

# CDA Content Evaluation Development: Getting Started

## Scope

The purpose of this document is to describe how a new developer gets started with the CDA Content Evaluation software.

## Software Components

You need to install and configure software on your system as described in this article: <https://docs.impHub.org/x/RQAq>

The software and environment consists of these components:

- Sun (Oracle) Java 7
- PostgreSQL database
- A Java Servlet container, normally Apache Tomcat 7.x
- Java Server Faces 2 (JSF 2)
- X Faces add-on to JSF
- XHTML and Java code written for this project
- Eclipse IDE for Java EE for development/testing

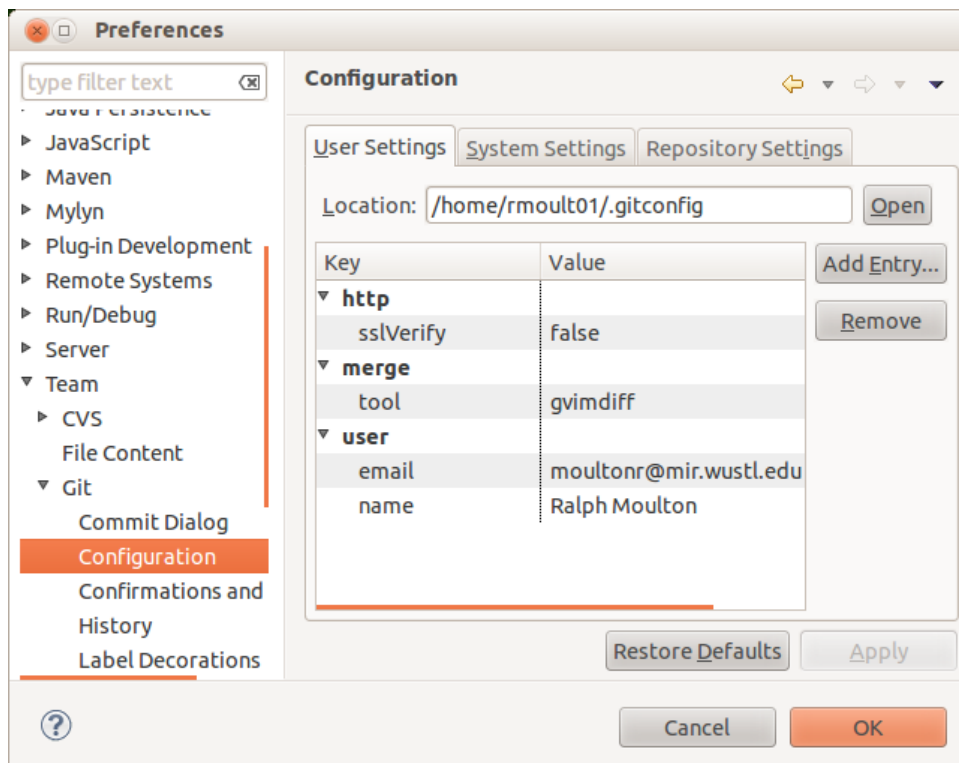
When the application is first run, it detects if the postgresql database exists and creates it if necessary.

## Clone the GIT Repository


### Standard Procedure

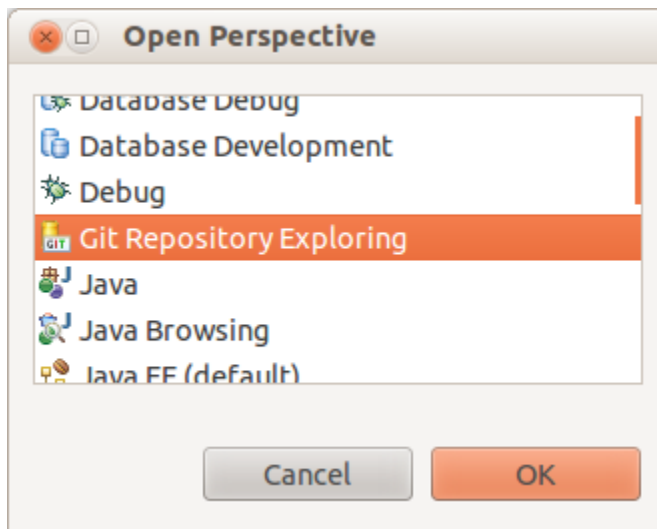
This assumes you are starting with a system that does not have an Eclipse workspace and does not have a clone of the repository. Please note that the workspace folder will remain separate from the repository clone on your system.

