



*LIGO Laboratory / LIGO Scientific Collaboration*

LIGO-T1300427-v1

*LIGO*

4/25/2013

---

LIGO CDS  
Software Development Process

---

R. Bork

Distribution of this document:  
LIGO Scientific Collaboration

This is an internal working note  
of the LIGO Laboratory.

**California Institute of Technology**  
**LIGO Project – MS 18-34**  
**1200 E. California Blvd.**  
**Pasadena, CA 91125**  
Phone (626) 395-2129  
Fax (626) 304-9834  
E-mail: [info@ligo.caltech.edu](mailto:info@ligo.caltech.edu)

**Massachusetts Institute of Technology**  
**LIGO Project – NW22-295**  
**185 Albany St**  
**Cambridge, MA 02139**  
Phone (617) 253-4824  
Fax (617) 253-7014  
E-mail: [info@ligo.mit.edu](mailto:info@ligo.mit.edu)

**LIGO Hanford Observatory**  
**P.O. Box 1970**  
**Mail Stop S9-02**  
**Richland WA 99352**  
Phone 509-372-8106  
Fax 509-372-8137

**LIGO Livingston Observatory**  
**P.O. Box 940**  
**Livingston, LA 70754**  
Phone 225-686-3100  
Fax 225-686-7189

<http://www.ligo.caltech.edu/>

## Table of Contents

|           |  |          |
|-----------|--|----------|
| <b>1</b>  | <b>Introduction</b> .....                              | <b>3</b> |
| <b>2</b>  | <b>Scope</b> .....                                     | <b>3</b> |
| <b>3</b>  | <b>Overview</b> .....                                  | <b>3</b> |
| 3.1       | <b>Process</b> .....                                   | <b>3</b> |
| 3.2       | <b>Code Development to Release</b> .....               | <b>3</b> |
| <b>4</b>  | <b>Bug Reports and Enhancement Requests</b> .....      | <b>4</b> |
| <b>5</b>  | <b>Requirements Definition and Design Review</b> ..... | <b>4</b> |
| 5.1       | <b>Weekly Meeting</b> .....                            | <b>4</b> |
| 5.2       | <b>Mailing List</b> .....                              | <b>5</b> |
| <b>6</b>  | <b>Engineering Change Request (ECR)</b> .....          | <b>5</b> |
| <b>7</b>  | <b>Code Implementation</b> .....                       | <b>5</b> |
| <b>8</b>  | <b>Code Review</b> .....                               | <b>5</b> |
| 8.1       | <b>Internal Review</b> .....                           | <b>6</b> |
| 8.2       | <b>External Review</b> .....                           | <b>6</b> |
| <b>9</b>  | <b>Code Test</b> .....                                 | <b>6</b> |
| <b>10</b> | <b>Code Documentation</b> .....                        | <b>6</b> |
| <b>11</b> | <b>Code Release</b> .....                              | <b>7</b> |

## 1 Introduction

The purpose of this document is to describe the process of software development within the LIGO Control and Data Systems (CDS) group.

## 2 Scope

The software development process described in this document is limited to software packages developed and maintained by the LIGO CDS group. The CDS software packages include:

- Real-time Code Generator (RCG): This is the standard package used by LIGO subsystem application developers to produce standard CDS real-time control and data acquisition software for use in LIGO.
- Diagnostic Test Tools (DTT): LIGO standard diagnostic tools designed to aid in system diagnosis and commission activities.
- Foton: Tool for defining control system IIR filter coefficients for use in LIGO control loops.
- Dataviewer: Control room tool used for retrieving and viewing time series data from the LIGO DAQ system.
- DAQD: The suite of software used to acquire and store data in the LIGO DAQ system.
- Network Data Server (NDS): Software package used to retrieve data from the DAQ system via a standard network interface.

## 3 Overview

At the present time, all CDS provided software packages have reached a mature state, with stable releases. They have been deployed and operational for several years at the LIGO sites, along with various university locations to provide control and data acquisition capabilities.

### 3.1 Process

This being the case, CDS software development has taken the form of software maintenance and enhancement. The procedure followed is briefly listed here, with more details provided in the later subsections of this document.

- 1) Bug report / new, or improvements to existing, feature request submitted to CDS Bugzilla site.
- 2) Assignment of bug fix/feature request.
- 3) Review at weekly CDS software meeting.
- 4) Engineering Change Request (ECR)
- 5) Code implementation.
- 6) Code Test
- 7) Code Documentation
- 8) Create new code release, with updated documentation and Document Change Notice (DCN).

### 3.2 Code Development to Release

All CDS software is maintained in a separate SVN repository. The general flow of software from implementation is release is:

- Develop new software in the “trunk” area.
- Once code planned for a new release is ready, the trunk is copied to a branch.
- Final testing is performed on the software included in the branch.
- The branch is copied to a release area as a major release (X.X). Source code in a release is never changed.
- If a bug is reported on a release, and deemed of a critical nature, then the branch from which the release was made is modified and tested. The branch is then copied to a minor (bug fix) release (X.X.X).

## 4 Bug Reports and Enhancement Requests

For CDS software, all bug reports and enhancement, new feature requests are made via a CDS Bugzilla system. Detailed information on its use is documented in LIGO-T1000496 (<https://dcc.ligo.org/LIGO-T1000496-v1>). This system allows for the tracking of requests, from initial report and information provided, through to resolution.

Once entered into the system, an email is sent to various CDS staff members. A priority and staff member is assigned to the request for further investigation. If a bug or feature request is determined to be critical in nature, then the implementation is immediately assigned to a software developer for resolution.

