UCRL-MA-108968 Rev. 1

PDBView User's Manual

for
PDBView Version 2.0

Stewart A. Brown
Dennis Braddy

Lawrence Livermore National Laboratory

Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Table of Contents

1.0 Introduction

The PACT system provides a number of tools for working with PDB files in a fairly generic fashion. In particular, PDBView lets you interactively browse a PDB file displaying the contents graphically or textually.

PDBView is an SX program. SX is the SCHEME dialect of the LISP programming language with PACT extensions. The extensions provide functionality for graphics, binary data handling, and other areas of functionality.

PDBView has a **help** command which provides information about available commands.

2.0 PDBView Syntax

PDBView uses a slightly different data description syntax than PDBLib. With PDBLib variable names and names of members of structures cannot contain the characters: “.”, “(“, “)”, “[“, and “]”. The characters “(“, “)”, “[“, and “]” are used in array reference and dimension definition expressions. In PDBView, “(“ and “)” CANNOT be used in variable reference expressions because they are special characters for PDBView. For example,

print a(2)

is illegal and results in an error. The legal expression is:

print a[2]

Other examples of legal expressions are:

print a.b[3,2]

print a.b[3][2]

print a[3].b[2][5].c

print a[3].b[2][3].c[3:4,5:50:10]

The first two forms are not identical. In the first form an element of a two dimensional array is being referenced. In the second form, the third element of the fourth array of arrays is being referenced (assuming zero based indexing in the definition of the b member of a).

For completeness and clarity in the following discussion, an index expression is defined as:

index expression := [*index list*]

index list := *index* | *index, index list*

index := *simple index* |

index-min : index-max | (*)

index-min : index-max : increment *(*)*

simple index := *integer*

index-min := *integer*

index-max := *integer*

increment := *integer*

If an index expression uses either of the two starred forms it is said to be a hyper-index expression. A hyper-index expression implies more than one data item. Only the terminating index expression in a data reference may be a hyper-index expression. An index expression is said to dereference an indirection (or pointer or array). For each level of indirection a suitable index expression dereferences the desired data.

For example, this means that a variable defined as:

```
char **s
```

is said to have two levels of indirection and can have parts accessed as follows:

print s	prints the entire item
print s[2]	prints the third character array of s
print s[3][10]	prints the eleventh character of the fourth character array of s

In the above example a zero based index is assumed.

When referring to part of a variable, especially a structured variable, the terminal node must be of primitive type or a structure containing no indirections and whose descendant members contain no indirections. Furthermore, the path to the desired part must contain one array reference for each level of indirection traversed.

3.0 Starting PDBView

On UNIX systems a shell script called **pdbview** is provided; it starts up SX which then loads the PDBView SCHEME forms. You must add a line to your **.cshrc** or **.profile** file that defines the environment variable SCHEME. This variable tells SX where to find the PDBView SCHEME files. Consult your system administrator or some other knowledgeable source to find out where these files are located on your system. If, for example, the directory **/usr/local/scheme** contains the PDBView SCHEME files, add the line:

```
setenv SCHEME /usr/local/scheme
```

to your **.cshrc** or **.profile** file.

Usage: pdbview [-d] [-h] [-l *command-file*] [-p *n*] [*data-file*]

Options:

d - dump the formatted contents of the data file in ASCII to stdout

h - print execute line information to stdout

l - load SCHEME forms (e.g. PDBView commands) from a file
(not available with -d option)

p - number of digits to use in displaying floating point numbers
(only available with -d option)

4.0 PDBView Commands

In this section the PDBView commands are listed alphabetically. The command name is followed by synonyms, if any. Each command is given with a brief description of its function and its usage. The examples are intended to illustrate the various ways of invoking each command. Optional arguments are enclosed in brackets. Alternate argument forms are separated by '|'. Formal arguments in italic font are replaced by actual arguments. Command and argument keywords are in regular non-italic font.

The file most recently referred to in a **cf** (**change-file**) command is the current file. Most commands implicitly refer to the current file or its contents. Some commands permit another file to be explicitly specified. A file may be referred to by name or by the alias assigned when the file was opened. Commands that operate on the contents of more than one file (e.g. **copy**, **copy-mapping**), require that the target file be specified, while the current file is assumed to be the source.

For the sake of brevity all menu items including mappings, images, and ULTRA curves are collectively referred to as mappings. All commands that accept mappings as arguments, accept images and curves as well.

CD

Change the current file directory. The new directory may be specified by either a relative path or a full path.

Usage: cd [*directory*]

Examples:

cd /zoo/mammals

cd ../reptiles

cd

CHANGE

Reset values in a variable or structure member. Note that the command keyword may be omitted. To change array elements, qualify the name with index expressions. If fewer values are supplied than the number of elements referenced, the last value is repeated. (See the PDBView Syntax section for a discussion of index expressions.)

