

.....

## **Apache Continuum 1.4.3-SNAPSHOT**

.....

**The Apache Continuum Project**



## Documentation

---

1. Documentation .....	i
2. Index (category) .....	1
3. Getting Started .....	2
4. Installation/Upgrade Guides .....	4
5. Installing Standalone .....	
6. Installing in Tomcat .....	
7. Upgrade .....	
8. Installing a Build Agent .....	16
9. User's Guides .....	18
10. Managing Projects .....	19
11. Add a Project .....	20
12. Edit a Project .....	28
13. Remove a Project .....	31
14. Working Copy .....	32
15. Managing Build Definitions .....	33
16. Managing Notification .....	35
17. Mail Notification .....	37
18. IRC Notification .....	39
19. Jabber Notification .....	41
20. MSN Notification .....	43
21. Wagon Notification .....	44
22. Building a project .....	46
23. Scheduled Build .....	47
24. Forced Build .....	48
25. Build Results Management .....	49
26. Cancelling builds .....	51
26. Release Management .....	52
26. Prepare Project Release .....	54
26. Perform Project Release .....	57
26. Release Results Management .....	59
26. Project Builds Report .....	60
26. Administrator's Guides .....	62
26. Managing Users and Security .....	63
26. Security Configuration .....	64
26. LDAP Configuration .....	65
26. Managing Project Groups .....	66
26. Managing Installations .....	69
26. Managing Build Environments .....	

26. <b>Managing Schedules</b> .....	72
26. <b>Managing General Configuration</b> .....	75
26. <b>Managing Local Repositories</b> .....	77
26. <b>Managing Purge Configuration</b> .....	79
26. <b>Managing Parallel Builds</b> .....	81
26. <b>Managing Build Queues</b> .....	82
26. <b>Managing Build Agents</b> .....	84
26. <b>Managing Build Agent Groups</b> .....	85
26. <b>Managing Project Queues</b> .....	86
26. <b>External databases</b> .....	88
26. <b>Monitoring Continuum</b> .....	91
26. <b>Log Files</b> .....	94
26. <b>Audit Logs</b> .....	95
26. <b>Continuum Logs</b> .....	96
26. <b>Security Logs</b> .....	97
26. <b>Appearance Configuration</b> .....	98
26. <b>Build Definition Template</b> .....	100
26. <b>Shutdown Continuum</b> .....	101
26. <b>Understanding Distributed Builds</b> .....	102
26. <b>WebDAV for Agent Working Copies</b> .....	105
26. <b>Developer's Guides</b> .....	106
26. <b>Building Continuum</b> .....	107
26. <b>XML-RPC</b> .....	108

# 1 Index (category)

---

## 1.1 Welcome to Apache Continuum

Apache Continuum is an enterprise-ready continuous integration server with features such as automated builds, release management, role-based security and integration with popular build tools and source control management systems. Whether you have a centralized build team or want to put control of releases in the hands of developers, Continuum can help you improve quality and maintain a consistent build environment.

### 1.1.1 Documentation

- [Release Notes](#)
- [Getting Started](#)
- [Installation and Upgrade Guide](#)
  - [Installing as a standalone service \(Linux, Mac OSX, Solaris, Windows, others\)](#)
  - [Installing the Web Application under a Tomcat server](#)
  - [How to upgrade from a previous version](#)
- [User's Guide](#)
- [Administrator's Guide](#)
- [Developer's Guide to building Continuum](#)

This documentation is also available in [PDF](#) format.

### 1.1.2 Knowledge Base

- [Frequently Asked Questions](#)
- [Wiki](#)

## 2 Getting Started

---

### 2.1 Getting Started

Getting started with Continuum's default configuration is very easy.

After unpacking the binary distribution, run the following command from the `bin` directory:

```
continuum console
```

Continuum will be started from the command line and run until quit with Ctrl-C. For more instructions on installing Continuum as a service or under an existing application server, refer to the [Installation Guides](#).

When you start Continuum for the first time (without an existing database), the first thing you will do is create the administrator account and perform the [General Configuration](#).

#### Create Admin User

Username:	admin
Full Name*:	<input type="text"/>
Email Address*:	<input type="text"/>
Password*:	<input type="password"/>
Confirm Password*:	<input type="password"/>

#### *Admin account creation*

After the administrator account has been created, you can log as the `admin` user with the password you selected. The next thing you will see is the **General Configuration** page.

#### General Configuration

Working Directory*:	<input type="text" value="/path/to/continuum/data/working-directory"/>	Enter the working directory of the Continuum web application
Build Output Directory*:	<input type="text" value="/path/to/continuum/data/build-output-directory"/>	Enter the build output directory of the Continuum web application
Release Output Directory:	<input type="text" value="/path/to/continuum/data/release-output-directory"/>	Enter the release output directory of the Continuum web application
Deployment Repository Directory:	<input type="text"/>	Enter the deployment repository directory of the Continuum web application
Base URL*:	<input type="text" value="http://localhost:8080/continuum"/>	Enter the base URL for the Continuum web application
Number of Allowed Builds in Parallel:	<input type="text" value="1"/>	Enter the number of Allowed Builds in Parallel
<input checked="" type="checkbox"/> Enable Distributed Builds		

#### *General Configuration*

The defaults are usually sufficient, so you can save the configuration. You may later make changes to this page if you are enabling Distributed Builds, for example.

## 2.2 Next Steps

Once the server is running, you will likely want to do the following:

- Create more users via the **Users** menu item, so that you no longer need to use the administrator account for daily tasks.
- [Add Projects](#) to build
- Configure Continuum to work with distributed builds if you intend to have multiple build slave machines
- Configure installed software packages and environment variables for project templates

## 3 Installation/Upgrade Guides

---

### 3.1 Installation/Upgrade Guides

In this section, you'll find information about Continuum installations:

- [Installing as a standalone service \(Linux, Mac OSX, Solaris, Windows, others\)](#)
- [Installing the Web Application under a Tomcat server](#)
- [How to upgrade from a previous version?](#)
- [Installation of Build Agent for Distributed Builds](#)

### 3.2 System Requirements

<b>Java</b>	Java 5 (deploying WAR files) or Java 6 (when using the standalone version with Jetty 8)
<b>Memory</b>	No minimum requirement
<b>Disk</b>	The Continuum application package is less than 30MB but will use more disk space when it's checking out and building sources
<b>Operating System</b>	No minimum requirement. Tested on Windows XP, Debian, Fedora Core, Solaris and Mac OS X



## 4 Installing Standalone

---

### 4.1 Installing Continuum Standalone

#### 4.1.1 Download and Configuration

- Download the standalone version from the [Download page](#)
- Extract the file

*Note:* The standalone version ships with Jetty 8, which requires Java 6.

##### 4.1.1.1 Defining JNDI Resources

#### 4. Mail server configuration

Before you start Continuum, you must configure your SMTP configuration for mail notification. In many servers, the default of delivering to localhost with no authentication will be properly relayed. However, if you need to configure a different SMTP server, the configuration is in `CONTINUUM_HOME/contexts/continuum.xml`:

```
<New id="validation_mail" class="org.eclipse.jetty.plus.jndi.Resource">
  <Arg><Ref id="wac"/></Arg>
  <Arg>mail/Session</Arg>
  <Arg>
    <New class="org.eclipse.jetty.jndi.factories.MailSessionReference">
      <Set name="user"></Set>
      <Set name="password"></Set>
      <Set name="properties">
        <New class="java.util.Properties">
          <Put name="mail.smtp.host">localhost</Put>
        </New>
      </Set>
    </New>
  </Arg>
</New>
```

#### 4. Database configuration

By default, Continuum use an embedded [Derby](#) database. If you want to use a different database, you can modify the JNDI configuration in `CONTINUUM_HOME/contexts/continuum.xml`:

```

<!-- continuum database -->

<New id="continuum" class="org.eclipse.jetty.plus.jndi.Resource">
  <Arg><Ref id="wac"/></Arg>
  <Arg>jdbc/continuum</Arg>
  <Arg>
    <New class="org.apache.commons.dbcp.BasicDataSource">
      <Set name="driverClassName">org.apache.derby.jdbc.EmbeddedDriver</Set>
      <Set name="url">jdbc:derby:<Property name="appserver.base" default="."/>/da
      <Set name="username">SA</Set>
      <Set name="password"></Set>
      <Set name="testOnBorrow">>true</Set>
      <Set name="validationQuery">SELECT count(*) FROM SYS.SYSTABLES</Set>
    </New>
  </Arg>
</New>

<!-- Users / Security Database -->

<New id="users" class="org.eclipse.jetty.plus.jndi.Resource">
  <Arg><Ref id="wac"/></Arg>
  <Arg>jdbc/users</Arg>
  <Arg>
    <New class="org.apache.commons.dbcp.BasicDataSource">
      <Set name="driverClassName">org.apache.derby.jdbc.EmbeddedDriver</Set>
      <Set name="url">jdbc:derby:<Property name="appserver.base" default="."/>/da
      <Set name="username">SA</Set>
      <Set name="password"></Set>
      <Set name="testOnBorrow">>true</Set>
      <Set name="validationQuery">SELECT count(*) FROM SYS.SYSTABLES</Set>
    </New>
  </Arg>
</New>

```

Please refer to the documentation for your JDBC driver for the correct settings to use. You may need to alter the `validationQuery` to one that can successfully execute on your database (or remove the setting if you do not want to validate connections before using them).

You must ensure the JDBC driver is available in the classpath, by copying it to the `$CONTINUUM_HOME/lib` directory and updating `$CONTINUUM_HOME/conf/wrapper.conf` to ensure it is listed instead of the Derby JAR.

Refer to the [Commons DBCP](#) documentation for more advanced configuration of the connection pool.

More information on supported databases is available in the Administrator's Guide on [External Databases](#).

#### 4.1.1.2 Testing the Configuration

Start Continuum from the console in the `$CONTINUUM_HOME/bin/` directory:

```
continuum console
```

If it appears to be working correctly, it is a good idea to set it up as a service so that it will run in the background under a suitable role account, and can be set to start whenever the server has started.

#### 4.1.2 Installing as a Windows Service

- Go to `$CONTINUUM_HOME/bin/` and run the following command:  
`continuum.bat install`
- Edit the **Apache Continuum** service
  - To see the services that are on your computer go to *Start|Run* and enter `services.msc`.
- Select the **Startup Type**
- Go to the **Log On** tab and select a real user. A real user is required because you'll need a home directory for Maven repository and some other things
- Validate your changes

#### 4.1.3 Installing as a Linux Service

Since the Continuum Linux script `bin/continuum` understands the same arguments as Linux boot scripts, there is no need to write a particular startup script to add Continuum to the Linux boot process. All you need to do, as root, is:

```
ln -s /usr/local/continuum-[VERSION]/bin/continuum /etc/init.d/continuum
```

This will allow you to run `service continuum start` and other commands.

To have the service start on boot, follow the instructions below for your particular family of operating system.

##### 4.1.3.1 On a Debian-based system

At this point you have Continuum ready to be symlinked from different runlevels. Debian GNU/Linux comes with a very handy utility to create these links, just run as root:

```
update-rc.d continuum defaults 80
```

If you run this command, you will see something like this:

```
Adding system startup for /etc/init.d/continuum ...
/etc/rc0.d/K80continuum -> ../init.d/continuum
/etc/rc1.d/K80continuum -> ../init.d/continuum
/etc/rc6.d/K80continuum -> ../init.d/continuum
/etc/rc2.d/S80continuum -> ../init.d/continuum
/etc/rc3.d/S80continuum -> ../init.d/continuum
/etc/rc4.d/S80continuum -> ../init.d/continuum
/etc/rc5.d/S80continuum -> ../init.d/continuum
```

What you see is the symlinks that would be created.

##### 4.1.3.2 On a RedHat-based system

Configuring Continuum on a RedHat-based system (like Fedora Core) is slightly different. Instead of running `update-rc.d`, you need to add a new service using the `chkconfig` command.

```
chkconfig --add continuum
chkconfig continuum on
```

#### 4.1.4 Installing as a Mac OS X Service

On OS X, you can use launchd to run a service. Create the following as root in `/Library/LaunchDaemons/org.apache.continuum.plist`:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Label</key>
  <string>org.apache.continuum</string>
  <key>ProgramArguments</key>
  <array>
    <string>/Applications/Continuum/apache-continuum-1.4.3-SNAPSHOT/bin/continuum
    <string>console</string>
  </array>
  <key>Disabled</key>
  <false/>
  <key>RunAtLoad</key>
  <true/>
  <key>UserName</key>
  <string>continuum</string>
  <key>StandardOutPath</key>
  <string>/Applications/Continuum/apache-continuum-1.4.3-SNAPSHOT/logs/launchd.lo
  <!-- Optional - store data separate from installation -->
  <key>EnvironmentVariables</key>
  <dict>
    <key>CONTINUUM_BASE</key>
    <string>/Users/continuum/Library/Continuum</string>
  </dict>
  <!-- Optional: force it to keep running
  <key>KeepAlive</key>
  <true/>
  -->
</dict>
</plist>
```

To install the service, run the following:

```
$ sudo chown root:wheel /Library/LaunchDaemons/org.apache.continuum.plist
$ sudo launchctl load -w /Library/LaunchDaemons/org.apache.continuum.plist
```

Start and stop the service with:

```
$ sudo launchctl start org.apache.continuum.plist
$ sudo launchctl stop org.apache.continuum.plist
```

To uninstall the service:

