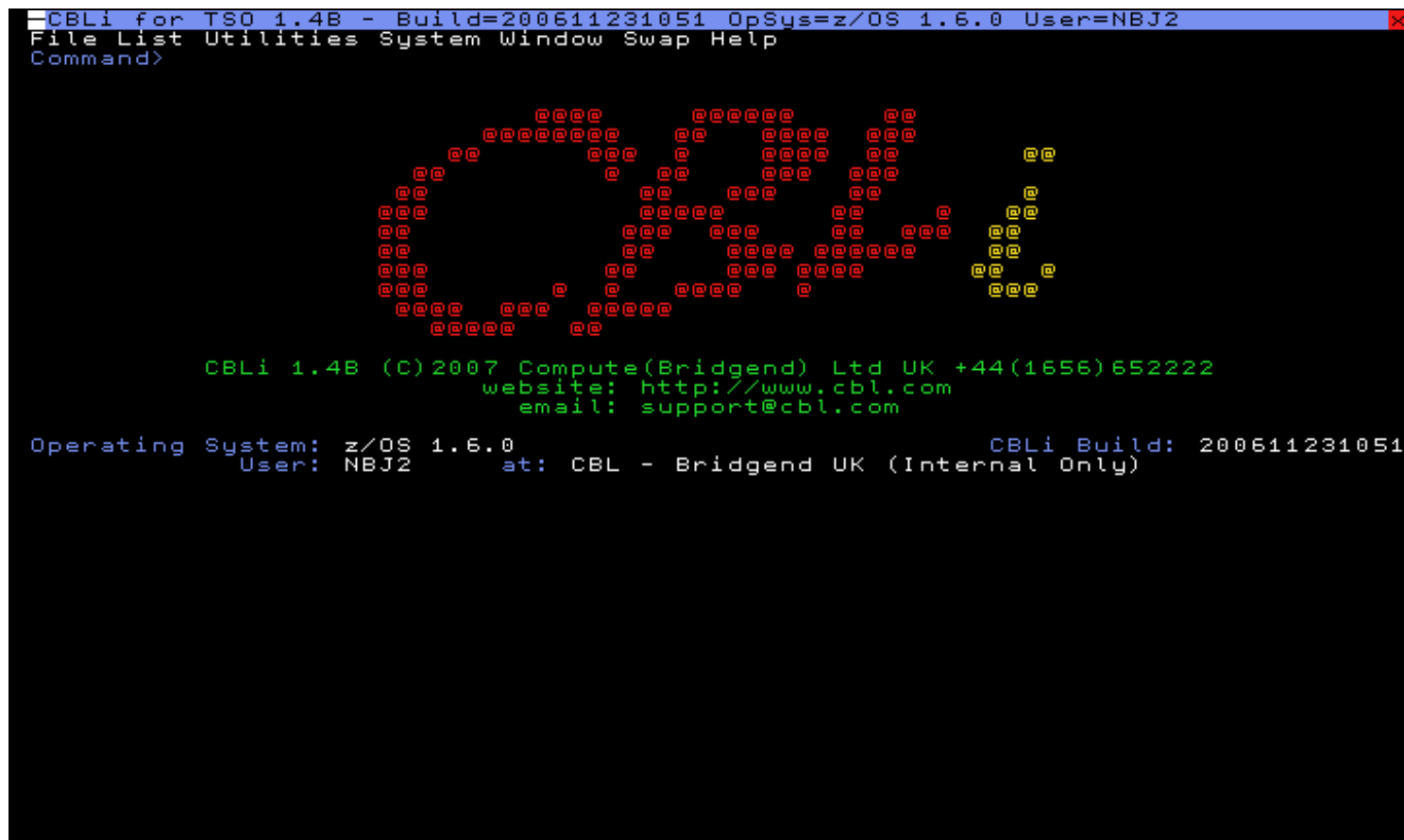


Figure 1. CBL Interactive Environment Main Window.

## CBL Interactive Environment Features

CBLi is a productivity aid designed to provide users with a powerful, intuitive, multi-windowed environment within the 3270 terminal display, in which to launch projects and manage files.

Detailed information on CBLi window operation and supported command reference is available in the [CBLi User Manual](#) and accompanying [CBLe Text Editor Manual](#).

Interactive Environment key features follow.

### Windowed Environment within the 3270 Display

This gives the user the ability to display information (edited views, lists, etc.) in any number of easily navigable windows concurrently. A significant improvement for those MVS users who are only able to swap between two, full-screen pages.

The windowed environment supports:

- Resizable and moveable windows that may be tiled, cascaded, minimised, maximised, restored and customised.
- Drop-down and pop-up menus.
- Drop-down Window List.
- Hot key in and out of ISPF split screens.

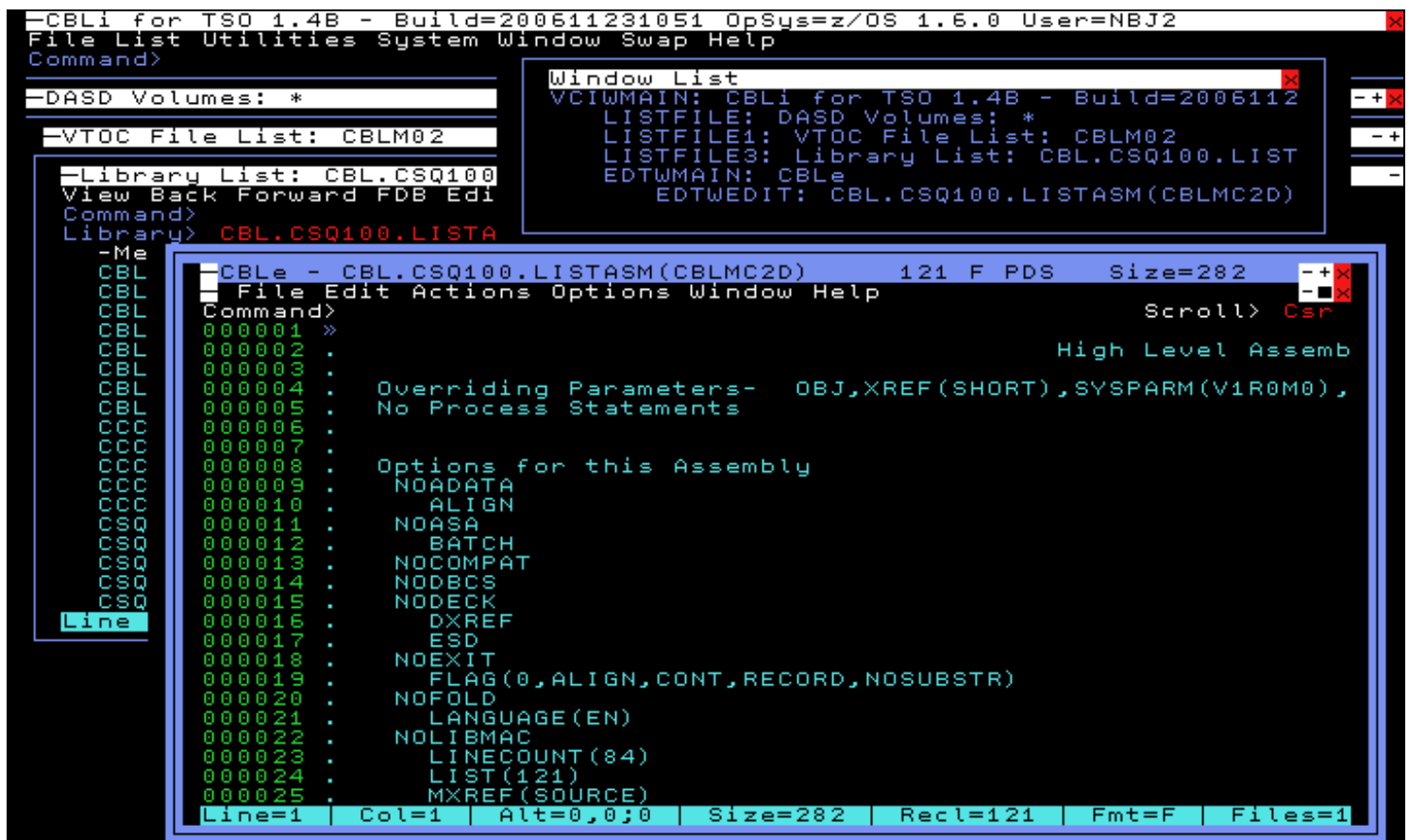


Figure 2. CBL Interactive Environment Window Resize.