Usage: [change] *variable* | *structure-member value*

Examples:

change a[10,15] 3.5

change time 0.0

x[1:2] 1 2

dir1/jkl.k 2

CHANGE-FILE / CF / OPEN

Change the current file. If it is not already open, open the file. Many commands refer to the current file or its contents by default. Any number of files may be open - only one is the current file. The mode may be: **r**, open file read only; **a**, open file read/write (default); or **w**, overwrite any existing file. If no alias is specified, **f*d*** is assigned, where *d* is an increasing decimal integer. The options available for type are determined by the output spokes you have installed. The default type is **pdb**.

Usage: change-file *filename* [*mode* [*alias* [*type*]]]

Examples:

    change-file foo
    cf foo a
    open foo w bar pdb

CLOSE

Close a file. The default is to close the current file.

Usage: close [*file*]

Examples:

    close
    close foo
    close f1

CLOSE-WINDOW / CLW

Close a graphics window. If no window is specified, close the current window.

Usage: close-window [*window*]

Examples:

    close-window "ABC"
    clw

COMMAND-LOG

Turn logging of typed commands on or off. A log file name may be specified in place of the keyword **on**. The default log file name is **pdbview.log**. If the log file already exists, it is appended to. Logging is off by default.

Usage: command-log [on | off | *filename*]

Examples:

    command-log on
    command-log mylog2

COPY

Copy variables from the current file directory to another file. If the variable list is *****, copy all variables in the current file directory. The target file must be open and may be specified by name or alias.

Usage: copy *target-file variable-list*

Examples:

copy foo *

copy f3 bird cat dog

COPY-MAPPING / CM

Copy mappings from the current file directory to another file. Mappings are referenced by the numbers displayed by the **menu** command. If the mapping list is *****, copy all mappings in the current file directory. The target file must be open and may be specified by name or alias.

Usage: copy-mapping *target-file mapping-list*

Examples:

copy-mapping foo 1 5

cm f2 *

DESC

Describe variables or structure members in the current file directory. To get a description of part of a variable or structure member, qualify the name with index expressions. (See the PDBView Syntax section for a discussion of index expressions.) If a selection pattern is specified in place of a variable name, all variables matching the pattern will be described. Each ‘?’ in the pattern matches any single character. Each ‘*’ in the pattern matches any string of zero or more characters. An optional type qualifier may also be specified in order restrict the list to a given type.

Usage: desc *variable* | *structure-member* | *pattern* [*type*]

Examples:

desc Mapping1

desc dir1/a.b.c[12:14]

desc * double

desc var? integer

DISPLAY-MAPPING / DM / PL

Plot the specified mappings. Mappings are referenced by the numbers displayed by the **menu** command or by the names of variables containing mapping information.

Usage: display-mapping *mapping-list*

Examples:

display-mapping 5

dm dir1/Mapping8

pl 5 8

END / QUIT

End the session of PDBView.

Usage: end

FILE

Describe the current file. Optionally, the file type alone may be requested.

Usage: file [t | type]

Example:

file type

FORMAT

Set the printing format for a specified data type. If the argument specifying the type has ‘1’ or ‘2’ appended, then the new format is applied only to arrays shorter than **array-length** or to arrays longer than or equal to **array-length**, respectively. Otherwise, the format applies to both. Invoking the format command with the single argument, **default**, causes the formats for all types to be reset to their default values. The *format* argument must be a Standard C I/O Library format string. Double quotes are only necessary if the format string contains embedded blanks. See the **set** command for more information about the **array-length** variable. This command overrides the settings of the **decimal-precision** and **bits-precision** display control parameters.

Usage: format integer[1 | 2] | long[1 | 2] | float[1 | 2] |
double[1 | 2] | short[1 | 2] | char[1 | 2] *format*

Usage: format default

Examples:

format double %12.5e

format double2 %10.2e

format char “%s “

HARDCOPY-WINDOW / HC / HCW

Draw the contents of the current window to all open hardcopy devices. Color output is indicated by the optional keyword, **color**, on the first call for a given device. The resolution is decreased below full device resolution by an integer resolution scale factor, if present. Hardcopy files remain open until explicitly closed or until PDBView is terminated.

Usage: hardcopy-window [color] [*resolution-scale-factor*]

Examples:

hardcopy-window

hc color 8

HELP

Print a list of commands or documentation for an optionally specified command.

Usage: help [*command*]

Examples:

help

help menu

LD

Read SCHEME forms (e.g. PDBView commands) from the specified ASCII disk file. The -l execute line option can be used to cause PDBView to read a file of SCHEME forms at start-up.

