C-JDBC Developer's Guide

Emmanuel Cecchet

Julie Marguerite

Mathieu Peltier

Version 1.0b9+

Table of Contents

1. Getting the Source Code

1.1. Source distribution

The C-JDBC source distribution can be downloaded from the C-JDBC's Web site (http://c-jdbc.objectweb.org/). It includes the source code of both C-JDBC driver and controller as well as the test suite and all tools.

The following three formats are available (where `x.y` is the C-JDBC release number):

- `c-jdbc-x.y-src-installer.jar`: Java™ graphical installer (powered by IzPack (http://www.izforge.com/izpack/)).
- `c-jdbc-x.y-src.tar.gz`: source distribution for the Unix platforms users.
- `c-jdbc-x.y-src.zip`: source distribution for the Windows platforms users.

The easiest way to install the source distribution is to use the Java graphical installer[1]. If you want to use the other distribution formats, just uncompress the downloaded file in the directory of your choice.

> **Note:** Nightly builds releases are also available for download. These releases are regenerated each night at 23:45 PM CEST.

1.2. CVS Access

The whole C-JDBC codebase can be checked out through anonymous access using any CVS (Concurrent Versions System) client. The CVS server to use is `cvs.c-jdbc.forge.objectweb.org` (in `pserver` mode). The C-JDBC repository is named `/cvsroot/c-jdbc`. Unix users can use for example the following commands:

```
bash> cvs -d:pserver:anonymous@cvs.c-jdbc.forge.objectweb.org:/cvsroot/c-jdbc login
bash> cvs -d:pserver:anonymous@cvs.c-jdbc.forge.objectweb.org:/cvsroot/c-jdbc co .
```

When prompted for a password, just hit enter.

The available modules are:

- `c-jdbc`: used to get the C-JDBC codebase.
- `web`: used to get the C-JDBC Web site.
- `.`: used to get the whole C-JDBC repository.

> **Note:** The latest snapshot of the CVS repository is also available for download. The files to download are: `c-jdbc-current-cvs.tar.gz` or `c-jdbc-current-cvs.zip`. These are automatically regenerated each night at 00:00 AM CEST.

1.3. Distribution Organization

C-JDBC source code comes with an Ant (http://ant.apache.org/) build file. You will need **ant** (release 1.5 or greater) to build C-JDBC. You can get Ant and its documentation from the Apache's Web site (http://ant.apache.org/).

Tunable build properties are set in the build.properties file in the root directory of the distribution. The only property you usually have to set is the installation directory (property install.dir) where you want to install the distribution once it is compiled.

ant clean dist will build both source and binary distribution files in the build directory.

ant clean install will build a C-JDBC binary distribution and install it in the directory specified by the install.dir property defined in the build.properties file.

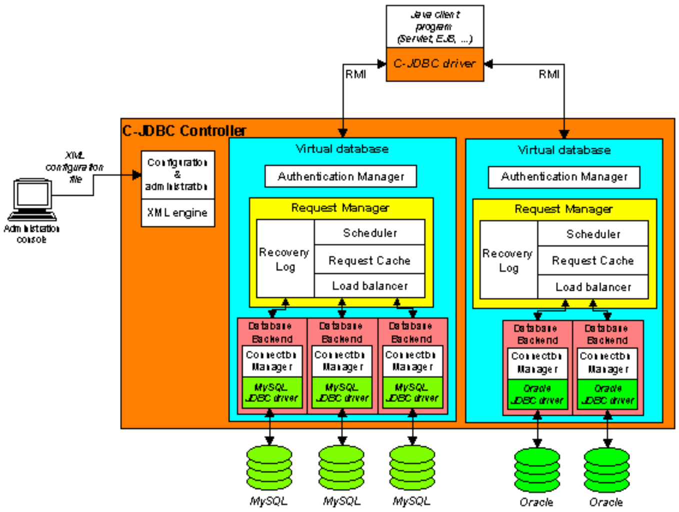
You can get an extensive list of the ant targets using the **ant -projecthelp** command. Here is an overview of the available targets:

- **clean**: deletes the ${build} and ${dist} directory trees and emacs backup files.
- **compile**: compiles the C-JDBC driver, controller and request player (the **compile-controller**, **compile-driver**, **compile-requestplayer** and **compile-tests** sub targets compile only the C-JDBC controller, driver, request player and JUnit (http://www.junit.org/) test suite, respectively).
- **dist**: creates binary and source C-JDBC distributions (tar.gz, zip formats and Java graphical installer powered by IzPack (http://www.izforge.com/izpack/)). The **dist-bin** and **dist-src** sub targets create only binary or source distributions, respectively.
- **doc**: generates documentation in various format from DocBook (http://www.docbook.org/) sources. Warning, this target requires the following tools: **make**, **LaTeX**, **jade**, **jadetex**, **dvips**, **dvipdf** and should only work under Unix platforms. Please refer to the specific documentation of your system to known how to install them.
- **doccheck**: checks the Javadoc syntax and produces detailed errors report, using the Sun Doc Check Utilities (http://java.sun.com/j2se/javadoc/doccheck/).
- **install**: installs C-JDBC distribution in the directory specified by ${install.dir}
- **jar**: creates JAR files for C-JDBC driver, controller and request player (the **jar-controller**, **jar-driver** and **jar-requestplayer** sub targets create only C-JDBC controller, driver and request player JAR files, respectively).
- **javadoc**: generates Javadoc documentation.
- **test**: runs all JUnit tests (warning junit.jar and xalan.jar must be found in ${ANT_HOME}/lib).
- **test-gui**: launches the GUI test runner of JUnit.
- **test-unit**: runs one JUnit test specified by -Dtest.class=<class name> (warning junit.jar and xalan.jar must be found in ${ANT_HOME}/lib).