## Disaster Recovery Aid

Not only does CBLi execute natively in CMS, TSO/E and as an ISPF application, but also as a VTAM application in MVS or VSE.

SELCOPY and CBLi VTAM installed on a DR volume would provide users with a working environment that includes system information windows, a text editor (CBLi) and job submission capabilities without the need for a MVS TSO or VSE ICCF.

### Interactive SELCOPY execution

Allows the user to debug SELCOPY job streams by stepping through control statements, setting break points and displaying changes to workarea storage, @ pointers, etc. This is discussed in more detail in [Section 2: New Facilities](#) below.

### Interactive CBLVCAT execution

Allows the user to generate dynamic reports of VTOC/Catalog contents and IDCAMS tuning recommendations for VSAM data sets.

```

-CBLi for TSO 1.4B - Build=200611231319 OpSys=z/OS 1.6.0 User=NBJ
File List Utilities System Window Swap Help
Execute CBLVCAT
View Back Forward FDB Edit Raw Help
Command>
VCAT Command> listcat key=nbj.cblidemo.v0000.ksds tune growth=0 avlrecl=2452
> define
>
VCAT Program> V210
-----SysPrint-----
ICF CAT CBLMCT (3390) TYPE NRECS PCNT ---- ALLOC TRACKS ---- FRSP
----- TOTAL PRIME SEC CI C

NBJ.CBLIDEMO.V0000.KSDS
KSDS 500 24.9 84 21 7*4
VOL1=CBLM05
IX 13 39.4 1 1 1
VOL1=CBLM05

*** SEV 3-06 *** CA SPLITS TOO HIGH (8 PC OF INSERTS)
*** SEV 3-08 *** CI SPLITS TOO HIGH (30 PC OF INSERTS)
** SEV 2-19 ** SEC EXTENTS TOO HIGH
** SEV 2-25 ** INEFFICIENT DATA CSIZE
** SEV 2-27 ** TUNING FOR RECORDS/AVLRECL CHANGE REQUEST
* SEV 1-15 * 3 CYLS CAN BE RECOVERED WHEN TUNED
* SEV 1-22 * SPEED NOT DEFINED - RECOVERY IS DEFAULT

*** WARN 016 *** LARGE ALLOC CHANGE

CBL TUNED
-----
DATA (
CISZ (10240) - * NEW PHYREC SIZE=10240, CURRENT=40
CYLINDERS (2,1) - * OPTIMISED FOR DEVICE GEOMETRY
RECORDSIZE (2452,4089) - * DEFINED AVLRECL=1024
BUFFERSPACE (24576) - * 24K MINIMUM FOR DIRECT PROCESSING
SPEED ) * DON'T ALLOW DEFAULT
*** WARN 019 *** SKELETON DECK ONLY - ATTENTION REQUIRED

Line 8 of 90 | Col 1 of 135 | Views 1 | select *

```

Figure 3. CBLVCAT Interactive File TUNE.

## CBLLe Text Editor

A function rich, user configurable text editor including support for:

- Interfaces for both **ISPF-Edit (PDF)** and **KEDIT/XEDIT** compatibility.
- MDI (Multiple Document Interface) Edit display windows.
- Command Line Interface (CLI) on all edit operations. (cf. ISPF Primary commands)
- Configurable PFKeys and Edit environment.
- Prefix area command support. (cf. ISPF Line commands)
- Text blocking as source for copy, move, delete, overlay, etc.
- Simple VSAM data set editing.
- REXX Edit Macro support, including EXTRACT feature to automatically create REXX stem variables.
- Automatic translation of System Symbols, environment and user-defined variables.
- ALLOC command (for when TSO is not available.)
- SUBMIT job streams to batch.
- TASK command to start a program as a sub-task.

CBLLe also includes the CMDTEXT function, a unique **point-and-shoot** facility enabling execution of any system (CMS/TSO/ISPF), CBLi or CBLLe text edit command stored as a text string in any editable file. This feature has brought about the use of **CMX** files (files containing related command strings and comment text) as "command-centres" for specific projects and tasks.

The screenshot displays the CBLLe Text Editor interface with the following components:

- Top Bar:** -CBLi for TSO 1.4B - Build=200611231319 OpSys=z/OS 1.6.0 User=NBJ
- Menu Bar:** File List Utilities System Window Swap Help
- Toolbar:** ? SV ToF BoF wS wR Pfx < >
- Window List:**
  - CBL.CMX(NBJ) 252 V PDS Size=878 Alt=0,0;1
  - CBL.CMX(ISPF) 252 V PDS Size=349 Alt=1,1;3
  - NBJ.TEST.KSDS 128 V KSDS(R) Size=6 Alt=0,0;0
- Main Edit Area:**
  - Command> |...+...1...+...2...+...3...+...4...+...5...+...6... Scroll> Csr
  - 000001 AAAAAAAAAA Record 1 length 80
  - 000002 AAAAAAAB Record 2 length 40
  - 000003 AAAAAAAC Record 3 length 66
  - 000004 AAAAAAAD Record 4 length 64
  - 000005 AAAAAAAE Record 5 leng
  - 000006 AAAAAAAF Record 6 leng
  - 000007 \* \* \* End of File \* \* \*
- Hex Edit Overlay (NBj.TEST.KSDS):**
  - RecNo> 2 Length> 40
  - 000000 C1C1C1C1 C1C1C1C2 AAAAAAAB
  - 000008 4040D985 83969984 Record
  - 000010 40F24093 859587A3 2 length
  - 000018 8840F4F0 40606060 h 40 ---
  - 000020 60606060 6060606E ----->
  - 000028 00000000 00000000
  - 000030 00000000 00000000
  - 000038 00000000 00000000
- Bottom Panel:**
  - 000046 <c can XXX all !cance
  - 000047
  - 000048
  - 000049
  - .C \*\*\* Change/CHG - (CBLLe simil: Change) \*\*\*
  - 000051 <synex bounds 11 20 !c xyz ZZZ all
  - 000052 <synex bounds 1 9999 !c xyz ZZZ .c .ce all
  - 000053 < chang xyz ZZZ .c .ce first
  - 000054 < chg xyz ZZZ .c .ce last
  - 000055 < c xyz ZZZ .c .ce next
  - 000056 xyz def hij
  - ==CHG> xyz ZZZ
  - ==CHG> xyzd ZZZd
  - ==CHG> xyz zZZZ
  - 000060 < c xyz ZZZ .c .ce prev
- Status Bar:** Line=1 Col=1 Alt=0,0;0 Size=6 Recl=128 Fmt=V Files=3 Views=3

Figure 4. CBLLe Text Editor.