Usage**:** ld *filename*

Example:

ld script.scm

LIST-FILES / LF

List the open files. The current file is indicated by an asterisk.

Usage: list-files

Examples:

list-files

lf

LS

List the names of variables, links, and directories in the current file directory. Directories have a terminal slash. An optional selection pattern may be specified. Each '?' in the pattern matches any single character. Each '*' in the pattern matches any string of zero or more characters. An optional type qualifier may also be specified in order restrict the list to a given type.

Usage**:** ls [*pattern* [*type*]]

Examples:

ls dir1/curve*

ls var? integer

ls * Directory

LS-ATTR

List the attributes in the current file.

Usage: ls-attr

MENU

List the labels of mappings in the current file directory. The ordinals preceding the labels are used to reference the mappings in other commands. An optional selection pattern may be specified. Each '?' in the pattern matches any single character in a label. Each '*' in the pattern matches any string of zero or more characters. Note that the pattern, if supplied, must match the entire label. Liberal use of "*"

Usage: menu [*pattern*]

Examples:

menu

menu ?*d

menu *foo*

MODE

Set the print mode for structures.

Display modes are:

- **full-path** - the full path name is printed at each branch, e.g. foo.bar[3].baz
- **indent** - indent 4 spaces at each branch (default)
- **tree** - display as a tree (lines connecting branches)

Type display is controlled by:

- **no-type** - turns off the display of types (default)
- **type** - displays the type of each item and branch

Display of recursive structures is controlled by:

- **recursive** - indent each level of recursive structures
- **iterative** - number each level of recursive structures (default)

Usage: mode full-path | indent | tree | no-type | type | recursive | iterative

Example:

    mode full-path

N-ENTRIES

Print the total number of variables, links, and directories in the current file.

Usage: n-entries

PALETTE

Set the current color palette.

Usage: palette standard | spectrum | rainbow | bw

Example:

    palette rainbow

PLOT

Plot one variable or part of a variable (range) against another (domain). Plot versus index (logical) if no domain given.

Usage: plot *range* [*domain*]

Examples:

    plot y x
    plot z
    plot dir1/ddd[0:60, 0:15, 20]

PRINT-MAPPING / PM

Print out the specified mappings. Mappings are referenced by the numbers displayed by the **menu** command.

Usage: print-mapping *mapping-list*

Example:

print-mapping 3 4

PRINT

Print out all or part of the specified variable or structure member. Note that the command keyword may be omitted. To print part of a variable or structure member, qualify the name with index expressions. (See the PDBView Syntax section for a discussion of index expressions.)

Usage: [print] *variable* | *structure-member*

Examples:

Mapping2

print Mapping4.domain.elements

print dir1/Mapping2.range.elements[1]

a[5,10:20,1:8:3]

print a.b[3].c[5,10:20,1:8:3]

PWD

Print the current file directory.

Usage: pwd

SET

Set the value of some array display parameters. Parameters:

line-length - the number of array elements per line

array-length - for arrays shorter than value, label each element individually

bits-precision - number of mantissa bits of precision

decimal-precision - number of digits of precision

Usage: set line-length | array-length | bits-precision | decimal-precision *value*

Examples:

set line-length 3

set decimal-precision 6

STRUCT

Describe the specified data type in the current file.

Usage: struct *data-type*

Examples:

struct double

struct PM_mapping

TYPES

List the data types in the current file.

Usage: types

5.0 Default State

The state in which PDBView starts out is:

- mode iterative
- mode indent
- mode no-type
- set array-length 6
- set line-length 3
- set decimal-precision 4
- palette spectrum
- command-log off

6.0 Other PACT Documents

Interested readers may wish to refer to the other PACT documents which describe the data structures and functionality underlying the more common PDB files upon which PDB-View operates. The PANACEA, PGS, and PDBLib manuals are of special interest to people who wish to generate and view data files with PACT.

The list of PACT Documents is:

PACT User's Guide, UCRL-MA-112087

SCORE User's Manual, UCRL-MA-108976 Rev.1

PPC User's Manual UCRL-MA-108964 Rev.1

PML User's Manual, UCRL-MA-108965 Rev.1

PDBLib User's Manual, M-270 Rev.2

PGS User's Manual, UCRL-MA-108966 Rev.1

PANACEA User's Manual, M-276 Rev.2

ULTRA II User's Manual, UCRL-MA-108967 Rev.1

PDBDiff User's Manual, UCRL-MA-108975 Rev.1

PDBView User's Manual, UCRL-MA-108968 Rev.1 (this document)

SX User's Manual, UCRL-MA-112315

Symbols

Numerics

A

B

C

D

E

F

G

H

I

T

V

W