



# BlueJ Teamwork Repository Configuration

Version 2.0  
for BlueJ Version 2.5.0 (and 2.2.x)

Davin McCall

School of Engineering & IT,  
Deakin University

# 1 Introduction

This document gives a brief description of how you might set up a version control repository for use with BlueJ's teamwork features. It is intended mainly as a "quick start" guide and not as a complete reference – for that you should refer to the version control software documentation (i.e. the CVS manual or the Subversion manual) – but it does explain some BlueJ-specific concepts (such as how BlueJ supports the notion of student groups or teams).

Setting up a repository usually requires a server to which you have "root" or administrator access. This may mean that you need to ask a Systems Administrator to set up the repository for you.

Since BlueJ version 2.5.0, both Subversion and CVS are supported version control systems. BlueJ version 2.2.x supports only CVS. BlueJ versions prior to 2.2.0 did not support teamwork features.

Chapters 2 and 3 explain how to set up and test a repository using CVS. Chapter 4 then covers the equivalent steps for using Subversion.

## 2 Setting up a simple single user CVS repository for testing the BlueJ teamwork features

### 2.1 Setting up the repository server

#### On Unix / Linux / MacOS X:

You must have the CVS software installed on the machine you intend to use as a server. There is a good chance that it is already installed, but if not, your vendor or distribution provider will almost certainly provide packages that can be installed.

Setting up a repository is quite simple. At a command line, type:

```
cvs -d /Users/fred/cvsroot init
```

Here, "/Users/fred/cvsroot" is the CVS root directory, in which all files needed to maintain the repository will be stored. Replace this with a valid path on your own system. The directory should not previously exist; it will be created automatically by this command.

Your server must provide ssh access for clients (such as BlueJ) to connect to it. On most systems, ssh access is enabled by default, so you can now share projects using BlueJ. (On MacOS, you may have to enable 'Remote Login' in the *Sharing* preferences.)

Remember the path of the directory you used in the "cvs init" command above. You

will need to specify this in BlueJ as the “Repository path”.

Continue with section 1.2 of this document to test your server.

### **On Windows:**

The suggested CVS server software is "CVSNT". You can download it free of charge from <http://www.cvsnt.org>. Under the "free download" column, select Windows as your operating system.

Log in as Administrator and execute the downloaded installer in order to begin the installation. Once this is complete, you may be asked to restart the computer.

After CVSNT has been installed, you can set up a repository using the CVSNT control panel, which you can access from the start menu (Start / Programs / CVSNT / CVSNT control panel). In the "Repository configuration" tab, press the "add" button, and choose a directory for the CVS repository (it is recommended to create a new directory). You will be asked whether CVSNT should initialise the location as a repository, and you should choose "yes". The location "name" defaults to the repository path without the drive letter designation, which is generally a good choice.

CVS offers two access methods to the repository, *pserver* and *ssh*. By default CVSNT provides *pserver* access to the repository. If you want to use the *ssh* access method (which we recommend as it is more secure) you need to install an ssh server program. One such program can be found at the following location:

<http://sshwndows.sourceforge.net/>

Remember the repository "name" – you will need to specify this in BlueJ as the “Repository Path” to connect to this repository.

You can now proceed to test your server.

## **2.2 Testing the server**

In BlueJ, switch on the teamwork controls (in *Preferences/Miscellaneous*) and select *Share this Project* from the *Tools/Team* menu. Specify your login details; leave the “group” blank.

Choose CVS as the server type (BlueJ 2.5.0+ only). Specify the server location, using the name of the server, and the repository path as mentioned above. Use “ext/ssh” as the protocol if you have ssh installed, otherwise use “pserver” (the server must be configured to support pserver access).

Using BlueJ on a different machine, you can now access this project using *Team/Checkout this Project* and providing the same details.

### **3 Creating an access-controlled CVS repository for use by groups**

There are many possible configurations that could be used when configuring a version control system such as CVS for use in a teaching environment. There are many factors that will influence the choices made. Many of these will relate to the current teaching computer resource infrastructure at each educational institution. It is therefore difficult to suggest, recommend or mandate one specific version control configuration that will work for everyone.