```
$ sudo launchctl unload -w /Library/LaunchDaemons/org.apache.continuum.plist
```

You can follow the same steps to install build agents as a service by replacing the appropriate paths, and changing `org.apache.continuum` to `org.apache.continuum.buildagent` in the label and `plist` filename.

#### 4.1.5 Installing with Puppet

If you use Puppet to manage your infrastructure, you can use a third-party Puppet module to install Continuum. This will take care of adding the required users, databases and configuration based on official release tarballs.

- [Puppet module for Apache Continuum and Continuum build agents](#)

## 5 Installing in Tomcat

---

### 5.1 Guide to Install Continuum on Tomcat

Instructions for installing, deploying, configuring Continuum for the Apache Tomcat web container.

#### 5.1.1 Server Configuration

With every Tomcat version you will need a few things before you can deploy Continuum.

1. Use at least Java JDK 1.5.0\_11 to run Tomcat with Continuum.
2. A defined `<Context>` xml section to define the JNDI resources.
3. The JavaMail / Activation JAR files.
4. The Apache Derby JAR files.
5. Configure `${appserver.base}` Java system property.

##### 5.1.1.1 Defining JNDI Resources

Continuum will, on startup, ask the web container for a few JNDI configured resources, two JDBC DataSources, and one JavaMail session.

To configure these JNDI resources in the Tomcat Web Container, you will need to specify a `<Context>` section that Tomcat can utilize for those requests coming from Continuum.

Tomcat has several ways to accomplish this, with the most practical being to create a `$(CATALINA_HOME)/conf/Catalina/localhost/continuum.xml` file containing a `<Context>` element.

The following are the JNDI names you will need to provide:

- `mail/Session`
- `jdbc/continuum`
- `jdbc/users`

The individual techniques for describing these resources, and the parameters associated with them are specific to the Tomcat version, resource type, and even JDBC implementation type.

For the purposes of this document, the following assumptions are made.

1. You are an Apache Tomcat administrator.
2. You have an SMTP Server on localhost, port 25, with no login / password.
3. You will be using the embedded Apache Derby database, rather than an external database.
4. Details specific to Apache Tomcat, JavaMail, or Apache Derby are left for the reader to research on those projects websites.

More information on using other supported databases is available in the Administrator's Guide on [External Databases](#).

##### 5.1.1.2 The JavaMail / Activation JAR files

**Note:** Continuum requires JavaMail 1.4 (or later)

Apache Tomcat does not typically ship with a copy of the JavaMail or Activation JAR files. In your role as the Apache Tomcat administrator of your installation, you will need to obtain these JAR files and place it into your preferred `lib` directory.

The appropriate `lib` directory to choose is a personal preference, and we do not encourage or enforce a specific location for it, as all installations of Apache Tomcat are different.

For the record, we personally put them in the `$CATALINA_HOME/common/lib/` directory.

Direct download links for these JAR files.

- JavaMail 1.4 - [mail-1.4.jar](#)
- Java Activation Framework 1.1 - [activation-1.1.jar](#)

#### 5.1.1.3 The Apache Derby JAR files

**Note:**Continuum has been tested with Apache Derby 10.1.3.1

The default installation of Continuum uses the Apache Derby 100% Java database to maintain Continuum-specific information, and also the Users / Security Database.

You will need to obtain `derby.jar` and place it into your preferred `lib` directory.

We put them into the `$CATALINA_HOME/common/lib/` directory.

Direct download links for this JAR file:

- Derby 10.1.3.1 - [derby-10.1.3.1.jar](#)

If you have configured a different database, please copy that JDBC driver in place of the Derby driver.

#### 5.1.1.4 Configure the `appserver.base` Java system property

The `${appserver.base}` system property is used by the Continuum internal logging configuration to determine where to output its logs to. It is important to define this property either in the `$CATALINA_OPTS` system environment variable (if Tomcat is being launched via the command line) or the service properties (if being launched as a service or daemon).

The format typically expected is `-Dappserver.base=/path/to/continuum-base`

You can utilize the `$CATALINA_HOME/bin/setenv.sh` script to set this value in a Tomcat specific way.

```
#!/bin/bash
# Keep the appserver.home and appserver.base values the same when running under Tomcat

export CATALINA_OPTS="-Dappserver.home=$CATALINA_HOME -Dappserver.base=$CATALINA_HOME"
```

#### 5.1.1.5 Context XML Example

The following `context.xml` has been tested on Tomcat v5.5.25 and Tomcat v6.0.14.

```
<Context path="/continuum" docBase="/path/to/continuum-webapp-1.4.3-SNAPSHOT.war">

  <Resource name="jdbc/users"
    auth="Container"
    type="javax.sql.DataSource"
    username="sa"
    password=""
    driverClassName="org.apache.derby.jdbc.EmbeddedDriver"
    url="jdbc:derby:database/users;create=true" />

  <Resource name="jdbc/continuum"
    auth="Container"
    type="javax.sql.DataSource"
    username="sa"
    password=""
    driverClassName="org.apache.derby.jdbc.EmbeddedDriver"
    url="jdbc:derby:database/continuum;create=true" />

  <Resource name="mail/Session"
    auth="Container"
    type="javax.mail.Session"
    mail.smtp.host="localhost" />
</Context>
```



## 6 Upgrade

---

### 6.1 Upgrading Continuum

This document will help you upgrade Continuum from 1.2.x to 1.3.3 and above.

When upgrading Continuum, it could have some database model changes. Usually these changes will be migrated for you, but in some cases you may need to use a backup from the previous version and restore that data into the new version. The Data Management tool exports data from the old database model and imports the data into the new database model.

If you had used the `APP_BASE` environment variable in Continuum 1.2 to differentiate your configuration from the installation, you should rename it to `CONTINUUM_BASE` in Continuum 1.3.

#### 6.1.1 Changes to Take Note of in Continuum 1.4.x

The Jetty version in Continuum 1.4.1 and above has been upgraded to 8.1.7. When upgrading to Continuum 1.4.1 or higher, there is a need to update the contents of `CONTINUUM_BASE/conf/wrapper.conf` to match the one included in the new distribution if any copies or modifications have been made. The `jetty.xml` format has also changed, and should be re-added from the distribution with any modifications previously made. It is no longer read from `CONTINUUM_BASE`, and JNDI connections can now be managed via the files in the `contexts` directory.

#### 6.1.2 Changes to Take Note of in Continuum 1.3.x

The Jetty version in Continuum 1.3.4 and above has been upgraded to 6.1.19. When upgrading to Continuum 1.3.4 or higher, there is a need to update the library contents listed in `CONTINUUM_BASE/conf/wrapper.conf` with the ones included in the new distribution especially if the `CONTINUUM_BASE` directory is separate from the installation.

In Continuum 1.3.6, configuration for local repositories in build agents were added. This is used to match the local repository in the build agent to that of the local repository (used by the project to be built) set in the master. So if you are upgrading from a lower version, you might need to add the `localRepositories` configuration in your build agent's configuration file. For more details, read [Installing and Configuring a Build Agent](#).

#### 6.1.3 Using Backup and Restore to upgrade

There are 2 databases that need to be considered: one for the builds and one for the users.

While the databases typically auto-upgrade across releases, you may wish to perform an export and import to have full control of the upgrade, make modifications to the data, or retain a backup in case a downgrade is needed.

First, download the Data Management tools you will need. The tool is a standalone JAR that you can download from the central repo.

You will need to download two versions of the tool, one for the export out of the old version and one for the import into the new version:

**Note:** The 1.2, 1.2.2 and 1.2.3 released versions of this tool have a bug. To export databases from 1.2.2 or 1.2.3, you will need to use version 1.2.3.1 of the tool. To export databases from 1.2, you may use the 1.1 version of the tool.

- [Data Management CLI for Continuum 1.4.3-SNAPSHOT](#)
- [Data Management CLI for Continuum 1.3.8](#)
- [Data Management CLI for Continuum 1.2.x](#)

- **Data Management CLI for Continuum 1.1**

Next, follow these steps to export data from the old version

- Stop the old version of Continuum
- Execute this command to create the `builds.xml` export file

```
java -Xmx512m -jar data-management-cli-1.2.3.1-app.jar \  
-buildsJdbcUrl jdbc:derby:${old.continuum.home}/data/databases/continuum \  
-mode EXPORT \  
-directory backups
```

Then, follow these steps to import the data to the new version

- Start the new version of Continuum to create the new data model, but do not configure it.
- Stop Continuum
- Execute this command to import the builds data from the XML file you created earlier:

```
java -Xmx512m -jar data-management-cli-1.4.3-SNAPSHOT-app.jar \  
-buildsJdbcUrl jdbc:derby:${new.continuum.home}/data/databases/continuum \  
-mode IMPORT \  
-directory backups -strict
```

**Note:** Remove `-strict` when importing data from 1.3.1 to 1.3.x to ignore unrecognized tags due to model changes.

Finally, be aware that sometimes the `NEXT_VAL` values in the `SEQUENCE_TABLE` need to be adjusted. Before starting Continuum for the first time after the import, connect to the database with a client like [Squirrel SQL](#) and check the values in the `NEXT_VAL` column of the `SEQUENCE_TABLE`.

Values must be greater than the max ID value in each table. For example, the next value of `org.apache.maven.continuum.model.Project` must be greater than the greatest ID in `Project` table.

Here are some example SQL statements. You may need to add or remove lines depending on the contents of your database.

```
UPDATE SEQUENCE_TABLE set NEXT_VAL = (select max(systemconfiguration_id)+1 from SYS
  WHERE SEQUENCE_NAME='org.apache.maven.continuum.model.system.SystemConfiguration'

UPDATE SEQUENCE_TABLE set NEXT_VAL = (select max(id)+1 from BUILDQUEUE)
  WHERE SEQUENCE_NAME='org.apache.maven.continuum.model.project.BuildQueue';

UPDATE SEQUENCE_TABLE set NEXT_VAL = (select max(id)+1 from SCHEDULE)
  WHERE SEQUENCE_NAME='org.apache.maven.continuum.model.project.Schedule';

UPDATE SEQUENCE_TABLE set NEXT_VAL = (select max(id)+1 from BUILDDEFINITION)
  WHERE SEQUENCE_NAME='org.apache.maven.continuum.model.project.BuildDefinition';

UPDATE SEQUENCE_TABLE set NEXT_VAL = (select max(id)+1 from LOCALREPOSITORY)
  WHERE SEQUENCE_NAME='org.apache.maven.continuum.model.repository.LocalRepository';

UPDATE SEQUENCE_TABLE set NEXT_VAL = (select max(id)+1 from PROJECTGROUP)
  WHERE SEQUENCE_NAME='org.apache.maven.continuum.model.project.ProjectGroup';

UPDATE SEQUENCE_TABLE set NEXT_VAL = (select max(scmresult_id)+1 from SCMRESULT)
  WHERE SEQUENCE_NAME='org.apache.maven.continuum.model.scm.ScmResult';

UPDATE SEQUENCE_TABLE set NEXT_VAL = (select max(projectdependency_id)+1 from PROJE
  WHERE SEQUENCE_NAME='org.apache.maven.continuum.model.project.ProjectDependency';

UPDATE SEQUENCE_TABLE set NEXT_VAL = (select max(id)+1 from BUILDDEFINITIONTEMPLATE
  WHERE SEQUENCE_NAME='org.apache.maven.continuum.model.project.BuildDefinitionTemp

UPDATE SEQUENCE_TABLE set NEXT_VAL = (select max(id)+1 from ABSTRACTPURGECONFIGURAT
  WHERE SEQUENCE_NAME='org.apache.maven.continuum.model.repository.AbstractPurgeConfigura
```

Now you can start your new version of Continuum.

## 7 Installing a Build Agent

---

### 7.1 Installing and Configuring a Build Agent

Installing a build agent follows a very similar process to the Continuum master installation.

First, [download](#) the Continuum Build Agent standalone distribution or web application.

**Note:** It is recommended to use Build Agents that come with the release of Continuum (Continuum Master) to be used. While 1.3.x Build Agents can be used with 1.4.x Continuum (Continuum Master), using a later version of Build Agent than the Continuum (Continuum Master) will not work.

#### 7.1.1 Standalone

The Build Agent is available as a separate standalone Jetty-bundled application. Installing it is as simple as extracting it to a preferred directory.

The Build Agent runs on a specific port (defaulting to 8181), so it is possible to install multiple Build Agents in one machine. It is also possible to install Build Agents in the same machine as the Master. To configure the Build Agent to run using a different port, edit the `$CONTINUUM_BUILDAGENT_HOME/conf/jetty.xml` file.

Another configuration file to look at is the `$CONTINUUM_BUILDAGENT_HOME/conf/continuum-buildagent.xml`. This file must be manually edited, as there is no web interface for this configuration.

```
<continuum-buildagent-configuration>
  <buildOutputDirectory>/path/to/build/agent/data/build-output-directory</buildOutputDirectory>
  <workingDirectory>/path/to/build/agent/data/working-directory</workingDirectory>
  <continuumServerUrl>http://builds.example.com:8080/continuum/master-xmlrpc</continuumServerUrl>
  <installations>
    <installation>
      <name>Maven Home</name>
      <type>maven2</type>
      <varValue>/Applications/apache-maven-2.0.9</varValue>
    </installation>
  </installations>
  <localRepositories>
    <localRepository>
      <name>default</name>
      <location>/home/user/.m2/repository</location>
      <layout>default</layout>
    </localRepository>
  </localRepositories>
  <sharedSecretPassword>PSK</sharedSecretPassword>
</continuum-buildagent-configuration>
```

The configuration will have the following fields:

- `continuumServerUrl` - the URL pointing to the XML-RPC context of the Continuum Master
- `sharedSecretPassword` - the pre-shared key (PSK) between master and agent. Copy the PSK that was configured in the Continuum Master configuration file `continuum.xml` (or corresponding web interface field).
- `buildOutputDirectory` - similar to the build output directory of Continuum.
- `workingDirectory` - similar to the working directory of Continuum

- `installations` - an optional list of the installations available on the Build Agent, including Maven, Ant, JDK, etc.
- `localRepositories` - an optional list of the local repositories that will be used by the build agent when building and releasing projects. The local repository name must match that of the local repository configured in the Continuum Master. This configuration is only available in Continuum 1.4.0+.

Note that all of your agents need not be identical, as there is a way to control which agent Continuum chooses by using a [Build Agent Group](#). In addition, you need to [configure](#) the Build Environment from the Master in order to specify which installation to use for the build.

Now start the Build Agent by executing the startup script available in `CONTINUUM_BUILDAGENT_HOME/bin/`. For example:

```
continuum-buildagent start
```

or if you are on Windows

```
continuum-buildagent.bat console
```

Next, [add the agent](#) to the Continuum Master server.

Note that agent will be automatically disabled if the PSK is not configured properly or if the agent is unreachable by the Continuum Master.

To conclude the installation, you should set it up as a service, following the same instructions as those shown for Continuum for [Windows](#) or [Linux](#).

### 7.1.2 Web Application

The Build Agent is also available as a web application which you can install into your own container. Unlike Continuum itself, you do not need to configure Mail and Databases, so the deployment is much simpler.

The Build Agent needs the following in order to function properly:

1. a `appserver.base` system property
2. a `continuum-buildagent.xml` configuration file in `${appserver.base}/conf`
3. a working area with `logs` and `temp` directories

Set the system property using the recommended method for your application server. An example for Tomcat is shown in the [same instructions as those shown for Continuum](#).

Next, create the `continuum-buildagent.xml` file as shown in the Standalone section above, and save it in `${appserver.base}/conf`. You will need to create these directories, as well as `${appserver.base}/logs` and `${appserver.base}/temp`.

Finally, drop the web application into your container, and start it.

### 7.1.3 Installing with Puppet

As with Continuum, if you use Puppet to manage your infrastructure, and wish to install the standalone build agent, you can use a third-party Puppet module.

- [Puppet module for Apache Continuum and Continuum build agents](#)

## 8 User's Guides

---

### 8.1 User's Guides

- [Building Projects](#)
- [Managing Build Definitions](#)
- [Managing Projects](#)
- [Notifications](#)
- [Releasing Projects](#)
- [Project Builds Report](#)

## 9 Managing Projects

---

### 9.1 Managing Projects

- [How to add a project](#)
- [How to edit a project](#)
- [How to remove a project](#)
- [Viewing the project's working copy](#)

## 10 Add a Project

---

### 10.1 Add a Project

#### 10.1.1 Maven project (Maven 2 or Maven 3)

From the menu, choose the **Maven Project** entry

Maven Project

*Add a Maven project*

The `pom.xml` file must be available through one of the following protocols: `http`, `https`, or `ftp`.

**Note:** The file protocol is also supported but is disabled by default for security and must be enabled manually.

The `pom.xml` can also be uploaded from a local file.

**Note:** This doesn't support multi module projects.

☐ Add Maven 2.0+ Project

POM Url*:	<input type="text"/>
Username:	<input type="text"/>
Password:	<input type="password"/>
	<input type="checkbox"/> Use SCM Credentials Cache, if available
	Enter the URL to the Maven 2 POM. Provide the username and password if it is a secured resource.
<b>OR</b>	
Upload POM:	<input type="text"/> <input type="button" value="Browse..."/>
	Enter the local filename of the Maven 2 POM to upload (works only for a single project without modules).
Project Group:	Defined by POM <input type="button" value="v"/>
	<input checked="" type="radio"/> Checkout multi-module project in separate directories <input type="radio"/> Checkout multi-module project in single directory <input type="radio"/> For multi-module project, load only root as recursive build
Build Definition Template:	Default <input type="button" value="v"/>
<input type="button" value="Add"/> <input type="button" value="Cancel"/>	

*Add a Maven project*

You can supply a username and password if the POM URL requires authentication. This information will be stored in the database in plain text and reused later for SCM checkouts and updates.

If your SCM stores credentials like CVS or SVN and you want to use the SCM Credentials cache, check the **Use SCM Credentials Cache** field.

You can define the group you want to use or **Defined by POM** in this case `project.name` will be used as **Project Group**.

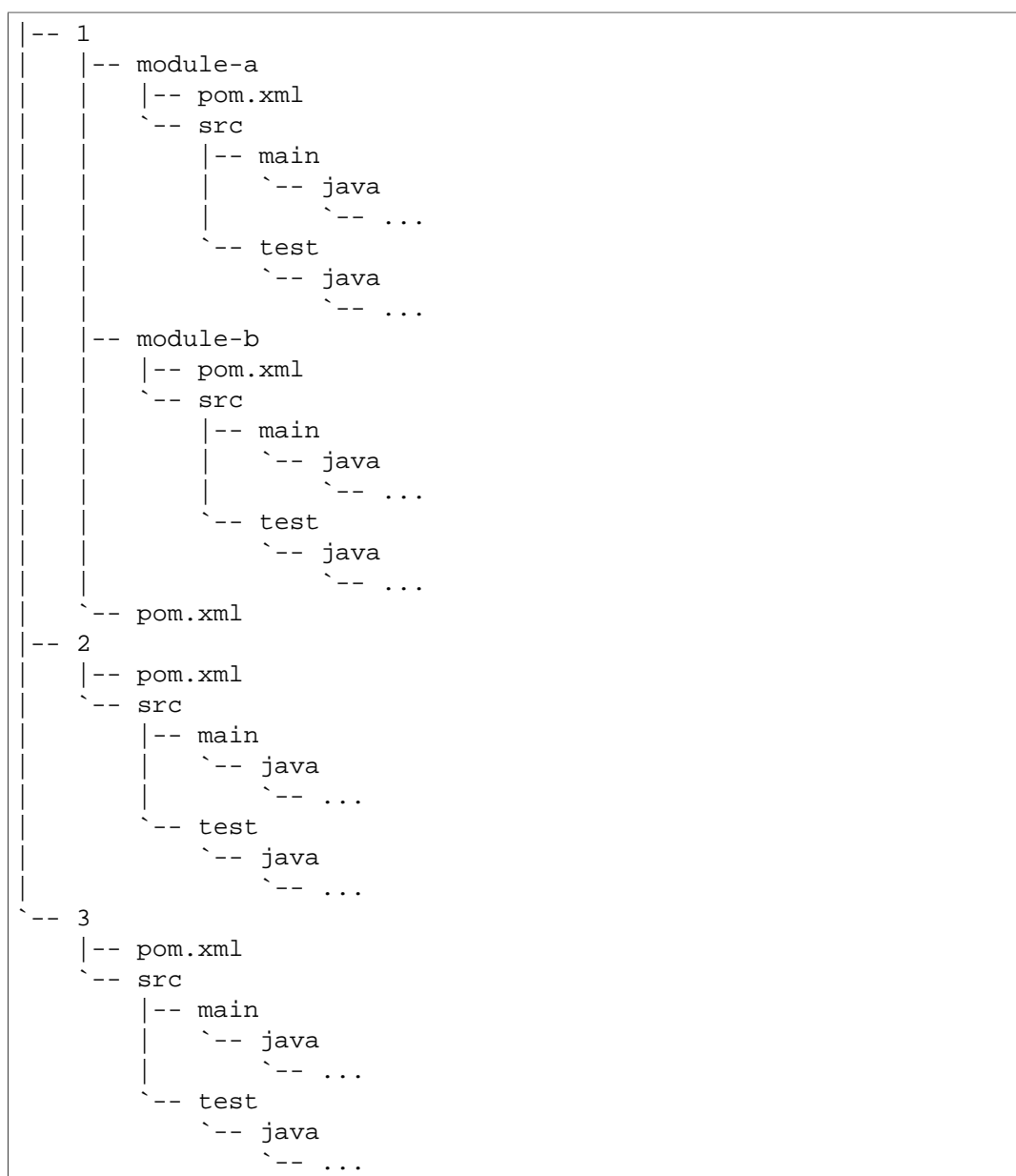
You cannot add a duplicate project (duplicates would mean another instance of a project with the same `groupId`, `artifactId`, and `version`) in the same Project Group.

If **Checkout multi-module project in separate directories** radio box is selected when a project is added to Continuum, each sub-module POM would be added as an individual Continuum Project. Each of these Continuum projects having it's own working directory. For example, you have a multi-module project with the following structure:



```
simple-multi-module-project
|-- module-a
|  |-- pom.xml
|  `-- src
|     |-- main
|     |  |-- java
|     |  `-- ...
|     `-- test
|         |-- java
|         `-- ...
|-- module-b
|  |-- pom.xml
|  `-- src
|     |-- main
|     |  |-- java
|     |  `-- ...
|     `-- test
|         |-- java
|         `-- ...
`-- pom.xml
```

This project tree means that there will be three Continuum Projects added in Continuum. Let's say, the top level project was assigned a `project id` of '1' when it was added in Continuum and module-a got a `project id` of '2' and module-b got a `project id` of '3'. Now in Continuum's working directory, you will see these projects checked out as below. Note that their `project ids` correspond to their own working directories.



You can force Continuum to check out the entire project into a single directory by selecting the **Checkout multi-module project in single directory** radio box.

**Note:** If your project has a flat structure similar to the one below, you might want to check out your project into a single directory to be able to release it in Continuum.

```

simple-flat-multi-module
|-- root-project
|   |-- pom.xml
|   |-- module-a
|       |-- pom.xml
|       |-- src
|           |-- main
|               |-- java
|               |-- ...
|           |-- test
|               |-- java
|               |-- ...
|-- module-b
    |-- pom.xml
    |-- src
        |-- main
            |-- ...

```

If you want to add the multi-module project as a single Continuum project, select the **load only root as recursive build** radio box.

For a project or POM that was added as a single Continuum project and as its own project group, the `--non-recursive` flag in the default project group build definition will be removed. But if the project or POM was added as a single Continuum project to an already existing project group, a project level build definition with the `--non-recursive` flag stripped from the `arguments` field will be created for the project and will be set as the project's default build definition. See [CONTINUUM-2563](#) for the related issue.

You can choose a **Build Definition Template** which will be applied to your project. If you choose a **Build Definition Template**, the build definitions from the template will be added in addition to any build definitions that are defined at the project group level.

When adding projects whose parent project is building on another server, make sure that the `settings.xml` with the necessary repositories exists in `[USER_HOME]/.m2/` directory. Continuum uses `[USER_HOME]/.m2/settings.xml` and not the one found in `[M2_HOME]/conf/`.

The scm connection url must be present in the POM and must match the actual location of the pom in the scm repository.

### 10.1.2 Maven 1 project

From the menu, choose the **Maven 1.x Project** entry

Maven 1.x Project

*Add a Maven 1 project*

The `project.xml` file must be available through one of the following format : http, https and ftp.

**Note:** The file protocol is off by default for security and must be enabled manually.

The `project.xml` can also be uploaded from a local file.

**Note:** This doesn't support multi modules project.

The project you're adding can't use a parent.

**Add Maven 1.x Project**

<b>M1 POM Url:</b>	<input type="text"/>
Username:	<input type="text"/>
Password:	<input type="text"/>
	<input type="checkbox"/> Use SCM Credentials Cache, if available
Enter the URL to the Maven 1 POM. Provide the username and password if it is a secured resource.	
<b>OR</b>	
<b>Upload POM:</b>	<input type="text"/> <input type="button" value="Browse..."/>
Enter the local filename of the Maven 1 POM to upload	
<b>Project Group:</b>	Defined by POM <input type="button" value="v"/>
<b>Build Definition Template:</b>	Default <input type="button" value="v"/>
<input type="button" value="Add"/> <input type="button" value="Cancel"/>	

### *Add a Maven 1 project*

You can define username/password if the POM URL require an authentication.

If your SCM store credentials like CVS or SVN and you want to use the SCM Credentials cache, check the **Use SCM Credentials Cache** field.

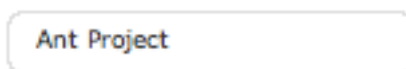
You can define the group you want to use or **Defined by POM**. In this case `project.name` will be used as the **Project Group**.

You cannot add a duplicate project (duplicates would mean another instance of a project with the same groupId, artifactId, and version) in the same **Project Group**.

You can choose a **Build Definition Template** which will be applied to your project.

#### 10.1.3 ANT Project

From the menu, choose the **Ant Project** entry



### *Add an Ant project*

TO WRITE

### ➤ Add Ant Project

<b>Project Name*:</b>	<input type="text"/>
	Enter the project name
<b>Version*:</b>	<input type="text"/>
	Enter the version of the project
<b>Scm Uri*:</b>	<input type="text"/>
	Enter the <u>Maven SCM URL</u>
<b>Scm Username:</b>	<input type="text"/>
	Enter the scm username
<b>Scm Password:</b>	<input type="text"/>
	Enter the scm password
<b>Scm Branch/Tag:</b>	<input type="text"/>
	Enter the scm branch/tag name ( For subversion, tag name must be in scm URL and not in this field )
	<input type="checkbox"/> Use SCM Credentials Cache, if available
<b>Project Group Name:</b>	Default Project Group ▾
<b>Build Definition Template:</b>	Default ▾
<input type="button" value="Add"/> <input type="button" value="Cancel"/>	

*Add an Ant project*

#### 10.1.4 Shell Project

From the menu, choose the **Shell Project** entry

Shell Project

*Add a shell project*

TO WRITE

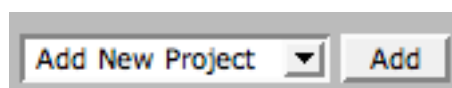
**Continuum - Add Shell Project**

<b>Project Name*:</b>	<input type="text"/>
	Enter the project name
<b>Version*:</b>	<input type="text"/>
	Enter the version of the project
<b>Scm Url*:</b>	<input type="text"/>
	Enter the <a href="#">Maven SCM URL</a>
<b>Scm Username:</b>	<input type="text"/>
	Enter the scm username
<b>Scm Password:</b>	<input type="text"/>
	Enter the scm password
<b>Scm Branch/Tag:</b>	<input type="text"/>
	Enter the scm branch/tag name ( For subversion, tag name must be in scm URL and not in this field )
	<input type="checkbox"/> Use SCM Credentials Cache, if available
<b>Project Group Name:</b>	Default Project Group ▾
<b>Build Definition Template:</b>	Default ▾
<input type="button" value="Add"/> <input type="button" value="Cancel"/>	

*Add a shell script project*

### 10.1.5 Add a project from the Project Group

From the **Project Group**, you can add a project without using the menu. With this operation, the **Project Group** will be set to the current group.



*Add a project from a project group*

### 10.1.6 SCM hints

#### 10.1.6.1 ClearCase

With ClearCase, you can configure SCM things in few ways. For example, you can use, in the scm URL in your POM, the absolute path of your config spec file like this:

```
<scm>
  <connection>scm:clearcase:absolute_path_of_config_spec</connection>
</scm>
```

The SCM URL format used for ClearCase is defined [here](#)

and you can create a clearcase-settings.xml file under \$ user.home/.scm/ with the following content:

```
<clearcase-settings>
  <viewstore>\\mymachine\myvwstore</viewstore>
  <useVWSParameter>true</useVWSParameter>
</clearcase-settings>
```

This configuration won't work with each ClearCase installation because each ClearCase configuration is different, so we recommend reading the [ClearCase page](#) on the Maven SCM site.

# 11 Edit a Project

## 11.1 Edit a Project

### 11.1.1 Project View

When you edit a project from the **Group Summary** page, you will see basic information. The first part of the page contains information regarding:

- project name
- project version
- project SCM URL
- project Last Build Date
- all build definitions attached to the project

Project Group Summary || **Project Information** || Builds || Working Copy

[-] **Continuum Project**

Project Name:	Continuum IRC Notifier
Version:	1.1-beta-4-SNAPSHOT
SCM Url:	scm:svn:http://svn.apache.org/repos/asf/maven/continuum/trunk/continuum-notifiers/continuum-notifier-irc
SCM Branch/Tag:	
Group:	Continuum Notifiers
Last Build Date:	oct. 16, 2007 10:04:34 PM CEST

[-] **Build Definitions**

Goals	Arguments	Build File	Schedule	Profile	From	Build Fresh	Default	Description	Type			
clean	--batch-mode	pom.xml	<u>DEFAULT_SCHEDULE</u>		GROUP	false	true	default maven2 buildDefinition	maven2			
install	--non-recursive											

*Project View Top*

With the **Edit** button, you can change some project information like the SCM URL if your project was moved. With the **Build Now** button, you build the project manually with the default build definition.

The second part contains information regarding:

- project notifiers
- project dependencies
- project developers



**Notifiers**

Type	Recipient	Events	From
mail	notifications@maven.apache.org	Success/Failures/Warnings/Errors	Project

**Dependencies**

Group Id	Artifact Id	Version
junit	junit	3.8.1
jmock	jmock	1.0.1
org.codehaus.plexus	plexus-component-api	1.0-alpha-20
org.codehaus.plexus	plexus-container-default	1.0-alpha-20
org.apache.maven.continuum	continuum-api	1.1-beta-4-SNAPSHOT
org.codehaus.plexus	plexus-notification	1.0-alpha-5
org.schwering	irc4lib	1.10
org.apache.maven.continuum	continuum-notifier-api	1.1-beta-4-SNAPSHOT
org.apache.maven.plugins	maven-remote-resources-plugin	1.0-alpha-5
org.codehaus.plexus	plexus-maven-plugin	1.3.3

**Developers**

Name	Email
Jason van Zyl	jason@maven.org
Arnaud Heritier	aheritier@apache.org
Brett Porter	brett@apache.org
Brian Fox	brianf@apache.org
Carlos Sanchez	carlos@apache.org

### Project View Bottom

#### 11.1.2 Build Results

You can view all project's **Build Results** with the link **Builds**. You will see all build results summary: start/end time, result. It's possible to delete some build results with checkboxes.

Project Group Summary | Project Information | **Builds** | Working Copy

**Build results for Continuum IRC Notifier**

<input type="checkbox"/>	Build #	Start Time	End Time	Duration	State	Build Definition Description	Result
<input type="checkbox"/>	4	mar., 16 oct. 2007 22:04:27 +0200	mar., 16 oct. 2007 22:04:34 +0200	Duration : 6 sec		default maven2 buildDefinition	<a href="#">Result</a>
<input type="checkbox"/>	3	mar., 16 oct. 2007 22:02:58 +0200	mar., 16 oct. 2007 22:03:05 +0200	Duration : 7 sec		default maven2 buildDefinition	<a href="#">Result</a>

### Build Results

Now you can edit a build result

#### 11.1.3 Build Result

Editing a build result will display:

- SCM changes (author and file path)
- project dependencies (project recorded in this Continuum instance)
- detail of the build definition used
- the full build command line output

**SCM Changes**  
No SCM changes

**Dependencies Changes**  
No dependencies changes

**Build Definition Used**

POM filename	pom.xml
Goals	clean install
Arguments	--batch-mode --non-recursive
Build Fresh	false
Always Build	false
Is it default ?	true
Schedule	DEFAULT_SCHEDULE
Description	default maven2 buildDefinition

**Output**

[Download as Text](#)

```
[INFO] Scanning for projects...
[INFO] -----
[INFO] Building Continuum IRC Notifier
[INFO]   task-segment: [clean, install]
[INFO] -----
[INFO] [clean:clean]
[INFO] Deleting directory C:\tmp\continuum\working-directory\3\target
[INFO] Deleting directory C:\tmp\continuum\working-directory\3\target\classes
[INFO] Deleting directory C:\tmp\continuum\working-directory\3\target\test-classes
[INFO] Deleting directory C:\tmp\continuum\working-directory\3\target\site
[INFO] [plexus:descriptor {execution: generate}]
[INFO] Setting property: classpath.resource.loader.class => 'org.codehaus.plexus.velocity.ContextClassLoaderResourceLoader'.
```

### *Build Result*

## 12 Remove a Project

---

### 12.1 Remove a Project

To remove a project, go to the **Project Group Summary** and click the **delete** icon.



*Delete icon*

You can also use the **Delete group** button in the **Project Group Summary** to remove the whole project group.

## 13 Working Copy







---

### 13.1 Project Working Copy

The **Working Copy** tab displays the working copy used for the project.

The source files can be viewed by clicking the filename and can be downloaded as text using the **Download as Text** link.

**Note:** The file is automatically loaded as an attachment for download when it exceeds 100KB in size.

Project Group Summary	Project Information	Builds	<b>Working Copy</b>
			
 /			
 pom.xml		1 Kb	Jan 27, 2010 08:00:31 PM PHT
 src			
 report-public.html		374 Kb	Jan 27, 2010 04:25:23 PM PHT
 test.html		97 Kb	Jan 27, 2010 02:09:47 PM PHT

[Download as Text](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
  transitional.dtd">
<html>
```

*Working Copy*

## 14 Managing Build Definitions

### 14.1 Managing Build Definitions

Each project or project group must have at least one attached build definition in order to build.

Depending on the project, you can define different values which will be used in order to build your project.

Continuum includes some default build definitions which can be changed in the [Build Definition Templates](#) screen.

You can add or edit build definitions at the **Project Group level** or at the **Project level**.

### 14.2 Project Group Build Definition

The project group level build definition is executed on all of the projects in the group. The following description covers the fields for a Maven project, though the fields for other types are similar.

Project Group Build Definitions of Maven Quick Start Archetype group													
Goals	Arguments	Build File	Schedule	Build Environment	From	Build Fresh	Default	Description	Type	Always Build			
clean install	--batch-mode --non-recursive	pom.xml	DEFAULT_SCHEDULE		GROUP	false	true	Default Maven 2 Build Definition	maven2	false			
<input type="button" value="Add"/>													

*Project Group Build Definitions*

To add a build definition to your project group, under the **Build Definitions** tab in the **Project Information** screen, click **Add**.

Supply the values for the necessary fields:

- Enter your **POM filename**, which is very likely the same as previous builds, pom.xml.
- In the **Goals** text box, enter your new goals. For example, clean site deploy or release:branch.
- You may have arguments for Maven, such as --batch-mode --non-recursive. Add your arguments in the **Arguments** text box or leave it empty.
- Check **Build Fresh** to do a fresh checkout from SCM before the build process. Check **Always Build** if you want to execute the goal every time you build the project. If you want this build definition to be your default for building your project, check the **Is it default?** check box.
- Select your build schedule from the pull-down menu. There will only be the DEFAULT\_SCHEDULE if you have not defined any others.
- Select from the pull-down menu the **Build Environment** for the build definition.
- Select the **Type** from the pull-down menu.
- Select the **Update Policy** from the pull-down menu. This describes when a Continuum project definition should be updated if changes are detected in the Maven POM.
- You can supply a short description of the build definition ( **Description**).

**Add/Edit Build Definition**

**POM filename\*:**

**Goals:**

**Arguments:**

Build Fresh (Run always a clean checkout instead of an SCM update)

Always Build

Is it default?

**Schedule:**

**Build Environment:**

**Type:**

**Update Policy:**

**Description:**

Click **Save** to complete the changes.

### 14.3 Project Build Definition

The project level build definition is executed for a specific project where the build definition is defined. The following description covers the fields for a Maven project, though the fields for other types are similar.

Build Definitions												
Goals	Arguments	Build File	Schedule	Build Environment	From	Build Fresh	Default	Description	Type			
clean install	--batch-mode --non-recursive	pom.xml	DEFAULT_SCHEDULE		GROUP	false	true	Default Maven 2 Build Definition	maven2			

#### *Project Build Definitions*

To add a build definition to your project, select a project within the group and under the **Build Definitions** section in the **Project Information** screen, click **Add**.

The fields are the same as for a project group build definition.

## 15 Managing Notification

### 15.1 Managing Notification

Notifiers can be attached to a Maven project by adding them to the POM in the [ciManagement section](#).

```
<ciManagement>
  <system>continuum</system>
  <url>http://127.0.0.1:8080/continuum</url>
  <notifiers>
    <notifier>
      <type>mail</type>
      <sendOnError>true</sendOnError>
      <sendOnFailure>true</sendOnFailure>
      <sendOnSuccess>false</sendOnSuccess>
      <sendOnWarning>false</sendOnWarning>
      <configuration>
        <address>continuum@127.0.0.1</address>
      </configuration>
    </notifier>
  </notifiers>
</ciManagement>
```

Or they can be added to a **Project Group** with the **Notifiers** link:

Project Group Summary	Members	Build Definitions	Notifiers
<b>Project Group Informations</b>			
<b>Project Group Name:</b>	Continuum Parent Project		
<b>Project Group Id:</b>	org.apache.maven.continuum		
<b>Description:</b>	Maven is a software project management and comprehension tool. can manage a project's build, reporting and documentation from a		
<b>Group Actions</b>			

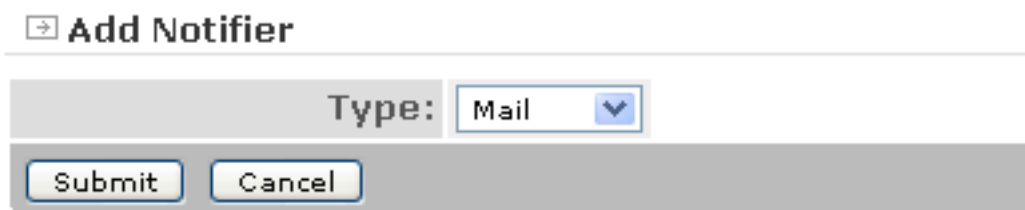
*Notifiers Link*

Or added to a project with the **Add** button on the **Edit Project** page:

Notifiers		
Type	Recipient	Events
mail	notifications@maven.apache.org	Success/Failures/Warnings/Errors
<input type="button" value="Add"/>		

*Notifier Button*

Here you select between the Notifier implementations provided with Continuum:



➔ **Add Notifier**

Type: Mail ▼

Submit Cancel

*Notifier Type Choice*

You have the following choices:

- [Mail](#)
- [IRC](#)
- [Jabber](#)
- [MSN](#)
- [Wagon](#)



## 16 Mail Notification

---

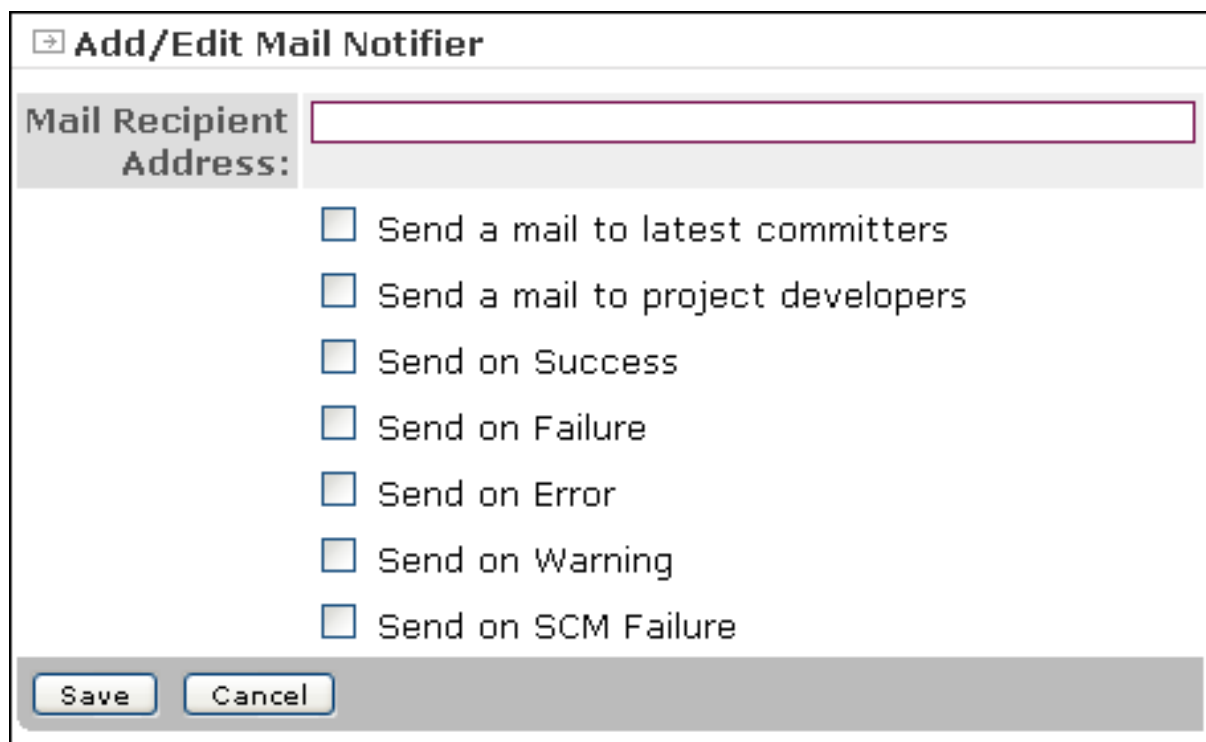
### 16.1 Mail Notification

If you are using Maven 2.0+, email notifiers configured in `pom.xml` will be automatically added, and you may have as many as you need.

```
<ciManagement>
  <system>continuum</system>
  <notifiers>
    <notifier>
      <configuration>
        <address>email address</address>
        <committers>true</committers>
        <developers>true</developers>
      </configuration>
    </notifier>
  </notifiers>
</ciManagement>
```

Or you can manually add mail notifiers at the project group or individual project level.

On the **Add Notifier** form select **Mail** from the pull-down menu and click **Submit**.



**Add/Edit Mail Notifier**

Mail Recipient Address:

- Send a mail to latest committers
- Send a mail to project developers
- Send on Success
- Send on Failure
- Send on Error
- Send on Warning
- Send on SCM Failure

*Add Email Notifier*

You must either provide a single email address, or select to notify the latest committers or project developers.

In addition you may select one or more **Send on...** options.  
Then, click **Save**.

## 17 IRC Notification

---

### 17.1 IRC Notification

1. Under **Notifiers** from the **Project Information** page, click **Add**.
2. On the **Add Notifier** form select **IRC** from the pull-down menu and click **Submit**.
3. On the next page, fill out the form with the following information:
  - IRC Host
  - IRC Port
  - IRC Channel
  - Nick Name
  - Alternate Nick Name
  - User Name
  - Full Name
  - Password which can be left blank if you did not specify one in your IRC account

Add/Edit IRC Notifier	
IRC Host*:	<input type="text"/>
IRC port:	<input type="text"/>
IRC channel*:	<input type="text"/>
Nick Name (default value is continuum):	<input type="text"/>
Alternate Nick Name (default value is continuum_):	<input type="text"/>
User Name (default value is the nick name):	<input type="text"/>
Full Name (default value is the nick name):	<input type="text" value="admin"/>
Password:	<input type="password" value="*****"/>
	<input type="checkbox"/> SSL
	<input type="checkbox"/> Send on Success
	<input type="checkbox"/> Send on Failure
	<input type="checkbox"/> Send on Error
	<input type="checkbox"/> Send on Warning
	<input type="checkbox"/> Send On SCM Failure
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

*Add IRC Notifier*

4. Select the **Send on ...** options you want.
5. Click **Save**. The IRC notifier you just added will be listed under **Notifiers** on the **Project Information** page along with the other notifiers you may have set up prior to this.