## List Windows

Display configurable rows of information for various list types that include:

- DASD Volumes.
- VTOC contents.
- Cataloged Datasets.
- Library Members.
- Enqueued Resources.
- Allocated Datasets.

```

-CBLI for TSO 1.4B - Build=200611231642 OpSys=z/OS 1.6.0 User=NBJ
File List Utilities System Window Swap Help
Command>

-Dataset List: NBJ***
View Back Forward FDB Edit Help
Command>
Entry> NBJ***
Catalog> USERCAT.CBLCAT
Types>
-----Entry-----
NBJ.CBL.INST.SELCOPY.NAM
NBJ.CBL.INST.SRC
NBJ.CBL.INST.TEST03.PACK
NBJ.CBL.INST.TXT
NBJ.CBL.TEST.INSTDATA.PACK

-DASD Volumes: CBL*
View Back Forward FDB Edit Help
Command>
Volume> CBL*
UNIT -VOL-- FREECYL FREETRK FREEXTN FRE
0AA0 CBLMCT 2174 32625 5
0AA1 CBLM01 401 6069 15
0AA2 CBLM02 272 4143 15
0AA3 CBLM03 222 3376 12
0AA4 CBLM04 490 7395 11
0AA5 CBLM05 443 6668 8
0AA6 CBLM06 458 6894 7

-VTOC File List: CBLM06
View Back Forward FDB Edit Help
Command>
Volume> CBLM06
Filter> *.*
-Vol-- -----Dsn----- Org RecFm Lrecl Blksz Alu
CBLM06 CBL.CBLI.F5 PO VB 1024 32760 C
CBLM06 CBL.CBLI.HELP.HTML PO VB 255 32760 C

-Library List: CBL.CBLI110.LSA
View Back Forward FDB Edit Help
Command>
Library> CBL.CBLI110.
-Member- Alias VV M
APEEINIT N 1
APEETERM N 1
CBLAVARL N
CBLAVCII N
CBLDLL00 N
CBLVIMSS N
CBLVIMST N
CBLVSQLO N
CBLVSVC N
Line 1 of 294 Col 1

-Allocated Datasets
View Back Forward FDB Edit Help
Command> where ddname>>D
DDName>
-DDName- -CSeq- -----DsN-----
DDBLSR 1 SYS06331.T095729.RA000.NBJ.R0102857
DDBLSRX 1 CBL.KSDS
DITPLIB 1 DIT130.SDITPLIB
DSQDEBUG 1 NBJ.NBJ.TSU07017.D00000102.?
DSQEDIT 1 SYS06331.T095729.RA000.NBJ.R0102856
DSQPRINT 1 NBJ.NBJ.TSU07017.D00000101.?
DSQSPILL 1 SYS06331.T095729.RA000.NBJ.SPILL.H01
Line 1 of 8 Col 1 of 108 Views 3 select * where
  
```

Figure 5. List Windows.

All list windows support the following:

- Point-and-shoot on column headers to sort column data.
- SQL style syntax for select, sort and filter.
- Prefix commands to copy, rename, delete, generate IDCAMS LISTCAT output, etc. as appropriate.

List windows provide a simple method of system navigation allowing the user to intuitively drill-down through lists of DASDs, VTOCs, Datasets, Libraries and finally, edit or browse a dataset or library member.

List windows are accessible via the **List** main menu item and directly via parameter driven command line interface commands. e.g.

<b>LD</b>	List Datasets.
<b>LL</b>	List Library Members.
<b>LA</b>	List Allocated Datasets.



## System Information List Windows (MVS Systems only)

The ability to display System Information is restricted by RACF profiles. Supported System List window types are:

- LPA Modules.
- Link Listed Libraries.
- APF Authorised Libraries.
- Task List.
- Private, CSA and SQA Storage Map.
- Loaded Programs.

**Operating System**

```

Operating system: z/OS
Operating system release: 1.6.0
VM guest machine id:
CPU serial number: 0A0000
  
```

**-LPA Modules**

Address	-Chain-	--RBP--	Dyn	--Name--	--EPA-
00C938E0	00000000	00000000	N		000000
00C93C30	00000000	00000000	N		000000
00C93C08	00000000	00000000	N		000000
00F4F428	00FA3910	00000000	N		000000
00C93C80	00000000	00000000	N		000000
00C93C58	00000000	00000000	N		000000
00C93BB8	00000000	0			000000
00FA0440	00F50080	0			000000
00F92050	00FA3940	0			000000

**-Link List**

```

View Back Forward FDB Edit Hel
Command>

-Seq- -----DsN-
1 SYS1.LINKLIB
2 SYS1.MIGLIB
3 SYS1.CSSLIB
4 USER.LINKLIB
5 SYS1.SIEALNKE
6 SYS1.SERBLINK
  
```

**-Task List**

Seq	Lvl	--TCB--	--Pgm--
1	0	008FDF30	IEAVAR00
2	1	008FD230	IEFSD060
3	2	008FD098	IKJEFT01
4	3	008DFD40	IKJEFT02
5	4	008DFBA8	IKJEFT09
6	5	008DFA10	ISPMMAIN
7	6	008DF878	ISPTASK
8	7	008DF548	DB

**-APF List**

DsNL	-----
12	SYS1.L
11	SYS1.S
11	SYS1.M
13	SYS1.S
11	SYS1.C
15	IGY330
15	EQA510
15	EQA510.SEQAAUTH
12	GIM.SGIMLMD0

**-Vol-- SMS**

Z6RES1	N
Z6RES1	N
Z6RES1	N
Z6RES1	N
Z6RES1	N
Z6RES2	N
Z6RES2	N
Z6RES2	N
Z6RES2	N

Figure 6. System List Windows.

## Dynamic SQL Interface to DB2 (MVS Systems only)

The DB2 Dynamic SQL window is used to submit SQL commands to a DB2 database and display the resultant messages and table views.

The screenshot shows a terminal window titled "Dynamic SQL: DB8G". The menu bar includes "View", "Back", "Forward", "FDB", "Edit", and "Help". The command prompt shows the following sequence of commands:

```

Command>
DB2 Subsystem> DB8G
Plan> CBLPLAN0
SQL Statement> SELECT * FROM SYSIBM.SYSCOLUMNS WHERE COLTYPE='CHAR'
                ORDER BY NAME
                >
                >
  
```

The results are displayed in a table with the following columns: NAME, TBNAME, TBCREATOR, COLNO, COLTYPE, LENGTH, and SC. The data is sorted by NAME.

NAME	TBNAME	TBCREATOR	COLNO	COLTYPE	LENGTH	SC
ACCESSPATH	SYSSTMT	SYSIBM	11	CHAR	1	1
ACCESSPATH	DGO_SYSPACKSTMT	IBMUSER	14	CHAR	1	1
ACCESSPATH	DGO_SYSSMT	IBMUSER	11	CHAR	1	1
ACCESSPATH	SYSPACKSTMT	SYSIBM	14	CHAR	1	1
ACCESSTYPE	PLAN_TABLE	IBMUSER	10	CHAR	2	2
ACCESSTYPE	PLAN_TABLE	DSN8810	10	CHAR	2	2
ACCESSTYPE	PLAN_TABLE	DPACK	10	CHAR	2	2
ACCTNO	SUPPLIER	Q	1	CHAR	5	5
ACQUIRE	DGO_SYSPPLAN	IBMUSER	12	CHAR	1	1
ACQUIRE	SYSPLAN	SYSIBM	12	CHAR	1	1
ACQUIRE	DGO_DGOPLAN	IBMUSER	13	CHAR	1	1
ACTION	TOPTVAL	DSN8810	2	CHAR	1	1
ACTION	VOPTVAL	DSN8810	2	CHAR	1	1
ACTION_IND	ADBCHK	IBMUSER	8	CHAR	1	1
ACTKWD	VACT	DSN8810	2	CHAR	6	6
ACTKWD	EACT	DSN8810	2	CHAR	6	6
ACTKWD	ACT	DSN8810	2	CHAR	6	6
ADMRDEPT	VDEPMG1	DSN8810	7	CHAR	3	3
ADMRDEPT	VDEPT	DSN8810	4	CHAR	3	3
ADMRDEPT	VHDEPT	DSN8810	4	CHAR	3	3
ADMRDEPT	NEWDEPT	DSN8810	4	CHAR	3	3
ADMRDEPT	EDEPT	DSN8810	4	CHAR	3	3
ADMRDEPT	DEPT	DSN8810	4	CHAR	3	3
ALTERAUTH	SYSTABAUTH	SYSIBM	14	CHAR	1	1
ALTERAUTH	SYSSEQUENCEAUTH	SYSIBM	7	CHAR	1	1

The status bar at the bottom indicates: "Line 1 of 1086", "Col 1 of 486", "Views 1", and "select \* sort NAME".