2. Design Overview

To be documented.

Figure 1. C-JDBC controller design overview example



3. Architecture Overview

To be documented.

4. Test Suite

> Here's the way testing goes:
>
> ```
> becomeTimidAndTestEverything
> while writingTheSameThingOverAndOverAgain
>   becomeMoreAgressive
>   writeFewerTests
>   writeTestsForMoreInterestingCases
>   if getBurnedByStupidDefect
>     becomeTimidAndTestEverything
>   end
> end
> ```
>
> The loop, as you can see never terminates.
>
> —JUnit FAQ (http://www.junit.org/doc/faq/faq.html)

To be documented.

5. Code Conventions

C-JDBC uses the Sun's Java standard code conventions (http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html) with little modifications (for example, braces in statements, ...). For more details, please see the source code itself.

Eclipse (http://www.eclipse.org/)[2] users can checkout from the C-JDBC CVS repository a property file containing all the Eclipse Code Formatter settings to suite with the C-JDBC code conventions. This file is named `.codeFormatter.properties` in the root of the CVS repository. Unfortunately it can not be imported in Eclipse (version 2.1): you have to change manually the Eclipse Code Formatter preferences.

The CVS repository also contains configuration files for the Checkstyle (http://checkstyle.sourceforge.net/) Eclipse plugin (2.1 and 3.1 versions). This plugin allows you to check code conventions and Javadoc automatically in Eclipse and provides the user with many facilities to correct errors. These files are named `.checkstyle.plugin.config.x.y` in the root of the CVS repository and can be imported to modify automatically the Checkstyle preferences (open "Window" menu, then choose "Preferences" and click to "Checkstyle"). Finally the repository contains the standard Checkstyle Eclipse plugin configuration file: `.checkstyle` to use with the 3.1 version. To use it, make sure you have first imported the configuration file.[3].

6. C-JDBC release step by step

Here are the steps to follow when releasing a new version of C-JDBC:

1. Copy the `c-jdbc.dtd` to `web/html/dtds/c-jdbc-VERSION.dtd` (example is `web/html/dtds/c-jdbc-1.0b8.dtd`).
2. Update the version number in `build.properties`.
3. Add the release name, date and CVS tag in `HISTORY.txt`.
4. Update `web/root/history.xml` with information from `HISTORY.txt`.
5. Add a news in `web/root/index.xml`.
6. Update C-JDBC current version number in `web/root/index.xml`.
7. Build a distribution using **ant dist**.
8. Commit all files in CVS repository.
9. Add a tag in CVS according to match what is described in `HISTORY.txt`.
10. Upload distribution files to the ObjectWeb Forge site using: **scp dist/* upload.forge.objectweb.org:incoming/**
11. Upload the documentation files and rename them with the version number. Here is an example for version 1.0b6:

```
scp build/doc/userGuide/userGuide.ps upload.forge.objectweb.org:incoming/userGuide-1.0b6.ps
scp build/doc/userGuide/userGuide.pdf upload.forge.objectweb.org:incoming/userGuide-1.0b6.pdf
scp build/doc/developerGuide/developerGuide.ps upload.forge.objectweb.org:incoming/developerGuide
scp build/doc/developerGuide/developerGuide.pdf upload.forge.objectweb.org:incoming/developerGuid
```

12. Add a new binary, source and documentation release on the ObjectWeb Forge site (https://forge.objectweb.org/project/admin/editpackages.php?group_id=42).
13. Copy/paste the HISTORY.txt in the release notes for binary, source and documentation releases.
14. Add the corresponding files and edit the file types for each of them (.tgz, .pdf, ...).
15. Finally send a email release notice for each release (step 4 on Forge page).
16. Post a news on C-JDBC's Forge front page.
17. Upload the static documentation on the web site using **scp -r build/doc jupiter.objectweb.org:/home/web/c-jdbc/not_in_cvs/current/**
18. Log on jupiter.objectweb.org and rebuild the web site using **cd /home/web/bin/ ; ./sudo-website-update.sh c-jdbc**
19. Announce the new release on freshmeat.net.

Notes

1. . A Java Virtual Machine™ is of course needed in this case.
2. Eclipse is a popular open source IDE for Java.
3. The CVS repository contains also standard Eclipse project files that can be used to manage a `c-jdbc` project (`.project` and `.classpath`).