## 18 Jabber Notification

---

### 18.1 Jabber Notification

1. Under **Notifiers** from the **Project Information** page, click **Add**.
2. Select **Jabber** from the pull-down menu and click Submit.
3. On the **Add/Edit Jabber Notifier** page, fill out the form with the following:

- Jabber Host
- Jabber Port is typically 5222
- Jabber Login
- Jabber Password
- Jabber Domain Name is the fully qualified domain name of your Jabber login
- Jabber Recipient Address

Add/Edit Jabber Notifier	
Jabber Host*:	<input type="text"/>
Jabber port:	<input type="text"/>
Jabber login*:	<input type="text" value="admin"/>
Jabber Password*:	<input type="password" value="....."/>
Jabber Domain Name:	<input type="text"/>
Jabber Recipient Address*:	<input type="text"/>
<input type="checkbox"/>	Is it a SSL connection?
<input type="checkbox"/>	Is it a Jabber group?
<input type="checkbox"/>	Send on Success
<input type="checkbox"/>	Send on Failure
<input type="checkbox"/>	Send on Error
<input type="checkbox"/>	Send on Warning
<input type="checkbox"/>	Send On SCM Failure
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

*Add Jabber Notifier*

The Jabber login and Jabber recipient address must be different. The Jabber recipient must also be added in the contact list of the sender (at least for Google talk).

4. Select the **Send on ...** options you want.

5. Click **Save**. The Jabber notifier you just added will be listed under **Notifiers** on the **Project Information** page along with the other notifiers you may have set up prior to this.

## 19 MSN Notification

---

### 19.1 MSN Notification

1. Under **Notifiers** from the **Project Information** page, click **Add**.
2. On the **Add Notifier** form select **MSN** from the pull-down menu and click **Submit**.
3. On the next page, fill out the form with the following:
  - MSN Login
  - MSN Password
  - MSN Recipient Address

**Add/Edit MSN Notifier**

MSN login \*: admin

MSN Password \*: .....

MSN Recipient Address \*:

Send on Success

Send on Failure

Send on Error

Send on Warning

Send On SCM Failure

Save Cancel

*Add MSN Notifier*

4. Select the **Send on ...** options you want.
5. Click **Save**. The MSN notifier you just added will be listed under **Notifiers** on the **Project Information** page along with the other notifiers you may have set up prior to this.

## 20 Wagon Notification

---

### 20.1 Wagon Notification

1. Under **Notifiers** from the **Project Information** page, click **Add**.
2. On the **Add Notifier** form select **Wagon** from the pull-down menu and click **Submit**.
3. On the next page, enter the Project Site URL.

It must be a Wagon URL like the URL used in `distributionManagement`. For the Webdav protocol, you can use `dav:[http_url]`, or `file:///path_to_the_directory` if you want to copy it locally with the file protocol.

4. Enter the `ServerId`. This should match the value of the `<id>` in the server configuration found in `settings.xml`. Sample configuration follows:

```
<server>
  <username>admin</username>
  <password>admin123</password>
  <id>continuum.site</id>
</server>
```

Using the server configuration in `[user_home]/.m2/settings.xml` is a workaround for user authentication using the Wagon notifier.

The `<username>` and `<password>` should be set to the user account that has the **Site Producer** and **Site Observer** roles.

**Add/Edit Wagon Notifier**

Project Site URL\*:

Server Id (defined in your settings.xml for authentication)\*:

Send on Success

Send on Failure

Send on Error

Send on Warning

*Add Wagon Notifier*

5. Select the **Send on ...** options you want.
6. Click **Save**. The Wagon notifier you just added will be listed under **Notifiers** on the **Project Information** page along with the other notifiers you may have set up prior to this.



The file protocol will create a `buildresult.txt` file locally in the location specified in the Wagon Notifier configuration. If you use the Webdav protocol, a `buildresult.txt` file will be copied to your disk, relative to where you told it to be placed when you set up the Wagon notifier.

You can also view the `buildresult.txt` file from a browser at the Project Site URL you specified.

## 21 Building a project

---

### 21.1 Building Projects

For Ant and Maven builds the following system properties will be added via -D command line:

- `continuum.project.group.name`
- `continuum.project.lastBuild.state` : last build state int flag
- `continuum.project.lastBuild.number`
- `continuum.project.nextBuild.number`
- `continuum.project.id`
- `continuum.project.name`
- `continuum.project.version`

When building projects, the build order is determined by the inter-dependency of the projects. This would mean that the dependencies of a project are built first.

In cases when a project group can build projects in parallel build queues but one of those projects depend on another project, the inter-dependent projects will build in the same build queue.

Projects can be built in two ways:

1. [Scheduled Builds](#)
2. [Forced Builds](#)

Projects will have the **in queue** icon once the forced build or a scheduled build is triggered. This means that the project is either in the **prepare build queue** or in the **build queue**.



Checkout or update of projects takes longer with distributed builds enabled because after an SCM checkout or SCM update, an SCM changelog follows to get the "latest update date" needed to determine whether projects should build or not.

#### 21.1.1 Cancelling builds

If a build has been started, you can cancel it before it completes. See [Cancelling Builds](#) for more information.

## 22 Scheduled Build

---

### 22.1 Scheduling Project Build

Schedules determine the project's time interval to build.


When scheduled builds are triggered, there is no dependency ordering between project groups and the following happens:

- Projects are updated from the SCM. These are queued under the **Prepare Build Queues** section in **Queues** page.
- Builds of the projects are executed. These are queued under the **Build Queues** in **Queues** page.

To configure a scheduled build:

1. Create a schedule. You can refer to [Managing Schedules](#) section.
2. Add a build definition to a project using the schedule created in step 1 for the **Schedule** field. You can refer to the section [Managing Build Definitions](#).

➔ **Add/Edit Build Definition**

<b>POM filename*:</b>	<input type="text" value="pom.xml"/>
<b>Goals:</b>	<input type="text"/>
<b>Arguments:</b>	<input type="text"/>
	<input type="checkbox"/> Build Fresh (Run always a clean checkout instead of an SCM update)
	<input type="checkbox"/> Always Build
	<input type="checkbox"/> Is it default?
<b>Schedule:</b>	DEFAULT_SCHEDULE ↕ 
<b>Build Environment:</b>	↕
<b>Type:</b>	maven2 ↕
<b>Update Policy:</b>	Always ↕
<b>Description:</b>	<input type="text"/>

#### *Scheduling build*

**Note:** In a distributed build setup, the project builds are not executed if no build agent is configured or if a build agent group with no build agent, is configured in the build environment that is attached to the project's build definition.

To view the results of the build, refer to [Build Results Management](#).

## 23 Forced Build

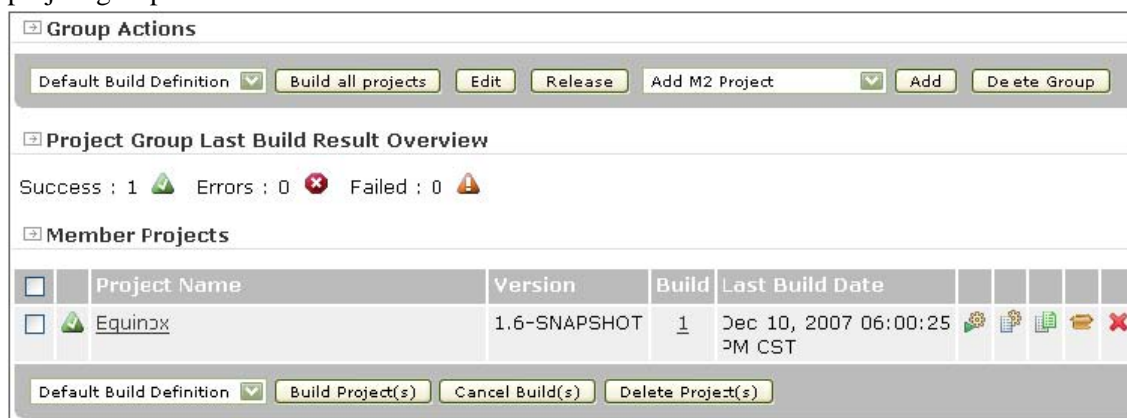
### 23.1 Force Project Build

Occasionally you may want to force a build outside of the regular schedule. When builds are triggered manually, the following happens:

- Projects are updated from the SCM. These are queued under the **Prepare Build Queues** section in **Queues** page.
- Builds of the projects are executed. These are queued under the **Build Queues** in **Queues** page.

To manually build a project perform the following steps:

1. Click **Show Project Groups**.
2. Click the project group link of the project you want to build. This will put you on the **Summary** page of the project group.
3. You can click **Build all projects** button under **Group Actions** to build all the projects in the project group.



#### *Project Group Actions*

4. To build an individual project, you can click on the build icon that is found along the row of the project you want to build. It is the first icon on the right of the project name. This will execute the default build goal you have defined for that project.

To build selected projects, check the box beside the name of the project, then click the **Build Project(s)** button.

If you want to build the project according to a specific build goal other than the default build goal, click on the project name to go to the **Project's Information** page. All build goal definitions for the project are listed under the **Build Definitions** section. Click on the build icon to the right of the specific build goal you want to execute.

- Note\* In a distributed build setup, the project builds are not executed if no build agent is configured or if a build agent group with no build agent, is configured in the build environment that is attached to the project's build definition.

To view the results of the build, refer to [Build Results Management](#).

## 24 Build Results Management

### 24.1 Build Results Management

Project Group Summary		Project Information		<b>Builds</b>	Working Copy	
[-] Build results for Equinox						
	Build #	Start Time	End Time		State	
<input type="checkbox"/>	4	Thu, 16 Aug 2007 11:39:59 +0800	Thu, 16 Aug 2007 11:40:03 +0800	Duration : 3 sec		<a href="#">Result</a>
<input checked="" type="checkbox"/>	3	Thu, 16 Aug 2007 11:39:42 +0800	Thu, 16 Aug 2007 11:39:46 +0800	Duration : 4 sec		<a href="#">Result</a>
<input checked="" type="checkbox"/>	2	Thu, 16 Aug 2007 11:39:07 +0800	Thu, 16 Aug 2007 11:39:11 +0800	Duration : 4 sec		<a href="#">Result</a>
<input type="checkbox"/>	1	Thu, 16 Aug 2007 11:38:56 +0800	Thu, 16 Aug 2007 11:39:04 +0800	Duration : 8 sec		<a href="#">Result</a>
Delete						

*Builds tab*

To view the result of the build:

1. Click the link for the project group of the project you want to view - the link will be under the **Name** column.
2. Click the build number link under the **Build** column of the project you want to view. Or you can click the project link again and click the **Builds** link at the top of the page, then click the **Result** link, next to the **State** column.

When builds are executed frequently, deleting unnecessary and failed build results is needed. To delete build results, follow these steps:

1. Click the name of the project group.
2. Click on the build number under the **Build** column. You will be directed to the build result page as shown below.

<b>Start Time:</b>	Jun 21, 2009 04:44:21 PM PHT
<b>End Time:</b>	Jun 21, 2009 04:44:30 PM PHT
<b>Duration:</b>	8 sec
<b>Build Trigger:</b>	Forced
<b>State:</b>	
<b>Build#:</b>	2
<b>Triggered by:</b>	admin
Delete	

*Build Results page*

The **Triggered by** field displays the username of the person who forced the build, or the name of the schedule if the build is triggered by a schedule.

Or you can click on the name of the project that you want to delete build results from.

- Click the **Builds** tab. The build results list will be displayed.
- Check the box of the build results that you want to delete.

3. Click the **Delete** button.

## 25 Cancelling builds

---

### 25.1 Cancelling Project Build

This will cancel the build of the selected project.

To cancel the build for project, click the **Cancel** button of the project on the **Project Group Summary** page or on the **Queues** page.



*Cancel Button*

### 25.2 Cancelling Project Group Build

This will cancel all builds of the project group and will also remove the projects from the build queue.

In the Project Group Summary page, click the

Group Actions



*Cancel Group Build Button*

## 26 Release Management

---

### 26.1 Releasing Projects

The release process has two steps, [Prepare](#) and [Perform](#).

1. First, it prepares your project for release by doing the following:
  - Checks that your project is in a "releasable" state.
  - Updates the POMs for the new development version.
  - Runs a test build.
  - Generates a tag for the release.
  - Generates the artifact to be deployed, such as a JAR file.
  - Generates the site and deploys it.
  - Commits the new POMs with the new development version. It will do the commit to the SCM connection URL you specify (explained later).
2. Then the release management system performs the release by checking out a clean copy of the tagged release and deploying the artifacts and site.

#### 26.1.1 Modifying POM for Release

The source code control system you use must be specified in your POM file in an `<scm>` entry similar to the one shown here:

```
<scm>
  <connection>
    scm:svn:http://svn.apache.org/repos/asf/maven/components/trunk
  </connection>
  <developerConnection>
    scm:svn:https://svn.apache.org/repos/asf/maven/components/trunk
  </developerConnection>
  <url>http://svn.apache.org/viewcvs.cgi/maven/components/trunk</url>
</scm>
```

To determine what to enter for your connection URL (the `<connection>` tag) go to the link <http://maven.apache.org/scm/scms-overview.html> for a list of supported SCMs and click on your source code control system to see the appropriate syntax for your particular software.

In general the format for an SCM URL is:

```
scm:<scm_provider><delimiter><provider_specific_part>
```

In the example above, the `scm_provider` is `svn` (Subversion). The `delimiter` is a colon. And the rest of the line is the `provider_specific_part`.

It is important to prepare the release just before performing the release. The reason these processes have been separated into two steps is to allow the user to confirm that everything has been set correctly before the actual release. If between the prepare and performing of the release there is a change to the SCM tree, an error will occur causing the prepare process to restart.

#### 26.1.2 Releasing Projects with Flat Structure

In order to release projects with flat structure in Continuum, the project must be checked out in a single directory. You can do this by checking the "Checkout multi-module project in single directory" field when you add your Maven Two project in Continuum. See [Add a Project](#) for more details.



Projects with flat structures are released in the same process described above.

### **26.1.3 Releasing projects in a distributed build**

Release will happen in the Build Agent where the last build of project occurred.

To view on-going releases from different build agents, click the **Releases** button under the **Distributed Builds** menu.

## 27 Prepare Project Release

### 27.1 Prepare for Release

Navigate to the **Project Group Summary** and click the **Release** button or icon.

*Release*

Choose **Prepare project for release** and click **Submit**.

*Release*

Complete the form, providing the SCM tag name, etc.

Check the **Use edit mode** checkbox if you are using an SCM system that requires you to obtain a lock before editing.

Click **Submit**.

When releasing in a [distributed builds](#) setup, a **Build environment** should be selected containing at least one enabled build agent. If none is selected, then Continuum will attempt to use a default agent, starting with the one last used to build the project.