## 5 Requirements Definition and Design Review

Once requests are entered into Bugzilla, two forums are provided for further discussion, further refinement of requirements, and review of code implementation details. These are:

- Weekly CDS software meetings
- CDS mailing list

### 5.1 Weekly Meeting

To promote the exchange of information on CDS code development and provide a forum for code requirement/design reviews, a CDS software meeting is held weekly. Regular attendees include:

- CDS software developers
- CDS site system administrations
- Various aLIGO system group leaders, in particular those with responsibility for controls implementation for their system.
- Users of CDS provided software tools.
- Members of the Detector Characterization group

Topics include:

- Status of bug fixes and code enhancements.
- Follow up on bug fix or enhancement requirements, if not sufficiently detailed in Bugzilla.
- Periodic presentations on new software development activities, particularly implementing new feature requests to help verify that requirements are being met and/or discuss changes.
- Schedule for upcoming code releases and their contents.

- Problems and issues

## 5.2 Mailing List

To include a broader group than attends the weekly meeting, a `cds_announce` mailing list is provided that LSC members can subscribe to. Presently, there are about 100 subscribers to this list, which include various LIGO management and staff members, as well as members of other institutions and facilities that employ the LIGO CDS software in their control systems.

This mailing list provides various functions:

- Dissemination of information from the weekly meetings, particular where a larger community should have input on requirements and new code designs.
- Information on new code releases and/or documentation.
- “Tech talk” line, where various users of the CDS software packages can request information from other users and/or code developers on code usage for their particular application or problems encountered and possible work arounds.

## 6 Engineering Change Request (ECR)

Rather than develop a separate system for software changes, the LIGO ECR process, used by engineering groups, has been adopted. However, to avoid flooding the ECR system, it is implemented for software in a slightly different fashion. Rather than submitting an ECR for each bug fix or new feature request, an ECR would be opened for each new planned code release, major and/or minor. In this ECR, all changes to be included in that release are written up in the ECR. This ECR is then submitted, with an expected delivery date and code release number.

## 7 Code Implementation

Once the code requirements and implementation design are reviewed in the weekly meeting, a software developer is assigned, along with the task priority. Where the code changes are implemented is dependent on the nature of the request, ie bug fix or code enhancement.

Code enhancement and new feature requests are only developed for and checked in with the latest CDS SVN source code repository, or “trunk” area, and scheduled for the next major release.

Where bug fixes are made is primarily based on the criticality of the code error:

- Critical to systems presently running latest, or earlier, releases of CDS software:
  - Software patched in appropriate SVN code branch(es), and scheduled for next bug fix release.
  - Software in latest CDS SVN trunk area also repaired, if applicable.
- Non-critical to presently operating systems
  - Software in latest CDS SVN trunk area repaired, if applicable.

## 8 Code Review

CDS code reviews are of two types: internal and external.

## 8.1 Internal Review

Once new code has been developed, or a bug fix made, and preliminary testing is done, a second CDS software developer is assigned to do a line-by-line review of the software. Any necessary code iterations are made, then code made available for formal testing.

## 8.2 External Review

External reviews ie reviews of CDS software by non-CDS and non-LIGO staff, are to be held on a periodic basis. The last such review was held in September of 2012. Unlike the detailed internal reviews, these reviews are intended to review CDS software at a more global level.

## 9 Code Test

Prior releases of CDS software (2.6 and earlier) were tested once released to a branch. This testing was performed by LIGO site staff by hand and written up in code release test reports.

For future releases, starting with V2.7, the testing process is being automated through the use of test scripts running on the LHO and LLO DAQ test systems. More information on this process is included in LIGO-T1000561 CDS Software Test Procedure ( <https://dcc.ligo.org/LIGO-T1000561-v2> ).

Following this plan, a test script is developed for new software, and/or modified, as necessary to test bug fixes, and implemented to run on one of the test systems. These test scripts have a standard format, such that they can produce a standardized test report. This test report is written to file in a form that may they be read in by the CDS code documentation tool, doxygen (see next section), and included as part of the documentation for a code release.

## 10 Code Documentation

The CDS group has standardized on a publicly licensed software package, doxygen, for the purposes of producing software documentation. Beginning with code release V2.7, a ‘make doc’ feature has been added, along with hand written doxygen configuration and information files, to produce:

- Overview text information, diagrams, links to DCC documentation, and the like to guide CDS software users and developers.
- Code documentation from CDS C and C++ software. This is a work in progress which primarily entails changing standard C/C++ comment lines to doxygen format.
- Code documentation from CDS Perl scripts, primarily used in the RCG parser functions. Since standard Perl comment lines cannot be directly read from Perl code, a filter has been developed to pre-process Perl comments for later inclusion into doxygen.

The output of doxygen is provided in two forms: html pages and Rich Text Format (RTF). The html pages will be included as part of the code release and the RTF output used to generate documentation for a release for submission to the LIGO DCC.

## 11 Code Release

Once an ECR has been approved and code testing in the appropriate SVN branch is completed, the code is ready for release:

- Make doc is run on the branch to produce the final set of documentation, including test reports.
- Test reports are checked in as part of the branch such that they will travel with the release.
- The SVN NEWS file is verified to be complete. This file contains release to release change information.
- The code branch is copied to a unique code release in SVN.

As the final step of this process, a standard LIGO Document Change Notice (DCN) is submitted. The DCN includes all documentation that has been added/changed in the process of executing the associated ECR.