JLAB-ACE-99-08

TOOLS FOR APPLICATION MANAGEMENT AT JEFFERSON LAB*

S. Schaffner, M. Bickley, A. Hofler, M. Keesee, D. Wetherholt, K. White, Thomas Jefferson National Accelerator Facility, Newport News, Virginia, USA

Abstract

The Software Controls Group at Thomas Jefferson National Accelerator Facility (Jefferson Lab) is responsible for slow controls for many Jefferson Lab facilities. The Experimental Physics and Industrial Control System (EPICS) is used as the basis of these control systems. The Controls Group developed and maintains over 150 control applications running on over 100 I/O controllers (IOCs). With so many applications, it becomes increasingly difficult to maintain and upgrade older applications and still produce new applications. The difficulties became especially apparent this year as a major effort was undertaken to upgrade all control system applications to the newest versions of EPICS and VxWorks. Over the past few years, the Controls Group has worked on constructing a framework within which to develop and maintain applications more efficiently. As the framework has matured and applications have been structured to fit the framework, a number of tools have been developed to help with software maintenance and upgrades. This paper will describe some of these tools and how they are used to enhance the maintainability and reliability of the control system.

1 APPLICATION CREATION

1.1 File System Organization

At Jefferson Lab, the file system for IOC applications is organized into two source areas, one for test code and another for production code, a single Concurrent Versions System (CVS) repository for version control, and an area for EPICS IOC core. There are multiple production areas for object code and IOC boot directories, including a development test area. The source areas, CVS repository, EPICS source, and the development test area are organized into a "fiefdom" which means that they share a Unix file server and are on their own subnet. The other production object areas are in separate fiefdoms. The system is organized in this manner to minimize disruptions in one fiefdom if parts of a separate fiefdom fail or are taken down for maintenance. The IOC application tools have been designed to help manage the development and installation of software across fiefdoms.

1.2 Application Versioning

All IOC applications are "versioned." For the source areas this means that all source code and the IOC startup scripts associated with a running version of an application are stored in the CVS repository and tagged with the same version number. Object code and operational IOC startup scripts associated with a tagged source version are stored in a subdirectory named by version in the production object areas for the various fiefdoms.

In practice, at least two versions of object code are kept in the production area: the current operational version and the rollback version (the version immediately preceding the current operational version). The rollback version has typically been in operations for weeks or months and has been "proven" to run correctly. The current operational version probably includes new or upgraded features and may cause problems in operations since it is sometimes difficult to adequately test all new features in the test area.

Some of the tools described below are used to help the software group quickly and reliably "roll out" a version of an IOC application which is causing trouble and restore the most recent operationally proven version until the developer has time to fix the problem. This has proven to minimize the impact on operations and improve the "up time" of the accelerator (a very important metric for our lab).

1.3 Application Types

Application creation begins with a request from a "customer." The request is analyzed and the level of effort assessed. A software engineer (developer) or team of developers is assigned to work on the project. The developer analyzes the requirements and specifications, determines what types of code need to be developed (driver code, device support, EPICS databases, etc.) and breaks the system down into one or more IOC applications.

IOC applications are classified into three types. The first type contains EPICS databases. These applications may or may not contain custom device/driver support. They are typically quite specialized to a particular control function or control device. The second type contains only device/driver support and is typically developed for commercial off-the-shelf control modules. These applications may be used by one or more EPICS database

* Work supported by the U.S. Department of Energy, contract DE-AC05-84ER40150

applications. The third type contains only EPICS record support and is used for custom records that are shared by multiple applications.

The directory structures, makefiles, IOC startup scripts, and installation procedures are all slightly different for each application type. A set of application creation scripts (mkNewapp, mkNewdev, and mkNewrec) has been developed for each application type.

Each script creates a source directory structure in the development source area. The directory structure is named by application and by default, the working subdirectories are placed in version 1-0 underneath the application name. A CVS module for the application is created. Template makefiles, EPICS .dbd files, and application IOC startup scripts are copied into the appropriate subdirectories. Soft links are made in the source directory to the production test area for later use by the installation procedure.

The template makefiles contain targets to create and install object code and IOC startup scripts to the various production areas. The developer adds source module names and install locations. The IOC startup scripts are kept in the application source area so they can be CVS'd and tagged along with the application source code.

At this point, the developer is ready to write the application source code, customize the template makefiles and IOC startup scripts appropriately, and test the application in the development test area. At this time the developer also sets up the installation procedure to install object code to the appropriate production area(s).

2 INITIAL INSTALLATION AND CHECKOUT

2.1 Move to Source Production Area

Before installing the application the developer uses the "CVS and Release Notes " tool, to store the application source code in the CVS repository, tag it with the version number, and write the application description. The application description is automatically converted to html format and stored in a location accessible by another tool, the "LLAPP Version Info" tool, which is a custom Web browser described later.

Another script, applNewver, is used when the source code for the application is ready to be installed in the production source area. This script takes the current version of the application out of the CVS repository, stores it in the production source area, and creates soft links to all the production object areas for later use by the installation procedure. At this point, the knowledge of the directory structure has been stored in the CVS repository, so a single script can be used to retrieve all application types.

Once the developer has compiled the source into object code and converted Capfast schematics to EPICS databases, the install procedures encoded in the makefiles are used to install the objects to the correct production area.