Figure 7. SQL DB2 Window.

## Dynamic AMS IDCAMS Execution

The AMS IDCAMS window is used to execute any IDCAMS syntax (DELETE, DEFINE, LISTCAT, REPRO, etc.) and then display the resultant SYSPRINT output.

```

-CBLi for TSO 1.4B - Build=200611231642 OpSys=z/OS 1.6.0 User=NBJ
File List Utilities System Window Swap Help
- IDCAMS Command: LISTCAT ENTRY(NBJ.CBLIDEMO.V000A.KSDS) ALL
View Back Forward FDB Edit Help
Command>
AMSCOMMAND> LISTCAT ENTRY(NBJ.CBLIDEMO.V000A.KSDS) ALL
>
Asa -----Line-----
1 IDCAMS SYSTEM SERVICES TIME: 1
0 MARGINS(1 32760)
0 IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
0
LISTCAT ENTRY(NBJ.CBLIDEMO.V000A.KSDS) ALL
0 CLUSTER ----- NBJ.CBLIDEMO.V000A.KSDS
IN-CAT --- USERCAT.CBLCAT
HISTORY
DATASET-OWNER----- (NULL) CREATION-----2006.331
RELEASE-----2 EXPIRATION-----0000.000
SMSDATA
STORAGECLASS ----CBLDFLT MANAGEMENTCLASS--CBLDFLT
DATACLASS ---- (NULL) LBACKUP ---0000.000.0000
BWO STATUS-----000000000 BWO TIMESTAMP---000000 00:00:00.0

- IDCAMS Command: REPRO INDATASET(NBJ.CBLIDEMO.V0000.KSDS)
View Back Forward FDB Edit Help
Command>
AMSCOMMAND> REPRO INDATASET(NBJ.CBLIDEMO.V0000.KSDS)
> OUTDATASET(NBJ.CBLIDEMO.V000A.KSDS)
Asa -----Line-----
1 IDCAMS SYSTEM SERVICES
0 MARGINS(1 32760)
0 IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
0
REPRO INDATASET(NBJ.CBLIDEMO.V0000.KSDS)
0 IDC0005I NUMBER OF RECORDS PROCESSED WAS 500
0 IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
0
0 IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0

Line 1 of 10 Col 1 of 126 Views 1 select *

```

Figure 8. AMS IDCAMS Window.

## General Utilities

CBLi contains some general tools that include:

- Multiple File search dialog window.
- Data set allocation and VSAM object definition dialog windows.
- Data set Copy, Rename and Delete dialog windows.
- VSE POWER command execution window.
- REXX calculator.
- Calendar with Julian day of year.

The screenshot displays three overlapping utility windows from the CBLi (Control Block List Interface) system. The top window is the 'File Search' dialog, showing a search for 'CBL.JCL(\*)' with a list of results including file names, record numbers, hit counts, and the commands to be executed. The middle window is the 'Define VSAM KSDS' dialog, showing the configuration for a new KSDS (Key-Seq-Data Set) named 'CBL.DEMO.KSDS'. The bottom window is the 'REXX Calculator', which provides instructions on how to use the calculator and shows a sample calculation:  $d2x(4 * 6683 \% 1024)$  resulting in the value  $1A$ .

```

-CBLi for TSO 1.4B - Build=200611231642 OpSys=z/OS 1.6.0 User=NBJ
File List Utilities System Window Swap Help
Command>

-File Search: CBL.JCL(*)
View Back Forward FDB Edit Help
Command>
  Dataset> CBL.JCL(*)
Search string> PGM=
-----Record-----
-Member- RecNo HitNo -----
CAESDREX 10 1 //RESTORE EXEC PGM=CAESDR,REGION=4M
CAESDRLK 8 1 //LINK EXEC PGM=IEWBLINK,PARM='REUS,AMODE=31'
CALIBADD 17 1 //INIT EXEC PGM=AFOLIBR,PARM='NRJS,NJTA'
CALIBADD 32 2 //ADD EXEC PGM=AFOLIBR,PARM='NRJS,NJTA'
CALIBADD 48 3 //LAMPNT EXEC PGM=IEBGENER
CALIBADD 57 4 //LAMPNT EXEC PGM=SELCOPY
CALSLC01 6 1 //SELCLAM EXEC PGM=SELCOPY
CAMEMADD 17 1 //ADD EXEC PGM=AFOLIBR,PARM='NRJS,NJTA'
CAMEMADD 49 2 //LAMPNT EXEC PGM=SELCOPY

-Define VSAM KSDS
Define Job AMS Help
>
Cluster Name> CBL.DEMO.KSDS Type> KSDS
Model> CBL.TEMP.KSDS Reuse> N
Catalog> Span> N
Speed> N
Avg rec len> 4089 Key length> 64 Writecheck> N
Max rec len> 4089 Key offset> 0 Erase> N
Buffer size>
Expiry>

Data name> CBL.D
Volumes> CBLM0
Space type> TRACK
CI Size> 4096

Index name> CBL.D
Volumes> CBLM0
Space type> TRACK
CI Size> 512

-REXX Calculator
Use this window to perform any calculation that can be
expressed in REXX syntax. Enter the expression below.

Note: All the normal arithmetic functions are available
together with conversion functions such as:
c2x Character to Hex c2d Character to Decimal
x2d Hex to Decimal d2x Decimal to Hex

Expression> d2x(4*6683 % 1024)
Value> 1A
  
```

Figure 9. Utility Windows.

## On-line Help System

Get on-line help for the current window or for a specific topic using the **HELP <topic>** command.



Figure 10. Help Windows.

## Section 02: New Facilities

### Interactive SELCOPY Debugger and Development Environment

SELCOPY Interactive debug is a major new feature for SELCOPY developers that executes within the CBL interactive environment.

SELCOPY Interactive is a Multiple Document Interface (MDI) application supporting CBL edit, List and storage type child windows. By default, the following child windows are displayed when SELCOPY is started:

1. SYSIN - SELCOPY Control Statements.
2. Work Area - (or Input Record if WORKLEN is not specified.)
3. SYSPRINT - SELCOPY Output Listing.
4. TRACE - Executed SELCOPY statements.

Other SELCOPY Interactive windows include:

1. WTO Log - SELCOPY Console output.
2. SQL Log - SELCOPY DB2 SQL Log.
3. @ Pointers - List of @ Pointer values.
4. EQUates - List of User assigned EQUated values.
5. POS Windows - Storage display of User fields and SELCOPY special positions.

The screenshot displays the SELCOPY Interactive Window with the following components:

- Top Menu Bar:** -CBLi for TSO 1.4B - Build=200701161527 OpSys=z/OS 1.6.0 User=NBJ. File List Utilities System Window Swap Help.
- Second Menu Bar:** SELCOPY. File View Go StepOver StepInto ReRun Window Help ? Sv ToF BoF Pfx < >
- Work Area (Main Editor):** Displays the SELCOPY control statements for NBJ.SELCOPY.SSDEMO01.SYSPRINT. The code includes comments like "AMEQU EXEC Q" and "Created by AMEQU CRE EXEC using the \*\*\* DO NOT CHANGE".
- SYSIN (Left Pane):** Shows the CBL code being executed, including conditional logic and data processing instructions.
- SYSPRINT (Right Pane):** Displays the output of the execution, showing a list of data records with fields like ABT01, ADA03, ADA08, etc.
- Status Bar:** Line=365 Col=1 Alt=0,0;100 Size=370 Recl=133 Fmt=V

Figure 11. SELCOPY Interactive Window.

