CDEV Utilities

Utility Applications of the Control Device Interface

Chip Watson, Jie Chen, Danjin Wu, Walt Akers

Version 1.5 - December 9, 1996

TJNAF - Thomas Jefferson National Accelerator Facility

DOCUMENT DATE: Table of contents generated: December 9, 1996 11:47 am

TRADEMARKS: UNIX is a registered trademark of AT&T in the USA and other countries.

VxWorks is a register trademark of Wind River Systems.

The X Window System is a trademark of Massachusetts Institute of Technology.

OSF/Motif and Motif are trademarks of Open Software Foundation, Inc.

Ultrix and DEC are registered trademarks of Digital Equipment Corporation.

HPUX is a registered trademark of Hewlett Packard.

SURA/CEBAF: The Southeastern Universities Research Association (SURA) operates the Continuous Electron Beam Accelerator Facility (CEBAF) for the United States Department of Energy under contract DE-AC05-84ER40150.

Table of Contents

List of Figures

1. The *cdevUtil* Command Line Utility

Overview

The cdevUtil application provides cdev user's with a simple application that can be used to communicate with cdev services. This program provides a direct interface to most of the capabilities of cdev and is commonly used to test a newly created service or to ensure that changes made to a DDL file are behaving as expected.

Location of Files

The source code for cdevUtil is located in the ***$(CDEV)/src/utility*** directory. This application is automatically compiled whenever the cdev library is built. The binary executable is placed in the directory ***$(CDEV)/bin/$(TARGET)*** when the application is compiled. Once the cdev libraries and applications are compiled, this directory should be placed in your search path.

Starting cdevUtil

The cdevUtil utility is executed from the UNIX command line by typing *cdevUtil <Return>*. If the cdev binary directory is not in your path, then you will have to provide the complete path on the command line. The CDEVDDL environment variable should be specified prior to starting the utility because cdevUtil will use this environment variable to locate and load the device definition file.

Arguments to cdevUtil

When starting the cdevUtil utility, the user may provide a list of strings as arguments. Each of these strings will be treated as a single command and will be processed before the command line prompt is displayed. Any command that is supported by cdevUtil may be submitted as an argument, including, "*debug on*", "*debug off*", and *"quit*". The syntax of individual commands is described in the section titled ***Syntax of cdevUtil Commands.***

The following figure shows how three command line arguments can be submitted to cdevUtil to be processed in batch mode.

Figure 1: Submitting batch commands to cdevUtil

```
host1:> cdevUtil "debug on" "IPM1S01 set VAL value=25.0" "quit"

Submitting batch cdev commands...
-------------------- DEBUG OUTPUT -------------------
Attempting to transmit using the following parameters
Device: IPM1S01
Verb:   set
Attrib: VAL
Tag: value = 25

SUCCESS: Device: IPM1S01 - Message: set VAL - Value: None
------------------ END DEBUG OUTPUT -----------------

host1:>
```

Syntax of cdevUtil Commands

The cdevUtil utility uses a simple syntax with the following format:

deviceName verb attributeName tag1=xxx tag2=xxx ...

deviceName The name of the cdev device that the message should be sent to.

verb The verb component of the message string. For a channel access device this will be get, set, monitorOn or monitorOff.

attributeName The name of the cdev device attribute that the verb will be executed on.

tag=xxxx These entries are used to place information in the outbound cdevData object. tag is the text string name of the cdevData tag. xxxx is the value that should be placed in the tag. At this time cdevUtil only supports the insertion of scalar values into the cdevData object.

The following example shows the commands necessary to submit a request to get the value of a device, and then set it to a new value.

Figure 2: Entering commands using cdevUtil

```
host1:> cdevUtil
> IPM1S01 get VAL
> SUCCESS: Device: IPM1S01 - Message: get VAL - Value: 0
> IPM1S01 set VAL value=1.0
> SUCCESS: Device: IPM1S01 - Message: set VAL - Value: None
> quit
host1:>
```

Exiting from cdevUtil

In order to exit from cdevUtil, the user must type ***quit <Return>***. This will remove any monitors that might have been created and will terminate the session.

cdevUtil Internal Commands

The following commands are processed by the cdevUtil application and are not passed through to cdev.

debug on This message tells cdevutil that the user wants extremely verbose output. When this is flag is set, the complete contents of the incoming and outgoing cdevData objects will be displayed. Typically only the value field of the incoming cdevData object is displayed.

debug off This message tells cdevUtil to stop displaying verbose output.

help This command causes cdevUtil to display an simple help screen that describes its usage and internal commands.

quit This command terminates the cdevUtil utility.

2. The *cddl* Device Definition File Compiler

Overview

The cddl compiler is used to convert the ascii text device definition files into a binary, pre-processed device definition file. The binary DDL files load as much as twenty times faster than their ascii text based counterparts. Additionally, the cddl compiler confirms that no syntax errors exist in the ascii data files when they are compiled. Therefore, this tool is also effect in validating the syntax and structure of the DDL files.

Location of Files

The source code for cddl is located in the ***$(CDEV)/src/utility*** directory. This application is automatically compiled whenever the cdev library is built. The binary executable is placed in the directory ***$(CDEV)/bin/$(TARGET)*** when the application is compiled***.*** Once the cdev libraries and applications are compiled, this directory should be placed in your search path.

Using the cddl Compiler

The syntax of cddl is very similar to that of the C preprocessor. The following is a sample command line for compiling the device definition file.

cddl inputfile -o outputfile

It is important to note that the ***inputfile*** parameter must contain a complete path specification to the device definition file. Any errors will be reported to the screen. The following is the typical output from executing the cddl compiler.

Figure 3: Typical output from the cddl compiler

```
host1:> cddl ~mccops/ddl/master.ddl –o master.bdd
converted from ascii to binary ddl file!
host1:>
```