Furthermore, the release will not happen in the following cases:

- no build agent found in the build agent group attached to the build environment
- all build agents configured in the selected build environment are disabled

#### Prepare Project for Release

Release Prepare Parameters

SCM Username:	admin
SCM Password:	●●●●●●●●
SCM Tag*:	hello-5.5
SCM Tag Base:	http://example.com/repos/hello/tags
SCM Comment Prefix:	
Preparation Goals*:	clean
Arguments:	
Build Environment:	JAVA_HOME ▾
	<input type="checkbox"/> Use edit mode
	<input checked="" type="checkbox"/> Add a schema to the POM if it was previously missing on release

Maven Quick Start Archetype

Release Version*	5.5
Next Development Version*	5.6-SNAPSHOT

#### *Release Prepare parameters*

Wait for the process to complete, then click **Done**.

**Note:** Hitting the **Rollback** button does not roll back or delete the SCM tag created during the `scm-tag` phase. So when re-doing the release prepare of the same version, make sure that the tag does not exist in SVN, if so, the tag should be manually deleted.

**Executing Release Goal**

update-working-copy  
 generate-reactor-projects  
 check-poms  
 scm-check-modifications

**Executing Release Goal** ency-snapshots

	Status	
create-backup-poms		
map-release-versions		update-working-copy
input-variables		generate-reactor-projects
map-development-versions		check-poms
rewrite-poms-for-release		scm-check-modifications
generate-release-poms		check-dependency-snapshots
run-preparation-goals		create-backup-poms
scm-commit-release		map-release-versions
scm-tag		input-variables
rewrite-poms-for-development		map-development-versions
remove-release-poms		rewrite-poms-for-release
scm-commit-development		generate-release-poms
end-release		run-preparation-goals
		scm-commit-release
		scm-tag
		rewrite-poms-for-development
		remove-release-poms
		scm-commit-development
		end-release

[View Output](#)

You will be returned to this page to make another selection.

**Choose Release Goal for hello**

---

Prepare project for release

Perform project release

1.0.19

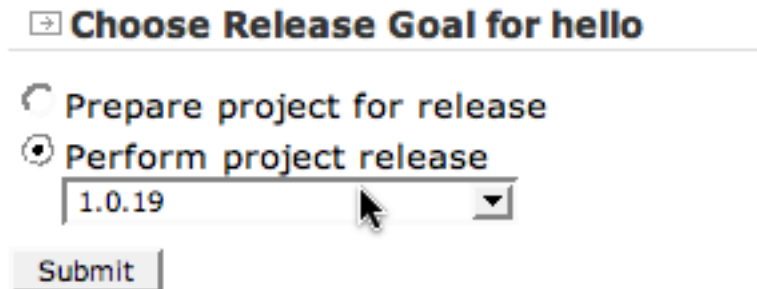
To view the release prepare results, refer to [Release Results Management](#).

## 28 Perform Project Release

---

### 28.1 Perform Release

Choose **Perform Release** and the version number



**Choose Release Goal for hello**

Prepare project for release

Perform project release

1.0.19

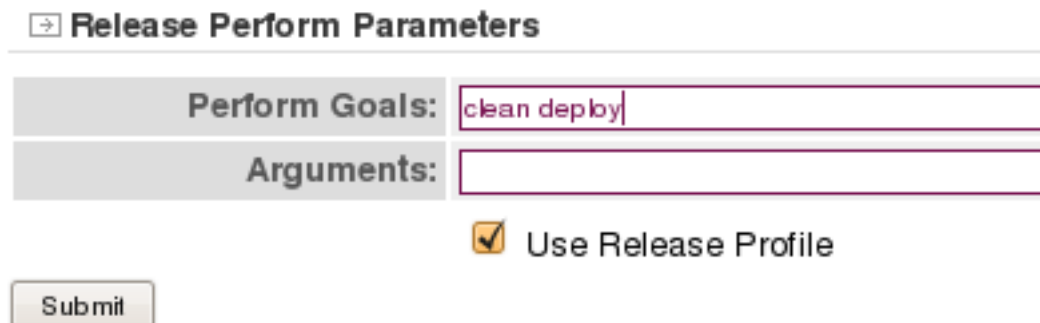
Submit

Fill in the form and click **Submit**

To specify a different Maven goal to execute during the release, fill in the **Perform Goals** field with another goal. By default, the goal for this field is `clean deploy`.

To specify additional arguments during execution, fill in the **Arguments** field.

### Perform Project Release



**Release Perform Parameters**

<b>Perform Goals:</b>	clean deploy
<b>Arguments:</b>	

Use Release Profile

Submit

Wait for the process to complete, then click **Done**.

The screenshot shows a dialog box titled "Executing Release Goal" with a table of tasks. The table has a header row with "Status" and a list of task names. The tasks listed are "verify-completed-prepare-phases", "checkout-project-from-scm", and "run-perform-goals". The status icons for these tasks are green triangles with checkmarks. Below the table, there is a "View Output" link, a "Rollback Changes" button, and a "Done" button. A "Refresh" button is also visible in the top-left corner of the dialog.

Status	
	verify-completed-prepare-phases
	checkout-project-from-scm
	run-perform-goals

To view the release perform results, refer to [Release Results Management](#).

## 29 Release Results Management

### 29.1 Release Results Management

The **Release Results** tab from the **Project Information** page displays all the release results of a project when executing the release goals prepare and perform.

Project Group Summary						
Members		Build Definitions		Notifiers		<b>Release Results</b>
Release Results of Maven Quick Start Archetype group						
<input type="checkbox"/>	Project Name	Release Goal	Start Time	End Time	State	
<input type="checkbox"/>	Maven Quick Start Archetype	prepare	Sep 24, 2008 02:49:44 PM CST	Sep 24, 2008 02:50:55 PM CST	SUCCESSFUL	<a href="#">View Result</a>
<input type="checkbox"/>	Maven Quick Start Archetype	perform	Sep 24, 2008 02:51:17 PM CST	Sep 24, 2008 02:54:37 PM CST	SUCCESSFUL	<a href="#">View Result</a>
<input type="button" value="Delete"/>						

*Release Results page*

To view the release result, click the **View Result** link in-line with the release goal being executed. The **Triggered by** field displays the username of the person who initiated the release.

#### Project Release Summary

Release Project Details	
<b>Project Name:</b>	Maven Quick Start Archetype
<b>Release Goal:</b>	prepare
<b>Start Time:</b>	Jun 23, 2009 11:19:46 PM PHT
<b>End Time:</b>	Jun 23, 2009 11:20:04 PM PHT
<b>State:</b>	SUCCESSFUL
<b>Triggered by:</b>	admin

#### Release Execution Output

```
[INFO] Updating local copy against the scm...
[INFO] Verifying that there are no local modifications...
[INFO] Checking dependencies and plugins for snapshots ...
[INFO] Transforming 'Maven Quick Start Archetype'...
[INFO] Not generating release POMs
```

*Release Summary*

To delete release results, select the boxes beside the project name then click **Delete**.

## 30 Project Builds Report

---

### 30.1 Project Builds Report

Project Builds Report page displays project builds and its information provided with search criteria.

[➔](#) **Project Builds Report**

<b>Project Group:</b>	ALL ▼
<b>Start Date:</b>	06/01/2010
<b>End Date:</b>	06/03/2010
<b>Build Status:</b>	ALL ▼
<b>Triggered By:</b>	
<b>Row Count:</b>	10

*Builds Report Page*

Searching using the default values of the fields will display all the project builds.

Search Criteria:

- **Project Group** - selected from the list of existing project groups. Default value is ALL.
- **Date** of the build (the start and/or end dates of the build)
- **Build Status** - selected from the list which can be ALL, Ok, Failed, or Error. Default value is ALL.
- **Triggered By** - user or build schedule which triggered the build.
- **Row Count** - sets the display of the number of build results to be displayed in a result page. Default value is 30 and the minimum required row count is 10.

To generate the report, click the **View Report** button.

To restore the default values of the fields, click the **Reset** button.

#### 30.1.1 Project Builds Report Sample

To export the builds report to a CSV file, click the link **Export to CSV**



**Project Builds Report**

Project Group:	Continuum Simple Example Project
Start Date:	06/04/2010
End Date:	06/04/2010
Build Status:	ALL
Triggered By:	
Row Count:	10

**Results**

Prev 1 [Next](#) [Export to CSV](#)

Project Group	Project Name	Build Date	Triggered By	Build Status
Continuum Simple Example Project	Continuum Simple Example Project	Jun 04, 2010 10:20:30 AM PHT	admin	
Continuum Simple Example Project	Continuum Simple Example Project	Jun 04, 2010 10:22:47 AM PHT	admin	
Continuum Simple Example Project	Continuum Simple Example Project	Jun 04, 2010 10:29:05 AM PHT	DEFAULT_SCHEDULE	
Continuum Simple Example Project	Continuum Simple Example Project	Jun 04, 2010 10:30:03 AM PHT	DEFAULT_SCHEDULE	
Continuum Simple Example Project	Continuum Simple Example Project	Jun 04, 2010 10:30:51 AM PHT	admin	
Continuum Simple Example Project	Continuum Simple Example Project	Jun 04, 2010 10:31:03 AM PHT	DEFAULT_SCHEDULE	
Continuum Simple Example Project	Continuum Simple Example Project	Jun 04, 2010 10:32:03 AM PHT	DEFAULT_SCHEDULE	
Continuum Simple Example Project	Continuum Simple Example Project	Jun 04, 2010 10:33:04 AM PHT	DEFAULT_SCHEDULE	
Continuum Simple Example Project	Continuum Simple Example Project	Jun 04, 2010 10:34:03 AM PHT	DEFAULT_SCHEDULE	
Continuum Simple Example Project	Continuum Simple Example Project	Jun 04, 2010 10:35:03 AM PHT	DEFAULT_SCHEDULE	

*Project Builds Report*

## 31 Administrator's Guides

---

### 31.1 Administrator's Guides

- [Managing Users and Security](#)
- [Managing Project Groups](#)
- [Managing Installations](#)
- [Managing Build Environments](#)
- [Managing Build Agents](#)
- [Managing Build Agent Groups](#)
- [Managing Schedules](#)
- [Managing General Configuration](#)
- [Managing Local Repositories](#)
- [Managing Purge Configuration](#)
- [Managing Parallel Builds](#)
- [Managing Build Queues](#)
- [Managing Project Queues](#)
- [External Databases](#)
- [Monitoring Continuum](#)
- [Log Files](#)
- [Appearance Configuration](#)
- [Build Definition Templates](#)
- [Shutting Down Continuum](#)
- [Understanding Distributed Builds](#)
- [Accessing remote working copies using WebDAV](#)

## 32 Managing Users and Security

---

### 32.1 Managing Users and Security

[Customising Security Configuration](#)

[LDAP Configuration](#)

## 33 Security Configuration

---

### 33.1 Continuum Security Configuration

Security properties and password rules can be configured in the `security.properties` file, which by default is searched for in:

- `~/.m2/security.properties`
- `$CONTINUUM_BASE/conf/security.properties`
- `$CONTINUUM_HOME/conf/security.properties`

(In the list above, `~` is the home directory of the user who is running Continuum, and `$CONTINUUM_HOME` is the directory where Continuum is installed, such as `/opt/continuum-1.2.`)

Following are some of the properties you can modify. For a complete list, consult the default properties file in Redback's svn repo: [config-defaults.properties](#)

```
# Security Policies
#security.policy.password.encoder=
security.policy.password.previous.count=6
security.policy.password.expiration.days=90
security.policy.allowed.login.attempt=3

# Password Rules
security.policy.password.rule.alphanumeric.enabled=false
security.policy.password.rule.alphacount.enabled=true
security.policy.password.rule.alphacount.minimum=1
security.policy.password.rule.characterlength.enabled=true
security.policy.password.rule.characterlength.minimum=1
security.policy.password.rule.characterlength.maximum=8
security.policy.password.rule.musthave.enabled=true
security.policy.password.rule.numericalcount.enabled=true
security.policy.password.rule.numericalcount.minimum=1
security.policy.password.rule.reuse.enabled=true
security.policy.password.rule.nowhitespace.enabled=true
```

**Note:** If installed standalone, Continuum's list of configuration files is *itself* configurable, and can be found in: `$CONTINUUM_HOME/apps/continuum/webapp/WEB-INF/classes/META-INF/plexus/application.xml`

#### 33.1.1 Additional CSRF Prevention

To help prevent cross-site request forgery, it is possible to enable a basic check that the referrer is the current site.

*Note:* This is only a generic solution that may prevent some types of attacks but not others. It may cause problems with certain user agents. By default, the check is off.

To enable the check, change the following configuration value in the `struts.xml` file in the `WEB-INF/classes` directory of the web application (2 locations):

```
<interceptor-ref name="redbackSecureActions">
  <param name="enableReferrerCheck">false</param>
</interceptor-ref>
```

## 34 LDAP Configuration

---

### 34.1 LDAP Configuration

Continuum supports LDAP for authentication. To configure it, you should follow these steps:

- Shutdown Continuum
- Add a `security.properties` files under `$CONTINUUM_HOME/conf/` or `$CONTINUUM_BASE/conf` with the following content:  
`security.policy.password.expiration.enabled=false`

```
user.manager.impl=ldap
ldap.bind.authenticator.enabled=true

ldap.config.context.factory=com.sun.jndi.ldap.LdapCtxFactory
ldap.config.hostname=[ldap_hostname]
ldap.config.base.dn=[ldap_base_dn]
ldap.config.port=[ldap_port]
ldap.config.mapper.attribute.user.id=cn
ldap.config.mapper.attribute.user.email=email
ldap.config.mapper.attribute.fullname=givenName
ldap.config.mapper.attribute.password=userPassword

redback.default.admin=[adminuser]
# Optional
# redback.default.guest=[guestuser]
```

**adminuser** is a LDAP user and will be the default Continuum admin. **guestuser** is a LDAP user and will be used for the `guest` role, generally, it is an utility LDAP account. If not specified, then a built-in guest account will be used.

- Restart Continuum

#### 34.1.1 Other resources

[Redback LDAP page](#)

## 35 Managing Project Groups

### 35.1 Managing Project Groups

#### Project Groups

Name	Group Id					Total
<a href="#">Default Project Group</a>	default				4 0 0	4
Summary					4 0 0	4

[Add Project Group](#)

*Group Summary homepage*

#### 35.1.1 Adding a Project Group

From the home page (Group Summary) Use the the Add Project Group button

[Add Project Group](#)

*Add Project Group*

Then You will have the project Group detail.

#### Add Project Group

Project Group Name*:	<input type="text"/>
Project Group Id*:	<input type="text"/>
Description:	<input type="text"/>
<a href="#">Save</a>	<a href="#">Cancel</a>

*Edit Project Group detail*

Some fields are mandatory :

- Project Group Name
- Project Group Id

#### 35.1.2 Editing a Project Group

From the home page (Group Summary), click the project group name link.

The project group information will then be displayed.