On startup, SELCOPY is loaded and executed so that processing stops prior to execution of the first control statement. Users can then execute SELCOPY to completion, step through the SELCOPY statements one at a time or set break points to stop execution at a selected control statement.

SELCOPY storage fields, work area, @pointer and LRECL values can be easily monitored and even updated over the course of the job execution.

SELCOPY POSitional expressions (represented by an arithmetic combination of @ pointers, EQUate names and integer values) may also be TRACKed in storage windows. As the value of the expression changes, the target storage position is highlighted in a user assigned colour.

## SELCOPY SYSIN/SYSIPT Control Statements Window

The SYSIN/SYSIPT window is a CBL edit view that contains the control statement source file as required for execution of SELCOPY. This window highlights the current operation and allows the user to set and unset break points.

If possible, the control statements file is edited read/write. If the file contents are altered during the SELCOPY execution, then the alterations can be saved and the job rerun using the **RR** command or the **ReRun** menu item.

The contents of the window scroll automatically in order to display the current statement in the SELCOPY execution. As for any edit view, CBL commands and macros may be used to manipulate, highlight and locate data in the view (e.g. FIND, CHANGE, TAG, ALL, SET ZONE, etc.)

In addition to any highlighting of user TRACKed variables, during the course of execution, control statements are highlighted as follows:

1. Next executable SELCOPY statement. Default highlight - pink reverse video.
2. Break Point. Default highlight - red reverse video.

Closing the SYSIN window also exits SELCOPY Interactive.

```

-CBL1 for TSO 1.4B - Build=200611231642 OpSys=z/OS 1.6.0 User=NBJ
File List Utilities System Window Swap Help
SELCOPY - SYSIN: CBL.SSC.CTL(SSDEM001) 218 V PDS Size=97 Alt=0,0;0
File View Go StepOver StepInto ReRun Window Help ? Sv ToF BoF Pfx <>
Command> |...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
000036 @arr = marr
000037 ==memloop== * Loop to build array of member names.
000038 if @arr > marr+marrl-1 !t plog 'Array exceeded' !t eoi
000039
000040
000041
000042
000043 rd pds1 dsn=in1 dir , , into @arr * Directory records only.
000044 pos @arr+8, @arr+lrecl-1 = , , * Blank rest of record follow
000045
000046 if eof pds1 !then dummy
000047 else @arr = @arr+marre * Next input position.
000048 then goto memloop
000049 *memloope*
000050
000051
000052
000053
000054 print from marr,@arr+marre-1 * Print array for debug.
000055
000056 pos tot len=totl = x'00' fill x'00' * Initialise to hex zeroes.
000057 pos mat len=matl = x'00' fill x'00' * Initialise to hex zeroes.
000058 pos unml len=unml xor pos unml * Does the same thing.
000059
000060
000061
000062 ==pdsloop==
000063 rd pds2 dsn=in2 dirdata into pdsin * Dir + Data records only.
000064 if eof pds2 !then do log_rtn * Log totals.
000065 then eoi * End of job.
000066
000067 if data pds2
000068 then print from pdsin * PRINT data records.
000069 * then write outdd from pdsin * Write data records to DD o
000070 *<alloc f(outdd) reuse shr da('cbl.ssc.ctl(outdd)') !la outdd
F1=StepOver F2=StepInto F4=Point-and-Shoot Options F13=Go F14=BreakPoint
Line=36 Col=1 Alt=0,0;0 Size=97 Recl=218 Fmt=V

```

Figure 12. SYSIN Control Statements.

## Point-and-Shoot Options Popup Menu

All CBL type SELCOPY Interactive windows, including the SYSIN/SYSIPT and SYSPRINT/SYSLST windows, support the point-and-shoot options popup menu, invoked by hitting PF4. The menu enables the user to quickly and easily perform commonly used tasks which include opening a POS storage window and tracking pointer variables.

The cursor position within the edited data identifies the focus text to be referenced in items of the point-and-shoot menu.

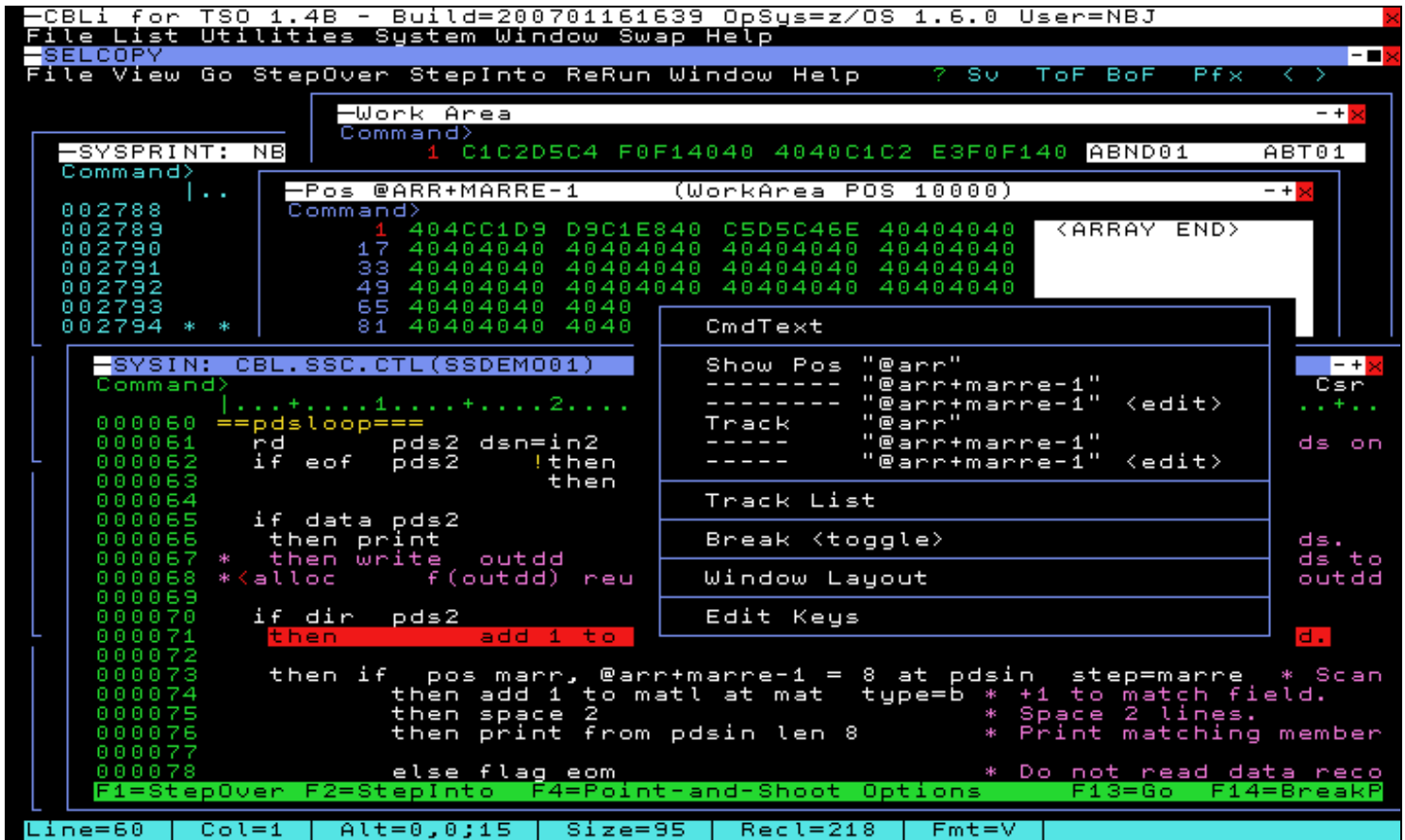


Figure 13. Point-and-Shoot Options Popup Menu.