This document provides an overview of how CVS could be configured to allow the use of BlueJ's teamwork features by students for group projects.

The design of the teamwork facilities in BlueJ is based on the assumption that students will work on projects in groups. Each member of a group will have access to the group's own projects, but not to projects belonging to other groups. In fact, each group will have access to their own repository, but not to the repository of other groups - this means that students can only see the projects to which they have access; it also means that the project that each group works on may have the same name.

In CVS terminology, each group has a different "repository root". Each project is a "module" within the group's repository.

#### **3.1 Configuring a server on Unix**

You will need a server to which you have root access.

Choose a location on the server for the repositories. This should be a directory (such as `/cvsroot`) on the server which is otherwise unused. This directory is the "repository prefix" (in the team settings dialog it is called the "Repository Path"). The directory must have execute permission set for all students. One way to achieve this is to give all users execute permission:

```
chmod o+x /cvsroot
```

(assuming `/cvsroot` is the repository prefix).

Inside the prefix directory, create a subdirectory for each group. The name of each subdirectory will be the name of the group, and must be provided by the group in the BlueJ team settings dialogue.

Each student in a group should have read, write and execute permissions on the group's repository directory, and not to any other group's repository directory. One way to achieve this is to create a Unix group for each teamwork group, and make each student in the group a member of the Unix group.

Then the Unix group of each group's repository directory can be set to the appropriate group:

```
chgrp groupa /cvsroot/groupa
chgrp groupb /cvsroot/groupb
chgrp groubc /cvsroot/groupc
... etc
```

The permissions of each of the directories should be changed appropriately. This can be done with a single command:

```
chmod o-rx,g+rws /repository/*
```

This allows students in a group to access the group repository, but denies access to students not in the group. It also ensures that all files created in a repository will belong to the correct group.

For each group's directory, initialize the repository, by executing the following command:

```
cvs -d /cvsroot/groupa init
```

For ssh access to the repository, there is nothing else to be done (other than ensuring students do have ssh access to the server). For pserver access, it is necessary to configure the "inetd" (or equivalent) program to run cvs in pserver mode when a remote connection is established. For more information see the CVS manual page on "Setting up the server for password authentication", here: [http://ximbiot.com/cvs/manual/cvs-1.11.22/cvs\\_2.html#SEC30](http://ximbiot.com/cvs/manual/cvs-1.11.22/cvs_2.html#SEC30)

Note, when configuring the inetd program to launch cvs (for pserver access), that each of the group repositories must be an allowed root.

## 4 Subversion repositories

Setting up a subversion repository is a similar process in many ways to setting up a CVS repository. For this reason, you should read over the previous chapters even if you intend only to use Subversion. This chapter then explains the differences between CVS and Subversion, in terms of setting up a repository and accessing it using BlueJ.

### 4.1 Creating the repository

You need a server with the Subversion software installed. The Subversion command to create a repository is:

```
svnadmin create /path/to/repository
```

Here "/path/to/repository" is a directory where the repository will be created; the directory itself should not exist (but the parent directory should). Naturally, on

Windows, the path should be a Windows-style path with a drive specifier and using backslashes instead of the forward slashes in the example above.

## **4.2 Repository access methods**

Like CVS, Subversion supports several access methods. The “svn” method is the equivalent of CVS’ pserver; “svn+ssh” is the equivalent of CVS’ ext method. Subversion also supports access via http (using the WebDAV protocol). The simplest method to configure on the server end is svn+ssh as, other than ssh access, it requires no further configuration, and provides good security as well as high configurability. For information on configuring the other access methods please consult the subversion documentation.

## **4.3 Testing the server**

This is done in exactly the same way as for CVS (see Chapter 2.2) however you must select “Subversion” as the server type.

## **4.4 Group support**

Group support with Subversion works in almost exactly the same way as for CVS (see Chapter 3) – you need a separate repository for each group. If using the svn+ssh access method, you can use unix groups to limit access as described in Chapter 3; for the svn access method you need to configure authentication and access rights according to the Subversion documentation; for the http method, access is controlled by configuring the web server (usually Apache Httpd).