**Project Group Summary**

Members

Build Definitions

Notifiers

Release Results

[-] Project Group Information "Default Project Group"

<b>Project Group Name:</b>	Default Project Group
<b>Project Group Id:</b>	default
<b>Description:</b>	Contains all projects that do not have a group of their own
<b>Local Repository:</b>	DEFAULT
<b>Homepage Uri:</b>	<a href="http://maven.apache.org">http://maven.apache.org</a>

[-] Project Group Scm Root

<b>Scm Root URL</b>
scm:svn:http://[redacted]/repos/multi-module/trunk

[-] Group Actions

Default Build Definition ▾
Build all projects
Edit
Release
Add M2 Project ▾
Add
Delete Group
Cancel Group Build

[-] Project Group Last Build Result Overview

Success : 4 Errors : 0 Failed : 0

[-] Member Projects

<input type="checkbox"/>	Project Name	Version	Build	Last Build Date					
<input type="checkbox"/>	<a href="#">Module A</a>	24-SNAPSHOT	<u>1</u>	Oct 06, 2009 03:46:17 PM PHT					
<input type="checkbox"/>	<a href="#">Module B</a>	24-SNAPSHOT	<u>1</u>	Oct 06, 2009 03:46:30 PM PHT					
<input type="checkbox"/>	<a href="#">Module C</a>	24-SNAPSHOT	<u>1</u>	Oct 06, 2009 03:46:45 PM PHT					
<input type="checkbox"/>	<a href="#">Multimodule Project</a>	24-SNAPSHOT	<u>1</u>	Oct 06, 2009 03:46:08 PM PHT					

Default Build Definition ▾
Build Project(s)
Cancel Build(s)
Delete Project(s)

*Project Group Summary*

Then, click on the `Edit` button.

**Update Project Group**

<b>Project Group Name*:</b>	Default Project Group
<b>Project Group Id:</b>	default
<b>Description:</b>	Contains all projects that do not have a group of their own
<b>Local Repository:</b>	DEFAULT ▾
<b>Homepage Url:</b>	http://maven.apache.org

**Projects**

Project Name	Move to Group
<b>Multimodule Project:</b>	Default Project Group ▾
<b>Module A:</b>	Default Project Group ▾
<b>Module B:</b>	Default Project Group ▾
<b>Module C:</b>	Default Project Group ▾

Save Cancel

### Update Project Group

The following fields can be updated:

- **Project Group Name:** The name of the project group.
- **Description:** A brief description about the group.
- **Local Repository:** The specific repository used for the project which points to the exact location of the repository in the machine used for building.
- **Homepage Url:** The site of the project.

And, projects can be moved to another project group.

#### 35.1.3 Deleting a Project Group

There are two ways to delete a project group.

From the home page (Group Summary), click the Delete icon to the right of the group you wish to delete.

From the Project Group Summary page, click the Delete Group button.

Confirm the deletion

Note that user roles for this project group will not be removed. This is intentional so that if you are sharing a user database among several Continuum instances, users will still be able to access the same group on a different server.



## 36 Managing Installations

---

### 36.1 Managing Installations

Continuum allows you to configure the build tools it uses to build projects by defining [Build Environments](#). Installations allow defining environment settings and are the basic unit of a Build Environment.

For example, say you need to maintain two build environments. One using JDK 5 and Maven 2.0.9 and another using JDK 6 and Maven 2.1.0-M1. To achieve this, you would need to do the following:

1. Create Installations for the versions of Java and Maven.
2. Create Installations for any environment variables you want to set such as MAVEN\_OPTS and ANT\_OPTS.
3. Create the desired Build Environments, specifying the combination of Installations to use.
4. Assign the Build Environments to your Build Definition Templates or project-specific Build Definitions.

#### 36.1.1 Installations

Installations are essentially named environment variables. There are currently two types of Installations: Tools and Environment Variables.

##### 36.1.1.1 Tools

Tools are meant to provide locations to pre-defined tool types. Currently, continuum allows you to define locations for the JDK, Maven 2, Maven 1 and Ant. When you create a tool installation, you specify a human-friendly name, the tool type, and the path to tool. Continuum verifies that the specified tool type resides in that location or fails to create the Installation.

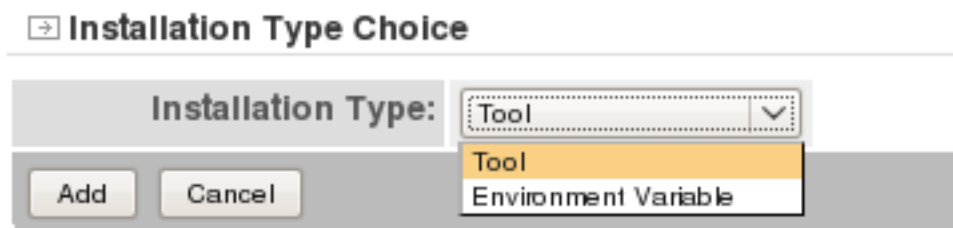
36.Example: Defining a Maven2 Installation

From the menu, choose the 'Installations' entry



*Installations*

Here you must choose the Installation Type you want to add (here a Tool)



*Installation Type Choice*

You must configure the tool you want to add

**Continuum - Installation**

Name\*:

Type:

Value/Path\*:

Build Environment with the Installation name

Save Cancel

### Tool Setup

The value 'Value/Path' field must specify the path to the tool:

- For maven2 : it must be similar to your M2\_HOME
- For maven1 : it must be similar to your MAVEN\_HOME
- For ant : it must be similar to your ANT\_HOME

Continuum validates the path specified depending on the type of Tool:

- For maven2 : <path>/bin/mvn -v will be tested
- For maven1 : <path>/bin/maven -v will be tested
- For ant : <path>/bin/ant -v will be tested

If the path you specify fails the test, the following error will be displayed:

**Continuum - Installation**

Name\*:

Type:

Failed to validate installation, check server log.

Value/Path\*:

Create a Build Environment with the Installation name

Save Cancel

### Installation validation failed

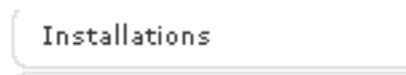
You can use the checkbox if you want to create a Build Environment with the same name as your Tool name.

#### 36.1.1.2 Environment Variables

Environment Variables are simply a means to store environment settings using a human-friendly name. Once defined, they can be used to affect the build tools used to build your projects. They consist of a human-friendly name, the name of the environment variable to set and the value. Unlike Tools, Environment Variables are not constrained by continuum. No validation is performed to validate the values.

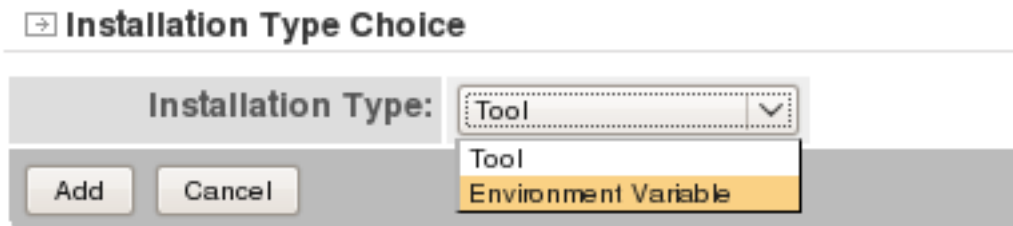
#### 36.Example: Defining a Java Heap size Environment Variable for Maven2 Tools

From the menu, choose the 'Installations' entry



*Installations*

Select the Environment Variable Installation type



*Select env var as tool type*

And finally, specify the human-friendly name, the environment variable name and the corresponding value. Here we define MAVEN\_OPTS that will set the Java heap size to 256 Megabytes.



*Define the env var name and value*





## 37 Managing Schedules

---

### 37.1 Managing Schedules

Schedules initiates the building of projects continuously. In this section, adding, modifying, and deleting of schedules are discussed.

Clicking the **Schedules** link from the menu on the left will display the following page:

Schedules						
Name	Description	Quiet Period	Cron Expression	Max Job Time	Active	
DEFAULT_SCHEDULE	Run hourly	0	0 0 * * * ?	3600	true	 
MONTHLY_SCHEDULE	Run monthly	0	0 0 9 1 * ?	0	true	 

*List of schedules page*

#### 37.1.1 Adding Schedules

To add a schedule, follow these steps:

1. Click the **Add** button below the list of schedules.
2. Fill in the needed information in the **Edit Schedules** page shown below.

Continuum - Edit Schedule

<b>Name*:</b>	DEFAULT_SCHEDULE															
	Enter the name of the schedule															
<b>Description*:</b>	Run hourly															
	Enter a description of the schedule															
<b>Cron Expression:</b>	<table border="1"> <tr> <td><b>Second:</b></td> <td>0</td> </tr> <tr> <td><b>Minute:</b></td> <td>0</td> </tr> <tr> <td><b>Hour:</b></td> <td>*</td> </tr> <tr> <td><b>Day of Month:</b></td> <td>*</td> </tr> <tr> <td><b>Month:</b></td> <td>*</td> </tr> <tr> <td><b>Day of Week:</b></td> <td>?</td> </tr> <tr> <td><b>Year [optional]:</b></td> <td></td> </tr> </table> <p>Enter the cron expression. Format is described there : <a href="#">Syntax</a></p>	<b>Second:</b>	0	<b>Minute:</b>	0	<b>Hour:</b>	*	<b>Day of Month:</b>	*	<b>Month:</b>	*	<b>Day of Week:</b>	?	<b>Year [optional]:</b>		
<b>Second:</b>	0															
<b>Minute:</b>	0															
<b>Hour:</b>	*															
<b>Day of Month:</b>	*															
<b>Month:</b>	*															
<b>Day of Week:</b>	?															
<b>Year [optional]:</b>																
<b>Maximum job execution time (seconds)*:</b>	3600															
	Enter the maximum number of seconds a job may execute in this schedule before it's terminated.															
<b>Quiet Period (seconds):</b>	0															
	Enter a quiet period period for this schedule															
<b>Add Build Queue:</b>	<table border="1"> <tr> <td>BUILD_QUEUE_2</td> <td>&lt;-</td> <td>DEFAULT_BUILD_QUEUE</td> </tr> <tr> <td>BUILD_QUEUE_3</td> <td>-&gt;</td> <td></td> </tr> <tr> <td>BUILD_QUEUE_4</td> <td>&lt;&lt;--</td> <td></td> </tr> <tr> <td></td> <td>--&gt;&gt;</td> <td></td> </tr> <tr> <td></td> <td>&lt;*&gt;</td> <td></td> </tr> </table>	BUILD_QUEUE_2	<-	DEFAULT_BUILD_QUEUE	BUILD_QUEUE_3	->		BUILD_QUEUE_4	<<--			-->>			<*>	
BUILD_QUEUE_2	<-	DEFAULT_BUILD_QUEUE														
BUILD_QUEUE_3	->															
BUILD_QUEUE_4	<<--															
	-->>															
	<*>															
	<input checked="" type="checkbox"/> Enabled Enable/Disable the schedule															
<input type="button" value="Save"/> <input type="button" value="Cancel"/>																

### Adding a build schedule

- **Name** The unique identifier of the schedule. This is a required field.
- **Description** Brief description of the configured schedule. This is a required field.
- **Cron Expression** Determines the exact time interval that the build schedule executes.
- **Maximum job execution time (seconds)** This sets the maximum time for a job to be executed. This is a required field.
- **Quiet Period (seconds)** This a setting that delays the build if there has been a commit in the defined number of seconds prior. This is useful when using CVS, since commits are not atomic and a developer might be committing midway through an update. It is not typically needed if using Subversion.
- **Add Build Queue** Move build queues from the left hand list box to the right hand list box by clicking the appropriate buttons in between the two list boxes. All build queues on the right hand list box will be attached to the schedule.

- **Enabled** When selected, scheduled build will be executed. []
3. Click **Save**.

### 37.1.2 Editing Schedules

To edit a schedule, follow these steps:

1. Click the edit icon which can be found to the right of the schedule.
2. Modify the fields in the **Edit Schedules** page.
3. Click **Save**.

### 37.1.3 Deleting Schedules

To delete a schedule, click the delete icon, the rightmost icon inline with the schedule to be deleted.

## 38 Managing General Configuration

### 38.1 Managing General Configuration

The first time you start Continuum, you will see the 'General Configuration' page. This page is preconfigured with default values.

General Configuration

Working Directory*:	<input type="text" value="/path/to/continuum/data/working-directory"/>	Enter the working directory of the Continuum web application
Build Output Directory*:	<input type="text" value="/path/to/continuum/data/build-output-directory"/>	Enter the build output directory of the Continuum web application
Release Output Directory:	<input type="text" value="/path/to/continuum/data/release-output-directory"/>	Enter the release output directory of the Continuum web application
Deployment Repository Directory:	<input type="text"/>	Enter the deployment repository directory of the Continuum web application
Base URL*:	<input type="text" value="http://localhost:8080/continuum"/>	Enter the base URL for the Continuum web application
Number of Allowed Builds in Parallel:	<input type="text" value="1"/>	Enter the number of Allowed Builds in Parallel

Enable Distributed Builds

#### General Configuration

Working Directory	The directory where all projects will be checked out
Build Output Directory	The directory where all build output will be stored
Release Output Directory	The directory where all release output will be stored
Deployment Repository Directory	The directory where generated maven2 artifacts will be stored. This directory will be respect a repository structure. It is independant of the maven deploy phase
Base URL	The base Continuum URL. This URL is used in notifications
Number of Allowed Builds in Parallel	The number of build queues that will be utilized to build projects. This allows concurrent checkouts and builds of multiple projects. Available since version 1.3.1.
Enable Distributed Builds	When checked, this Continuum instance will delegate all builds to configured agents.
Shared Secret Password	The pre-shared key (PSK) between the master and all configured agents. The PSK will need to be copied to all configured agents.

You can modify values or keep defaults then save.

Although Continuum defaults to storing its data within the installation, it is a best practice to separate the installation and data directories.

These values can be pre-configured in `continuum.xml`. Make sure that Continuum is not running when editing this file.

You can modify values at a later time with the following steps:

1. Login to Continuum with an Administrator account
2. Click the **Configuration** link under the **Administration** section



*General Configuration Menu*

3. Click the **Edit** button




## 39 Managing Local Repositories

### 39.1 Local Repository

Local repositories can be configured for a specific project group's use. This is where the artifacts used for building the projects can be found instead of the default (`$USER_HOME/.m2/repository`).

Click the **Local Repositories** link under the **Administration** section



*Local Repositories*

You will see all the local repositories that have been created.

To display:

Local Repositories		
Name	Location	Layout
repository1	/home/marica/repository1	default

Add

In this screen you can add/edit/delete/purge a local repository.

You can purge a repository if it has a default [Purge Configuration](#), otherwise the purge icon is disabled.



*disabled purge icon*

#### 39.1.1 Adding / Editing a Local Repository

All fields are mandatory.