## SELCOPY SYSPRINT Output Window

SELCOPY Interactive intercepts output to SYSPRINT/SYSLST and displays it in the SYSPRINT window instead. For this reason, SYSPRINT or SYSLST does not need to be allocated and no output is written to the system spool.

The SYSPRINT window is an edit view which supports execution of CBL commands and macros. The contents of the window scroll automatically to display any new output to SYSPRINT/SYSLST. Data written to the SYSPRINT window is maintained until the SELCOPY Interactive session is closed. Therefore, so long as the SELCOPY Interactive session is not closed, the job may be re-run any number of times without losing the SYSPRINT/SYSLST output from a previous run.



```

-CBLi for TSO 1.4B - Build=200611301630 OpSys=z/OS 1.6.0 User=NBJ
File List Utilities System Window Swap Help
SELCOPY - SYSPRINT: NBJ.SELCOPY.SSDEM001.SYSPRINT 133 V SEQ Size=224
File View Go StepOver StepInto ReRun Window Help ? SV ToF BoF Pfx < > Scroll> Csr
Command> |...+....1....+....2....+....3....+....4....+....5....+....6....+....7..

000098
000099      21.      else flag eom      * Do not read data
000100      22.      then log from pdsin len 8      * Log mismatching m
000101      23.      then add 1 to unml at unm type=b      * +1 to mismatch fi
000102
000103
000104      24.      goto pdsloop      * Get next record.
000105      *pdsloope*
000106
000107
000108
000109
000110      ==log_rtn==
000111      -----
000112      *      .....1.....2.....3.....4.....
000113      25.      pos lstr = 'Total Members: xxx, Matching members: xxx, Mis-m
000114      26.      cvbc totl at tot to lstr+15 fmt zz9
000115      27.      cvbc matl at mat to lstr+39 fmt zz9
000116      28.      cvbc unml at unm to lstr+66 fmt zz9
000117      29.      plog fr lstr len lstrl
000118      *log_rtn*
000119      30.      =ret=
000120
000121
000122      INPUT      SEL      SEL
000123      RECNO      TOT      ID.
000124      ---      ---      ---
000125      998      1      9      ABND01      ABT01      ADA01      ADA02      ADA03      AD
000126      ADA09      ADA10      ADA11      ADDLIT      AMEQU      AM
000127      ARIT04      ARIT05      ARIT06      ARIT07      AT01      AT
000128      AT07      AT08      AT09      AT10      AT11      AT
000129      AT17      AT18      AT19      BBABS01      BBACBIAS      BB
000130      BBCATJ01      BBCATJ02      BBCATJ03      BBCATJ04      BBCATJ05      BB
000131      BBCLONW      BBCONT01      BBDBRM23      BBDB201      BBDCB01      BB
000132      BBDYN01      BBEOFNW      BBEOFW      BBESDS      BBGDG      BB
000133      BBIMS04      BBIMS05      BBIMS06      BBINC01      BBMODIFY      BB
Line=98 Col=1 Alt=0,0;100 Size=224 Recl=133 Fmt=V

```

Figure 14. SYSPRINT Window.

## Work Area and POS Storage Windows

The current status of the user work area (or input record buffer if no work area is allocated) is displayed in the **Work Area** and **POS** storage display windows. Note, if WORKLEN is not supplied, the Work Area window has the title, **Current Input Record**.

The contents of the storage window may be scrolled to display different positions in the work area. Alternatively, storage starting at a specific position in the work area may be displayed simply by entering the position in the first work area position, in the first row of the storage window (highlighted in red by default.)

Data in the work area may be altered manually at any point during the run by overtyping text in either the character or hexadecimal display. A change to text in the one display will automatically be reflected in the other. This allows users to test conditions based on alternative input data without having to re-start the SELCOPY run.

If a TRACKed pointer position is within the storage window display, then that position is highlighted with a user nominated colour in reverse video.



Figure 15. Work Area/Current Input Record Window.

Any number of storage display windows may be opened pointing to the start of the work area (command: Window WORKLEN) or pointing at any position that may be resolved by a valid SELCOPY POS expression. POS expressions may contain any combination of SELCOPY reserved POS names (e.g. DATE), @ pointers, EQUate names and numeric integers. e.g.

```
WINDOW POS @ARR+MARRE-10
```

At each break in the SELCOPY execution, the POS expression is re-evaluated and the contents of the POS window redisplayed starting at the new position in storage.

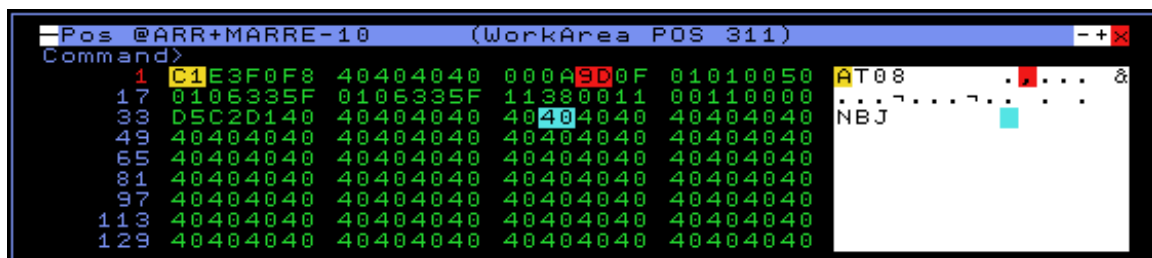


Figure 16. POS Expression Window.

Furthermore, the appearance of any individual storage window may be updated using the storage window display options popup menu defined on PF5 by default.

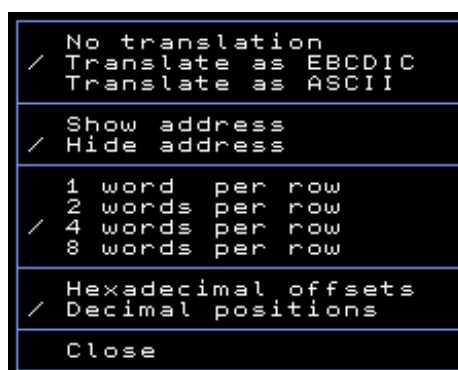


Figure 17. Storage Window Popup Menu.

## TRACE Window

The TRACE window is a CBL edit view that contains all the SELCOPY control statements at which processing has been stopped, i.e the point at which a break point was encountered. Each logged statement begins with the statement's selection id.

