CDEV Administrator's Guide

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 11:44 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

1. Directory Structure of the CDEV Distribution

Directory Structure

./src	*The CDEV source parent directory, it contains makefiles and subdirectories with the source code necessary to create the CDEV libraries.*
. . . . ./src/ca	*Souce code providing CDEV service support for channel access and EPICS.*
. . . . ./src/cdevIO	*Souce code for the cdevData object and its components.*
. . . . ./src/common	*Source code for CDEV components that are used by class internals.*
. . . . ./src/callback	*Source code for the cdevCallback object used with callback operations and monitors.*
. . . . ./src/device	*Source code for the cdevDevice and cdevRequestObject objects and their associated component classes.*
. . . . ./src/directory	*Source code for the cdevDirectory anc cdevSvcFinder classes.*
. . . . ./src/group	*Source code for the cdevGroup and cdevTranObj classes.*
. . . . ./src/lib	*Directory for library Makefile and temporary object files.*
. . . . ./src/namesvc	*Source code for tools used to parse the CDEV DDL file.*
. . . . ./src/system	*Source code for the cdevSystem and cdevService classes and their associated components.*
. . . . ./src/utility	*Source code for a CDEV utility application.*
./include	*Header files for CDEV applications, this directory will be populated during the first execution of make*
./lib	*The CDEV library parent directory, will be populated wth subdirectories for each target architecture.*
./bin	*The CDEV binary parent directory, will be populated wth subdirectories for each target architecture.*
./doc	*Contains most up to date versions of the CDEV documentation.*
./example	*Sample applications and services that use CDEV.*
./extensions	*Software extensions and services for CDEV that are being developed by individual developers. These services add value to CDEV, however, they are not required by the system.*

2. Installing and Building the CDEV Libraries

How to Start

The following steps should be taken in order to obtain and install CDEV at your site.

1. **Obtain a copy of the CDEV distribution.** A complete copy of the CDEV distribution may be obtained by anonymous ftp at **ftp.cebaf.gov** in the directory **pub/cdev**.

2. **Unarchive and install the distribution.** The CDEV distribution is provided in a compressed and archived format. The archive may be decompressed using the GNU *gzip* program. The archive may then be extracted using the UNIX *tar* command.

Configuring Static Services

3. **Configure static services (archive libraries).** If you will be using the archive library version of CDEV, you must tell CDEV the names of the services it will be expected to use. This must be done before any other installation procedures.

   The CDEV system provides a way to configure services so that applications can access them by name. For example, if you have the channel access service (ca) and a locally developed service (xyz), you can configure CDEV to use these services with the following command.

   ```
   configSvc services=ca xyz headers=/usr/local/epics/include \
      /usr/local/cdev/src/ca /usr/local/cdev/src/xyz
   ```

   In the above examples, *services* are the names of the CDEV services that should be supported, and *headers* are the directories that should be searched for the associated header files for these services. The *configSvc* script will generate a Makefile and cdevSvcConfig.cc in the *./src/directory* directory.

Setting the Local Environment

4. **Set CDEV required make variables.** The following environmental variables must be defined prior to making the CDEV libraries.

CDEV:	This is the root directory of the CDEV distribution.
CDEVDDL:	This is the location of the CDEV Device Definition Language file.
CDEVSHOBJ:	This is the directory where the CDEV shared libraries and shared objects will be stored.
CDEVINCLUDE:	This is the directory whewre the CDEV header files are located.
SHOBJ:	This variable indicates that a sharable object library should be compiled. In order to generate a sharable library, *SHOBJ="YES"* should be specified. This variable is undefined by default.

Compiling the Distribution

5. Change directories to the CDEV distribution ./src directory. This is where the Makefile and Makefile.common will be located.

6. If you have compiled the CDEV distribution for another TARGET in the past, you should execute a ‘make clean’ to remove any outdated object files.

7. If you want to change default time-out value and connection retry frequency in channel access or EPICS, you should change DEFAULT_TIMEOUT and DEFAULT_CONN_FREQUENCY values in the caRequestObject.h.

8. To begin compilation type the command ‘make all’ or ‘make’. This will compile all libraries and utility applications and place them in the appropriate TARGET directories.

9. To remove objects after making archive libraries, type ‘make cleanObjs’. This is useful to make the archive libraries and then make the shared libraries.