



CruiseControl is a framework for a continuous build process. It can be used as a tool to monitor the changes made to a source code repository. The latest version supports a wide variety of plug-ins, and is easy to install and use.

**N**

o matter what sophisticated source code version control tools you use, monitoring the changes made to your project source code repository is always vital to the success of your project. Monitoring becomes even more important when there are multi-site teams involved in project development. Even though most version control tools have some in-built notification mechanisms, they are either not feasible for use or consume too much time. In this article, we shall see how you can easily monitor the changes made to the ClearCase based source code repository using a freely available tool called CruiseControl.

In case you haven't heard, ClearCase is a software configuration management system from Rational Software, Inc. It keeps track of the versions of files used to build each release of a software product and it helps organise the work of groups of engineers.

# CruiseControl

## Monitor Code Repositories With Ease



CruiseControl is a framework for a continuous build process. It can be used as a tool to monitor the changes made to a source code repository. The latest version supports a wide variety of plug-ins—including plug-ins for ClearCase, CVS, FileSystem, etc. You can get the complete list of the modification set plug-ins from [http://cruisecontrol.sourceforge.net/main/configxml.html#modificationset.]

CruiseControl can be downloaded free of cost from [http://cruisecontrol.sourceforge.net/download.html.] In order to use CruiseControl, you also need the ANT tool from Apache (discussed in another LFY article this month).

## Installing and running CruiseControl

The installation of CruiseControl is fairly simple. First, extract the CruiseControl zip file to your system's hard disk. The downloaded file will not have the executable JAR file, so you need to build it using the ANT tool. Edit build.xml under the main directory and remove the test targets, as we don't need them, and run the ANT tool. If everything goes well, the executable JAR file—cruisecontrol.jar—will be created under the appropriate directory.

To run CruiseControl, all that you need to do is to define two xml files for your project—config.xml and build.xml and pass it to the CruiseControl tool.

The code for a sample build.xml is shown below. As we don't plan to use CruiseControl to build our source code repository, we define a dummy target *build*. CruiseControl will call this target whenever it detects a change in the source code repository.

```
<project name="cc-helper" default="build" basedir=".">
  <target name="build">
    <echo message="Nothing to build..." />
  </target>
</project>
```

The next important file is config.xml. This should contain all the required configuration details for CruiseControl. A sample configuration file is shown below:

```
<cruisecontrol>
  <project name="LFY" buildafterfailed="false">
    <modificationset requiremodification="true"
      quietperiod="60">
      <ClearCase branch="lfy_dev"
        viewpath="/vob/lfy_dev">
      </ClearCase>
    </modificationset>

    <schedule interval="14400"> <!-- 4 hours -->
      <ant />
    </schedule>
  </project>
</cruisecontrol>
```

```
<publishers>
  <htmlmail mailhost="email.com"
    returnaddress="rajark@email.com"
    skipusers="true"
    logdir="logs/LFY"
    xslfile="modifications.xsl"
    subjectprefix="File Modification Report - ">
  <always address="rajark@email.com" />
  <failure address="rajark@email.com" />
</htmlmail>
</publishers>
</project>
</cruisecontrol>
```

The configuration file shown above consists of three sections—modificationset, schedule and publishers.

The modificationset is where you specify your source code repository details. In the above example we instruct the CruiseControl to detect changes made to the ClearCase source code repository (ClearCase VOB). Under ClearCase modificationset, you need to specify the branch and view path details. In the above example, the branch name is *lfy\_dev* and view path is */vob/lfy\_dev*

The schedule section is where you specify the monitor interval. In the above example, we instruct CruiseControl to monitor the ClearCase VOB every four hours. Whenever CruiseControl detects changes to the source code repository, it will execute the instructions enclosed in the schedule section. In the above example, we instruct CruiseControl to run the ANT tool, which will in turn (by default) call our build.xml with the default target *build*.

Next in line is the *publishers* section. This is where you specify the notification reporting details. In the above example, we instruct CruiseControl to notify through the e-mail notification mechanism.

Once you define the build.xml and config.xml files, you can start CruiseControl using the shell script *cruisecontrol.sh* under the main/bin directory.

Figure 1 shows a sample e-mail notification from CruiseControl.

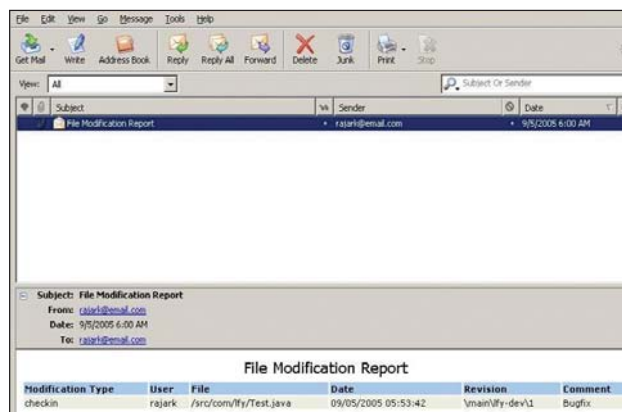


Figure 1: A sample e-mail configuration from CruiseControl

In Figure 1, you can see that the user *rajark* has checked-in a file */src/com/lfy/Test.java* in *\main\lfy-dev\1* branch.

## Behind the scenes

So far we have learnt how to use CruiseControl from a user perspective. Now let's see what it does behind the scenes to detect the changes made to your ClearCase source code repository. Internally, CruiseControl uses the following command to detect changes:

```
cleartool lshistory -branch <branch_name> -r -nco -since
<date_time> -fmt <format_string>
```

The ClearCase *lshistory* command is used to get list event records for VOB-database objects. *The branch* option specifies the branch name and the *-r* option is an instruction to process the sub-directories. The *-nco* option instructs is an instruction to exclude files/directories that are in a checked-out state. The *-since* option instructs cleartool, the primary command-line interface to ClearCase, to list the changes recorded since the specified date-time. The *-fmt* option instructs the cleartool to list the records in the specified format. A sample format string is shown below:

```
%u %Nd %En %Vn %o %l %Nc\n
```


Where:

- %u lists the login name of the user
- %Nd lists the date and time in the numeric format

- %En lists the element name (file or directory name)
- %Vn lists the version
- %o lists the operation type (usually the name of the cleartool subcommand—say checkin, checkout, etc)
- %l lists the label name attached to an element (if any)
- %Nc lists the comment string

Example output is listed below:

```
rajark 20051130.061223 Test.java \main\lfy-dev\1 checkin Code
Refactored
```

Monitoring the source code repository in such a fashion not only helps project managers in handling and controlling the project source code, but also helps the project team members to detect the changes that could affect their modules in the long run. CruiseControl is easy to install and use, and presents information in an easy to manage format. It is, truly, the best friend of a project manager. 

## REFERENCES

- ClearCase : <http://www-306.ibm.com/software/awdtools/clearcase/>
- CruiseControl : <http://cruisecontrol.sourceforge.net/>
- The GUI for CruiseControl configuration : <http://cc-config.sourceforge.net/>
- The Apache ANT tool : <http://www.apache.org/dist/ant/>

*By: Raja R.K. The author is a lead engineer with HCL Technologies, Chennai.*

**Some of you will spend THOUSANDS to learn OLD technologies**

**The smarter ones will grab every opportunity to master the future**



**Read LINUX For You**

For more info, log on to:  
[www.linuxforu.com](http://www.linuxforu.com)



ASIA'S FIRST  
**LINUX**  
MAGAZINE