Note that the **STEPINTO** and **STEPOVER** commands dynamically set and unset break points to stop execution after the next control statements to be executed. Therefore, when stepping through the control statements, the TRACE window displays a log of the statements executed so far.

```

-CBLI for TSO 1.4B - Build=200611301630 OpSys=z/OS 1.6.0 User=NBJ
File List Utilities System Window Swap Help
-SELCOPY - TRACE: NBJ.SELCOPY.SSDEM001.TRACE 133 V SEQ Size=768 Alt=-
File View Go StepOver StepInto ReRun Window Help ? Sv ToF BoF Pfx < > Scroll> Csr
Command>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
000734 ** SELCOPY interactive execution started 2006/12/01 14:19:53 ***
000735 1. @arr = marr
000736 if @arr > marr+marrl-1
000737 4. rd pds1 dsn=in1 dir , into @arr * Directory record
000738 5. pos @arr+8, @arr+lrecl-1 = , * Blank rest of rec
000739 if eof pds1
000740 7. else @arr = @arr+marre * Next input positi
000741 8. then goto memloop
000742 if @arr > marr+marrl-1
000743 4. rd pds1 dsn=in1 dir , into @arr * Directory record
000744 5. pos @arr+8, @arr+lrecl-1 = , * Blank rest of rec
000745 if eof pds1
000746 7. else @arr = @arr+marre * Next input positi
000747 8. then goto memloop
000748 if @arr > marr+marrl-1
000749 4. rd pds1 dsn=in1 dir , into @arr * Directory record
000750 5. pos @arr+8, @arr+lrecl-1 = , * Blank rest of rec
000751 if eof pds1
000752 7. else @arr = @arr+marre * Next input positi
000753 8. then goto memloop
000754 if @arr > marr+marrl-1
000755 9. print from marr,@arr+marre-1 * Print array for de
000756 10. pos tot len=totl = x'00' fill x'00' * Initialise to hex
000757 11. pos mat len=matl = x'00' fill x'00' * Initialise to hex
000758 12. pos unml len=unml xor pos unml * Does the same thi
000759 13. rd pds2 dsn=in2 dirdata into pdsin * Dir + Data record
000760 if eof pds2
000761 if data pds2
000762 if dir pds2
000763 then add 1 to totl at tot type=b * +1 to total field
000764 then if pos marr, @arr+marre-1 = 8 at pdsin step=marre *
000765 18. then add 1 to matl at mat type=b * +1 to match field
000766 19. then space 2 * Space 2 lines.
000767 20. then print from pdsin len 8 * Print matching me
000768 24. goto pdsloop * Get next record.
000769 * * * End of File * * *
Line=734 Col=1 Alt=0,0;100 Size=768 Recl=133 Fmt=V

```

Figure 18. Trace Window Log.

## SQL Log Window

The SQL Log window is an edit view which supports execution of CBL commands and macros.

A SELCOPY job that submits SQL statements to a DB2 data base, also writes detailed information about the SELCOPY SQL processing to a data set allocated to ddname **CBLSQLLOG**.

SELCOPY Interactive intercepts output to CBLSQLLOG and displays it in the SQL Log window instead. Because of this, CBLSQLLOG does not need to be allocated to display this information.

```

-CBLi for TSO 1.4B - Build=200611301630 OpSys=z/OS 1.6.0 User=NBj
File List Utilities System Window Swap Help
SELCOPY - SQLLOG: NBj.SELCOPY.COURSE10.SQLLOG 133 V SEQ Size=33 Alt=
File View Go StepOver StepInto ReRun Window Help ? SV ToF BoF Pfx < >
Command> |...+...1...+...2...+...3...+...4...+...5...+...6...+...7..
000000 * * * Top of File * * *
000001 *** CBL Dynamic SQL Interface Version 2.01 AT ***
000002 -----
000003 CBLs010I 14:43:44 CBL Dynamic SQL Interface is started. Date: 2006-12-01
000004
000005 CBLs000I 14:43:45 (Sel 1) Connected to DB2 Version 8.1.0
000006 Subsystem:DB8G Plan:CBLPLAN0
000007 User:NBj Current SQLID:NBj
000008
000009 CBLs007I 14:43:47 (Sel 1) EXECUTE CREATE SQL Code=0
000010
000011 create table ssidemo ( char char(20) not null with default, int
000012 null with default, date date not null with default, time
000013 null with default )
000014 DB2 CPU= 000000.124703 sec
000015
000016 CBLs007I 14:44:28 (Sel 2) EXECUTE COMMIT SQL Code=0
000017 DB2 CPU= 000000.021067 sec
000018
000019 CBLs007I 14:44:29 (Sel 3) EXECUTE INSERT SQL Code=0
000020
000021 insert into ssidemo values ('First row ',1,current date,current
000022 Rows Inserted=1 DB2 CPU= 000000.047224 sec
000023
000024 CBLs007I 14:44:30 (Sel 4) EXECUTE INSERT SQL Code=0
000025
000026 insert into ssidemo values ('Second row ',2,current date,current
000027 Rows Inserted=1 DB2 CPU= 000000.034424 sec
000028
000029 CBLs007I 14:44:31 (Sel 5) EXECUTE INSERT SQL Code=0
000030
000031 insert into ssidemo values ('Third row ',3,current date,current
000032 Rows Inserted=1 DB2 CPU= 000000.032925 sec
000033
000034 * * * End of File * * *
Line=0 Col=1 Alt=0,0;34 Size=33 Recl=133 Fmt=V

```

Figure 19. SQL Window Log.

## WTO Log Window

The WTO Log window is an edit view which supports execution of CBL commands and macros.

SYSLOG output to the Operator's Console, TSO, CMS or ICCF user terminals is intercepted by SELCOPY Interactive and is displayed in the WTOLOG window instead.

The WTO Log window is opened automatically when SYSLOG output is received. This may be warning/error messages returned by SELCOPY, or output generated by a SELCOPY LOG operation.

```

WTOLOG: NBj.SELCOPY.SSDEMO01.WTOLOG 133 V SEQ Size=5 Alt=1,1;7 -+
Command> |...+...1...+...2...+...3...+...4...+...5...+...6...+...
000000 * * * Top of File * * *
000001 BBCARD01
000002 SS10264
000003 SS10264B
000004 WITHSEQN
000005 Total Members: 17, Matching members: 13, Mis-matching members: 4
000006 * * * End of File * * *

```

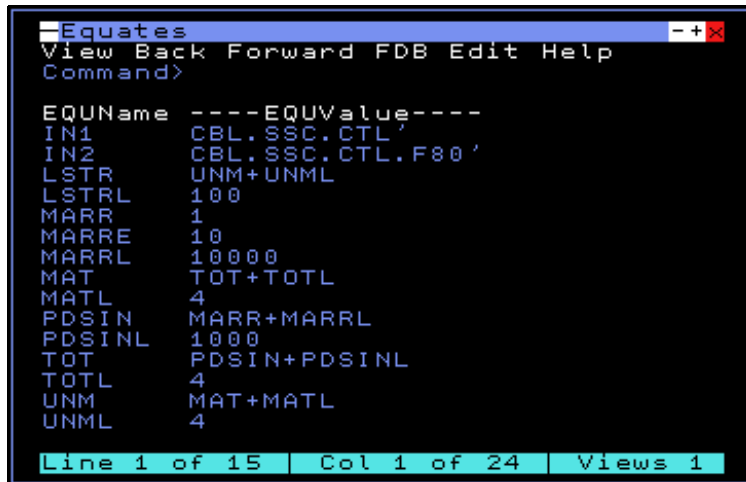
Figure 20. WTO Window Log.

## EQUates and @ Pointer Windows

The **EQUates** window and **@ Pointers** window are list type windows supporting select, sort and filter.

All equated names and their values, set by the user via an EQU statement and subsequently allocated by SELCOPY during control statement analysis, are displayed in the Equates window.

The current status of the @ pointer, LRECL and of all the user @ pointers used in the current execution of SELCOPY, is displayed in the @ Pointer window.



```

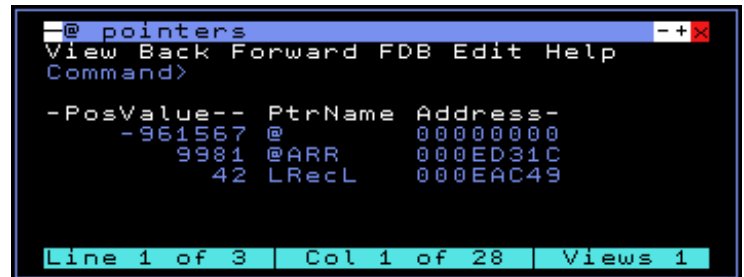
EQUates
View Back Forward FDB Edit Help
Command>

EQUName  ----EQUValue----
IN1      CBL.SSC.CTL'
IN2      CBL.SSC.CTL.F80'
LSTR     UNM+UNML
LSTR     100
MARR      1
MARRE     10
MARRL     10000
MAT       TOT+TOTL
MATL      4
PDSIN     MARR+MARRL
PDSINL    1000
TOT       PDSIN+PDSINL
TOTL      4
UNM       MAT+MATL
UNML      4

Line 1 of 15 | Col 1 of 24 | Views 1

```

Figure 21. Equates List Window.



```

@ pointers
View Back Forward FDB Edit Help
Command>

-PosValue-- PtrName Address-
-961567 @ 00000000
9981 @ARR 000ED31C
42 LReCL 000EAC49

Line 1 of 3 | Col 1 of 28 | Views 1

```

Figure 22. @ Pointers List Window.