1. Start Eclipse; it will request a folder name. Our practice is to use a folder name that makes it clear this is an Eclipse workspace. For example:
  - a. cdacontentevaluation-workspace
2. Edit the Git configuration.
  - a. From the Eclipse menu bar, select View > Preferences (on a Mac, Preferences is under the Eclipse label; there is no View menu label).
  - b. In the Preferences dialog left side menu tree, open Team > Git > Configuration.
  - c. In the Configuration dialog, select the User Settings tab (may already be selected).
  - d. Edit the key-value pairs to look like this (your name and email, of course):



3. Close the Welcome window and open the Git Repository Exploring perspective.

- a. Click on the Open Perspective icon:  which is in the upper right area of the screen.
- b. In the Open Perspective dialog, select Git Repository Exploring and Click OK.



4. The left panel of the Git Repository Exploring perspective contains the Git Repositories tab. Since this is a new project and has no identified Git Repositories, the following links will appear in the middle of the tab:

Select one of the following to add a repository to this view:

-  [Add an existing local Git repository](#)
-  [Clone a Git repository](#)
-  [Create a new local Git repository](#)

5. Click on "Clone a Git repository" link, which will start the wizard. Fill in the first dialog as shown, using your own imphub login and password, and click Next:

<https://code.imphub.org/scm/cda/cdacontentevaluation.git>

**Clone Git Repository**

**Source Git Repository**  
Enter the location of the source repository.

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

User:

Password:


Store in Secure Store ☐

6. In the Branch Selection dialog, select the master branch and click Next.

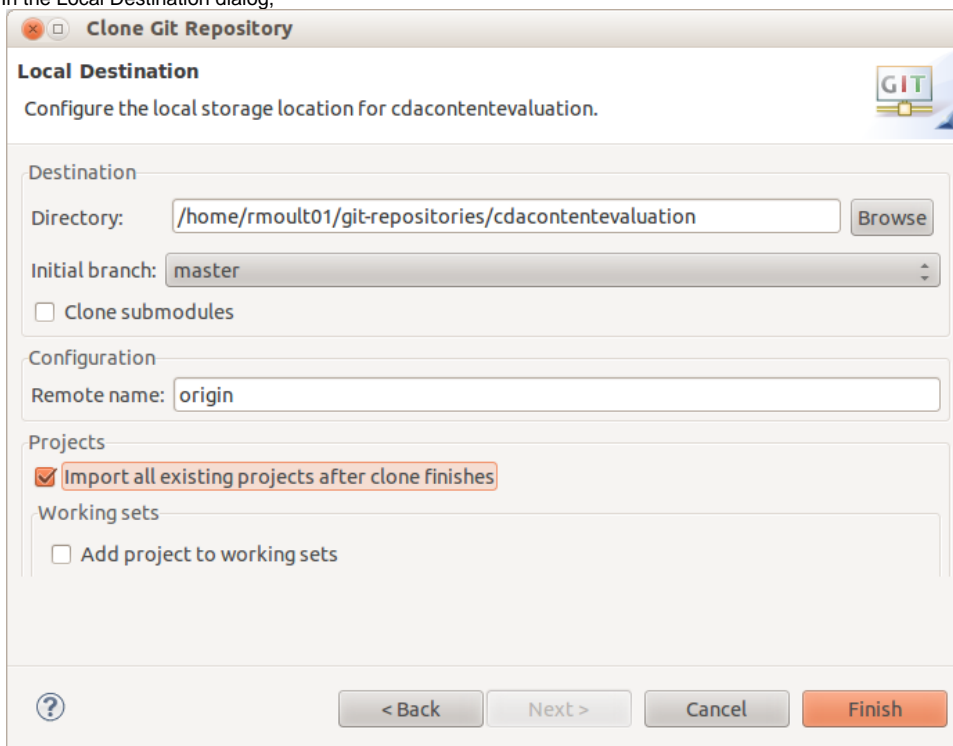
**Clone Git Repository**

**Branch Selection**  
Select branches to clone from remote repository. Remote tracking branches will be created to track updates for these

Branches of https://code.imphub.org/scm/cda/cdacontentevaluation.git:

☒  master

7. In the Local Destination dialog,



The next form indicates where you

will place your git repository.

- a. Enter the directory where your local Git repository will be placed. This **should not** be in your workspace.
  - b. Check: Import all existing projects after clone finishes.
  - c. Click Finish
8. The wizard will clone the Git repository to your local system and import the CDA Content Evaluation project to Eclipse.
9. Switch to the Java EE perspective.
10. Update Eclipse with the project maven configuration.
- a. In the Project Explorer tab, right click on the CDACContentEvaluation line.
  - b. On the context menu, select Maven > Update Project...
  - c. In the Update Maven Project, make sure you select the one project (check box is not enough), and the click OK.
11. Right click on the CDACContentEvaluation line
- a. Run as Maven build (not Maven build ...)
  - b. Set goals: clean package install
  - c. Run the application
  - d. This should build several war files in the target folder. If this fails, it might be due to a non-existent folder. In Eclipse, make sure the folder "src/profiles/dev" exists. The dev folder should be parallel to the folders eu and na.

## Attach Tomcat to Eclipse

## Run the Application