2.2 Scheduling Installation and Writing a Test Plan

Before an IOC application can be installed, a test plan must be written and approved and the installation must be scheduled during one of the approximately bimonthly accelerator maintenance periods. A Web-based tool has been written for the developer to use for all these functions.

Using a template form, the developer provides basic information needed by maintenance schedulers such as which IOCs will be affected, which applications will be affected, what types of changes will be made and how long the installation will take. Maintenance schedulers have this information available well before the actual maintenance period allowing hardware and software tasks to be coordinated.

A full test plan detailing the new and rollback versions for the application, the IOC installation and setup procedure, installation and rollbacks for affected Unix files (medm screens, burt files, etc.), test procedures, and rollback procedures must be written and approved before the maintenance period. The developer uses the same template form started for the scheduler to complete the test plan. When completed, the developer electronically submits the test plan. Submission means that test plan approvers are e-mailed the URL of the test plan. Approvers read the test plan and either approve it or request more information from the developer. Maintenance day schedulers are e-mailed the URL of the test plan once it has been approved.

2.3 Managing Soft Links in the IOC Boot Directory

When it is time to install the new or upgraded application another tool is used to configure and change the soft links in the IOC boot directories. All applications and IOC startup scripts are accessed in the IOC boot directories through soft links. This provides the means to quickly access new or rollback versions of applications. In addition, the ability to change the soft link for a single application minimizes the chance that other applications running on the same IOC which have not been changed will be affected.

The IOC soft link configuration is stored in a configuration file in the IOC boot directory. A "linkmaker" tool makes the actual soft links. This tool reads the configuration file, makes the appropriate soft links, and verifies the existence of the files pointed to by the links. Part of the linkmaker tool includes a link browser. The browser reads a configuration file, displays the links that will be made for an IOC (without actually making the links) and verifies the existence of files pointed to by the links. A "future" configuration file can be set up and developers can check to make sure all needed files have been correctly installed in the production object area prior to the maintenance period, which saves time. The linkmaker tool has proven very useful, particularly when upgrading all applications on an IOC, as when the control system changed to a new version of EPICS and VxWorks for Y2K compliance this year. An added advantage of using the link configuration files along with a well-defined procedure for naming and changing them is that an unambiguous record exists of the rollback path for an IOC.

2.4 Completing the Installation

Once the maintenance period begins, most of the planning and code installation has already been completed. The only preparation that needs to be done is to modify the IOC link configuration files according the correct procedure (which is provided as part of the test plan template), run linkmaker to change the IOC links, reboot the IOC and follow the test procedure outlined in the test plan. If the new software checks out OK, the new version of the application is left running on the IOC. If a problem is encountered, it is a simple matter to roll back the application to its previous version by running linkmaker to change the soft link and rebooting the IOC. Once an application has been successfully installed, comments or test results are added to the test plan and it is checked off as completed. The URL for completed test plans, along with the test results, are routed to one or more of the electronic logbooks used in operations so that a record of the IOC changes is kept.

3 APPLICATION UPGRADES AND SOFTWARE ON-CALL

3.1 Application Upgrades

The procedures to upgrade an application are very similar to the procedures for the initial installation. The current version is committed to the CVS repository and tagged using the CVS Release Notes tool. A new version is created with applNewver. The application is modified and tested in the test area, if possible. Installation of the upgrade is scheduled.

Modifications are committed to CVS and the differences between the new version and the current operational version are documented using the CVS and Release Notes tool. Release notes for an application are created and accessed by version so it is easy to determine when changes to a system were introduced. Some developers do this step before the application is installed, others wait until the application has been installed before performing this step.

A test plan is written and approved. The new version is compiled and installed. On the maintenance day, the linkmaker tool is run to change the application links, the IOC is rebooted and the changes tested according to the procedures outlined in the test plan. The test results are recorded and the test plan completed and recorded in the electronic logbooks.

3.2 Software On-call

The Controls Group provides 24-hour software on-call support to operations. The software on-call person is responsible for diagnosing any operational problems which may be caused by software, fixing the problem if possible, or contacting the appropriate system developer if the problem requires an expert.

Many of the tools described above aid in this process. As mentioned previously, the CVS and Release Notes tool interfaces with the LLAPP Version Info Web browser. The Web browser displays information about the control system such as which versions of what applications are running on each IOC, which fiefdom the IOC belongs to, when each application was last updated and which developer owns the application.

In addition to access to the application description and release notes, the CVS and Release Notes tool enables developers to store “ancient history” about the application. These are release notes maintained by developers before a common interface was developed. The CVS and Release Notes tool also enables developers to store troubleshooting information for each application.

All of this information is easily accessible through LLAPP Version Info and can be used by software on-call personnel even when they are off-site. The information can be used to quickly identify applications that have recently changed, evaluate the recent changes, determine if these changes could be causing the operational problem, and identify the developer of the application.

If it becomes necessary to roll back an application during operations, the roll back information in the completed test plan along with the linkmaker tool allows problem software to be removed and working software to be restored quickly, even if the application developer is unavailable.

4 FUTURE DIRECTIONS

The IOC management tools and procedures have proven to be very useful and have improved the reliability of the IOC software management. The next step in tool development will be to integrate a relational database into the system. Currently, much configuration information is stored in flat files scattered throughout the file system and maintained through text editors. The tools described do a good job of “discovering” the IOC configuration but much improvement is needed to make the system more accurate and reliable. This task will be more easily accomplished and improved tools will be provided to software developers with the use of a relational database to store and retrieve configuration information.