## Non-standard 3270 Terminal Displays

The CBL Interactive Environment and SELCOPY Interactive take full advantage of large 3270 terminal sizes and colour display that may be achieved using IBM-Dynamic TN3270E VTAM logmodes.

CBL recommends that system administrators configure the TN3270 servers and 3270 emulation software to allow users to start emulated terminals with non-standard 3270 terminal sizes.

See the CBLi 3270 Emulators page of the CBL web site at:  
<http://www.cbl.com/cbli3270.html>

Also, IBM's Techdocs library entry, "Creating dynamic 3270 screen size definitions for increased productivity" at:  
<http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD102151>

```

--CBLI for ISO 1.48 - Build=200611301630 OpSys=z/OS 1.6.0 User=MBJ
File List Utilities System Window Snap Help
--SELCOPY
File View Go Stepover Stepinto Break Window Help ? Su ToF BoF Fz <>

--SYSPRM: CBL.DEMO.S.CTL(COURSE10) 80 F FDS Size=52 Alt=0.000 --
Command>
1...+...1...+...2...+...3...+...4...+...5...+...6...+...
000001 ** CBL.DEMO.S.CTL(COURSE10) ** L=002 --- 2004/10/08 09:56:51
000002
000003 * SELCOPY DB2 Example 2: Create, populate, print and drop a table
000004 *
000005 * This example shows how SELCOPY can be used to execute SQL to
000006 * create test data.
000007 *
000008 option WorkLen=256
000009
000010 equ selq 101
000011
000012 db2 sql="create table ssidemo
000013 (
000014         char      char(26)      not null with default.
000015         int        int           not null with default.
000016         date       date          not null with default.
000017         time       time          not null with default.
000018 )"
000019 db2 sql="commit"
000020
000021
000022 db2 sql="insert into ssidemo values
000023 ('First row','1,current date,current time)"
000024 db2 sql="insert into ssidemo values
000025 ('Second row','2,current date,current time)"
000026 db2 sql="insert into ssidemo values
000027 ('Third row','3,current date,current time)"
000028 db2 sql="commit"
000029
000030 pos selq = 'select * from ssidemo' stopat 1
000031 goto open stopat 1
000032 ==msgql==
000033 pos selq = 'select date from ssidemo where int = 2' stopat 1
000034 ==open==
000035 open ssidemo sql=40 at selq
000036 ==readloop==
000037 read ssidemo
000038 if eof ssidemo
000039 then goto close
000040 print type=mc
000041 goto readloop
000042
[=Stepover =2=Stepinto =4=Point-and-Shoot Options File=Go Fz=Break]

--IRACE: MBJ.SELCOPY.COURSE10.IRACE 133 V SEQ Size=9 Alt=0.0010 --
Command>
1...+...1...+...2...+...3...+...4...+...5...+...6...+...
000007 6. db2 sql="commit"
000008 7. pos selq = 'select * from ssidemo' stopat 1
000009 8. goto open stopat 1
000010 * * * End of File * * *

--Work Area --Pos SQLCA (Not in WorkArea) --
Command>
1 40404
17 404
33 404
49 404
65 404
81 404
97 404
113 344
129 404

--Pos SQLHA (Not in WorkArea) --
Command>
1 00000000 000130E1 00000000 00000000
17 00FD6888 00000000 7FFFFFF0 7FFFFFF0
33 7FFFFFF0 7FFFFFF0 7FFFFFF0 7FFFFFF0
49 00000000 00000000 7FFFFFF0 7FFFFFF0
65 00000000 00000000 00FD6888
81 00000000 00000000 00000000 000140E1

--SYSPRM: MBJ.SELCOPY.COURSE10.SYSPRM 133 V SEQ Size=11 Alt=0.000 --
Command>
1...+...1...+...2...+...3...+...4...+...5...+...6...+...
000034 7. pos selq = 'select * from ssidemo' stopat 1
000035 8. goto open stopat 1

--SQLLOG: MBJ.SELCOPY.COURSE10.SQLLOG 133 V SEQ Size=36 Alt=0.000 --
Command>
1...+...1...+...2...+...3...+...4...+...5...+...6...+...
000018
000019 CBL0007I 16:37:40 CSEL 3) EXECUTE INSERT SQL Code=0
000020
000021 insert into ssidemo values ('First row','1,current date,cu
000022 Rows Inserted=1 DB2 CPU= 000000.04774
000023
000024 CBL0007I 16:37:40 CSEL 4) EXECUTE INSERT SQL Code=0
000025
000026 insert into ssidemo values ('Second row','2,current date,cu
000027 Rows Inserted=1 DB2 CPU= 000000.03407
000028
000029 CBL0007I 16:37:41 CSEL 5) EXECUTE INSERT SQL Code=0
000030
000031 insert into ssidemo values ('Third row','3,current date,cu
000032 Rows Inserted=1 DB2 CPU= 000000.03356
000033
000034 CBL30
000035
000036
000037 * * *

000062
000063
000064 17.
000065 18.
000066
000067
000068
000069
000070
000071
000072 * * * End of File * * *

--Pos SELSQ (WorkArea POS 101) --
Command>
1 02853385 3703405C 40369996 34400202 select * from ss
17 33838534 36404040 40404040 40404040 idemo
33 40404040 40404040 40404040 40404040
49 40404040 40404040 40404040 40404040
65 40404040 40404040 40404040 40404040
81 40404

--Equates
View Back Forward F08 Edit Help
Command>
EQUName EQUValue
SELsq 101
Line 1 of 1 Col 1 of 16 View 1 select:

Line=1 Col=1 Alt=0.000 Size=52 RecL=80 Pri=F

```

Figure 23. SELCOPY Interactive 62x160.

## Section 03: Other Changes

---

### SELCOPY 2.01 Zaps applied

Zap ID	Op. Sys.	Query Ref.	Description
S201z01	All	(SQ11599 - 2006/11/15)	ERROR 102 was returned erroneously when RECFM=U is specified and multiple READ statements exist for the same frame.
S201z02	VSE	(SQ11605 - 2006/12/21)	<p>SELCOPY 2.00 zap s200z84 introduced support for the SELCOPY startup EXIT routine, CBLXS001, when running in an MVS system. (See <a href="#">SELCOPY 2.01 New Features</a>.)</p> <p>For VSE systems only, an 0C4 at SELCOPY+1F240 occurred when running SELCOPY 2.01 or SELCOPY 2.00 with s200z84 applied.</p>

### SELCOPY 2.00 Zaps applied

Please refer to the [SELCOPY 2.01 New Features](#) document for details on SELCOPY 2.00 zaps applied in SELCOPY 2.01.

**Note:**

Several fixes that were applied to SELCOPY release 2.01, closed loopholes in SELCOPY syntax. These fixes addressed potential I/O problems within SELCOPY and also inconsistencies that existed between the way in which file geometry information on I/O statements was interpreted and its documented effect as described by the SELCOPY User manual.

### SELCOPY Archive

Changes introduced and zaps applied in previous releases can be referenced in previous new feature documents:

- [SELCOPY 2.00 New Features](#).
- [SELCOPY 9.80 New Features](#).