- Name: must be a unique repository name.
- Location: must be the absolute path of a unique repository location.
- Layout: "default" or "legacy". Default layout is for maven2, while legacy layout is for maven1.

Continuum - Add/Edit Local Repository

Name*:	<input type="text"/>
Location*:	<input type="text"/>
Layout:	default
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

Adding a local repository will automatically create a default [Purge Configuration](#) for that repository.

If you're using distributed builds, you must configure a `localRepository` with the same name in the build agent's configuration file where the `location` is the local path to the local repository in the build agent machine. See [Distributed Builds](#) for more details.

### 39.1.2 Default

The default local repository location comes from the settings of the user running Continuum. The location cannot be edited within Continuum, but can be changed by specifying it in `~/ .m2/ settings.xml`.

On Windows 7, you can pass the argument `-Duser.home=C:\Users\<<Username>\` to the JVM explicitly. Since running Continuum as an Administrator will give you a different `user.home` value.

## 40 Managing Purge Configuration

### 40.1 Purge Configuration

Click the **Purge Configurations** link under the **Administration** section.

Purge Configurations

*Purge Configurations*

You will see all repository purge configurations and all directory purge configurations.

- Note:\* If you have distributed builds enabled, then repository purging is not currently available on build agents. See [CONTINUUM-2697](#) for information.

Repository Purge Configurations

Repository	Days Older	Retention Count	Delete All	Delete Released Snapshots	Schedule	Default	Enabled	Description			
repository1	100	2	false	false		true	true				

Add

Directory Purge Configurations

Directory Type	Days Older	Retention Count	Delete All	Schedule	Default	Enabled	Description			
releases	2	3	false	DEFAULT_SCHEDULE	false	false				

Add

Here you can add/edit/delete the purge configuration.

You can also purge the repository or directory by clicking the corresponding purge icon of the purge configuration.



*purge now icon*

#### 40.1.1 Adding / Editing Repository Purge Configuration

Specify the repository to purge.

Add/Edit Purge Configuration

<b>Repository*:</b>	repository1
<b>Days Older:</b>	100
<b>Retention Count:</b>	2
<input type="checkbox"/>	Delete All
<input type="checkbox"/>	Delete Released Snapshots
<input type="checkbox"/>	Is it Default?
<b>Schedule:</b>	
<b>Description:</b>	
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

#### 40.1.2 Adding / Editing Directory Purge Configuration

There are two types of directory purge configuration:

- releases - this will purge the working copies associated with releases.
- buildOutput - this will purge the build output of each project.

**Add/Edit Purge Configuration**

<b>Directory Type:</b>	releases
<b>Days Older:</b>	100
<b>Retention Count:</b>	2
<input type="checkbox"/> Delete All <input type="checkbox"/> Is it Default?	
<b>Schedule:</b>	
<b>Description:</b>	
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

#### 40.1.3 Purging Repository

- Delete all - will delete the entire contents of the repository
- Days Old - will delete the artifact if it's older than the number of days old but still satisfies the number of retention count.

Example:

Days\_Old = 30 days

Retention\_Count = 2

If I have 5 artifacts that are older than 30 days, only 3 of them will be deleted because I have to retain any 2 artifacts (Retention count).

- Retention Count - the number of artifacts to retain; all in excess will be deleted
- Delete Released Snapshots - will delete all released snapshots

#### 40.1.4 Purging Directory

- Delete all
  - Releases Directory - will delete all releases-\* directories
  - Build Output Directory - will delete all build output of each project
- Days Old - will delete the directory if it's older than the number of days old
- Retention Count - number of directories to retain

#### 40.1.5 Steps in purging:

- Perform Delete All purging if it is checked.
- If delete all is not checked, then is days old has a value greater than 0? If it has then perform days old purging.
- If days old is 0 then perform retention count purging.
- Delete released snapshots purging will always be performed if it is checked, unless the Delete All is also checked or it's a directory purging.

## 41 Managing Parallel Builds

### 41.1 Managing Parallel Builds

Since 1.3.1, Continuum can be configured to checkout and build projects in parallel or concurrently.

#### 41.1.1 Configuring Parallel Builds

The number of build queues to be used when checking out or building projects can be configured in the Configuration page, via the Number of Allowed Builds in Parallel field (see [Managing General Configuration](#)). The value should be greater than one (1) to be able to create a build queue aside from the default.

Another thing to note is that if Distributed Builds is enabled, Parallel Builds is automatically disabled. To make it short, you're only allowed to use one of these functionalities at a time in the current implementation.

To configure parallel builds:

1. Create a build queue to be used for the projects. Refer to [Managing Build Queues](#)
2. Attach the build queue(s) to a build schedule. Refer to [Attaching Build Queue to a Schedule](#)
3. Create or edit the project's build definition to use the build schedule configured with the build queue(s). Refer to [Managing Build Definitions](#)

A screen similar to the following is displayed under the **Queues** page when the project is built (force/sheduled).

Current Build			
Build Queue	Project Name	Build Definition	
BUILD_QUEUE_2	Maven Clean Plugin	Default Maven 2 Build Definition	
DEFAULT_BUILD_QUEUE	Maven Assembly Plugin	Default Maven 2 Build Definition	
BUILD_QUEUE_3	Maven Install Plugin	Default Maven 2 Build Definition	

Continuum - Build Queue			
Build Queue	Project Name	Build Definition	
<input type="checkbox"/> BUILD_QUEUE_2	Maven Deploy Plugin	Default Maven 2 Build Definition	
<input type="checkbox"/> DEFAULT_BUILD_QUEUE	Maven Changelog Plugin	Default Maven 2 Build Definition	

Cancel Entries

Current Checkout	
Build Queue	Project Name

Checkout Queue	
Build Queue	Project Name

Cancel Entries

### Queues

#### 41.1.2 Limitations

Below are some feature limitations for the current implementation:

1. Project groups cannot be built multiple times simultaneously.
2. Concurrent build of inter-dependent projects in a group is not yet supported.
3. All projects in a project group will be enqueued to the same [Build Queue](#).

## 42 Managing Build Queues

---

### 42.1 Managing Build Queues

#### 42.1.1 Adding a Build Queue

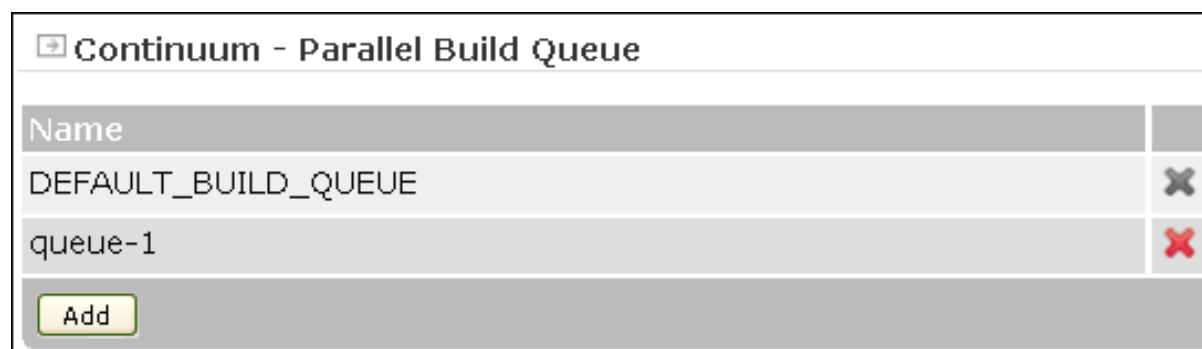
By default, there is already a DEFAULT\_BUILD\_QUEUE that cannot be deleted configured in Continuum. To add more build queues..

Click the **Build Queue** link under the **Administration** section.



*Build Queue*

You will see all the build queues.



*Build Queues list*

Here you can add/delete the build queue. Take note that you would only be allowed to create N number of build queues, where N = the Number of Allowed Builds in Parallel set in the Configuration page.

Input the name for the build queue and save.



*Add Parallel Build Queue*

#### 42.1.2 Deleting a Build Queue

From the build queue page, you can delete a build queue by clicking the delete icon



*Delete icon*


## 43 Managing Build Agents

---

### 43.1 Managing Build Agents

Before this, you may want to enable the Distributed Builds option in the [General Configuration](#) and [install a Build Agent](#), or read about the [Distributed Builds](#) feature.

From the menu, choose the **Build Agents** item



*Build Agents Menu*

You will see a list of the existing Build Agents, if any.

Build Agents			
Build Agent URL	Enabled	Description	
<a href="http://builds.example.com:8181/continuum-buildagent/xmlrpc">http://builds.example.com:8181/continuum-buildagent/xmlrpc</a>	false		 
<input type="button" value="Add"/>			

*Build Agents List*

To add a new Build Agent, click the **Add** button. You will be prompted to enter the Build Agent URL and Description.

To edit an existing Build Agent, click the **Edit** link to the right of the entry. You will be prompted to edit the Build Agent Description.

**Note:** Once a Build Agent has been saved you can not change its URL.

Continuum - Add/Edit Build Agent	
Build Agent URL*:	<input type="text" value="http://builds.example.com:8181/continuum-buildagent/xmlrpc"/>
Description:	<input type="text"/>
	<input type="checkbox"/> Enabled
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

*Add/Edit Build Agent*

Be sure to enable the agent by checking the **Enabled** checkbox.

After you save your changes, verify that the **Enabled** checkbox is checked. If it is not checked, most likely Continuum was unable to ping the agent. Consult the log file for additional information.



## 44 Managing Build Agent Groups

### 44.1 Managing Build Agent Groups

A Build Agent Group provides a logical association of Build Agents so that it can be attached to a Build Environment providing the user control on which agents it wants the project to be built.

From the menu, choose the 'Build Agents' item



*Build Agents Menu*

You will see a list of the existing Build Agent Groups in the Build Agent Groups Section, if any.

Build Agent Groups	
Name	Build Agents
linux-OS-agents	<ul style="list-style-type: none"> <li>• http://localhost:9191/xmlrpc</li> </ul>

Add

*Build Agent Groups List*

To add a new Build Agent Group, click the 'Add' button.

To edit an existing Build Agent Group, click the 'edit' link to the right of the entry.

You will be prompted to enter or edit the Build Agent Group Name and its associated Build Agent(s):

Add/Edit Build Agent Group	
Name*	linux-OS-agents
Configure the used Build Agents:	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>--- Available Build Agents ---</p> </div> <div style="width: 45%;"> <p>--- Used Build Agents ---</p> <p>http://localhost:9191/xmlrpc</p> </div> </div> <div style="display: flex; justify-content: center; gap: 10px; margin-top: 5px;"> <span>&lt;</span> <span>&gt;</span> <span>&lt;&lt;</span> <span>&gt;&gt;</span> </div>
Save	Cancel

*Add/Edit Build Agent Group*

Build Agent can be associated to one or more Build Agent Group

To use a Build Agent Group in a Project, you need to attach it to a build environment.

Refer to [Attach Build Agent Group to a Build Environment](#)

## 45 Managing Project Queues

---

### 45.1 Managing Queues

Current checkouts and builds, and all other projects queued in the checkout, build, and prepare build queues can be viewed in the **Queues** page.

Queues

*Queues*

#### Prepare Build Queue

Current Prepare Build

Project Group	SCM Root URL
Continuum 1.3.x Branch	scm:svn:http://svn.apache.org/repos/asf/continuum/branches/continuum-1.3.x

Prepare Build Queue

<input type="checkbox"/> Project Group	SCM Root URL	
<input type="checkbox"/> Continuum 1.4.x Trunk	scm:svn:http://svn.apache.org/repos/asf/continuum/trunk	

Cancel Entries

*Prepare Build Queue*

#### Build Queues

Current Build

Project Name	Build Definition	
Continuum Project	clean install	

Continuum - Build Queue

<input type="checkbox"/> Project Name	Build Definition	
<input type="checkbox"/> Continuum Configuration	clean install	
<input type="checkbox"/> Continuum Model	clean install	
<input type="checkbox"/> Continuum API	clean install	
<input type="checkbox"/> Continuum Test	clean install	
<input type="checkbox"/> Continuum Reports	clean install	
<input type="checkbox"/> Continuum Store	clean install	
<input type="checkbox"/> Continuum Commons	clean install	
<input type="checkbox"/> Continuum Security	clean install	
<input type="checkbox"/> Continuum Release System	clean install	
<input type="checkbox"/> Continuum Notifiers	clean install	




*Queues View*

#### Checkout Queue

[-] **Current Checkout**

Build Queue	Project Name	
DEFAULT_BUILD_QUEUE	<a href="#">Continuum :: Distributed Build :: Build Agent API</a>	

[-] **Checkout Queue**

<input type="checkbox"/>	Build Queue	Project Name	
<input type="checkbox"/>	DEFAULT_BUILD_QUEUE	<a href="#">Continuum :: Distributed Build :: Build Agent :: Core</a>	
<input type="checkbox"/>	DEFAULT_BUILD_QUEUE	<a href="#">Continuum :: Distributed Build :: Build Agent :: Webapp</a>	
<input type="checkbox"/>	DEFAULT_BUILD_QUEUE	<a href="#">Apache Continuum Build Agent</a>	

*Checkout Queue*

You must have a **Manage Continuum Queues** role for you to:

1. Cancel current checkouts, current builds, queued prepare builds, queued checkouts and queued builds by clicking the



*cancel build icon.*

2. Cancel multiple queued tasks by selecting the corresponding checkboxes of the checkouts, builds, or project updates you want to cancel and then click the **Cancel Entries** button.

Note: Cancelling current prepare build is not allowed to prevent data corruption.

## 46 External databases

---

### 46.1 How to use an external database?

#### 46.1.1 Supported databases

- Derby
- MySQL
- Postgres
- MS SQL Server (with modifications)
- Oracle (with modifications)

#### 46.1.2 Configure Continuum

There is some information in the installation guides for the [standalone Continuum installation](#) and the [Continuum web application deployed to Tomcat](#).

##### 46.1.2.1 Standalone version

To use an external database with Continuum standalone, you must configure DataSources in `$CONTINUUM_HOME/contexts/continuum.xml`

```

<!-- continuum database -->

<New id="continuum" class="org.eclipse.jetty.plus.jndi.Resource">
  <Arg><Ref id="wac"/></Arg>
  <Arg>jdbc/continuum</Arg>
  <Arg>
    <New class="org.apache.commons.dbcp.BasicDataSource">
      <Set name="driverClassName">org.apache.derby.jdbc.EmbeddedDriver</Set>
      <Set name="url">jdbc:derby:<Property name="appserver.base" default="."/>/da
      <Set name="username">SA</Set>
      <Set name="password"></Set>
      <Set name="testOnBorrow">>true</Set>
      <Set name="validationQuery">SELECT count(*) FROM SYS.SYSTABLES</Set>
    </New>
  </Arg>
</New>

<!-- Users / Security Database -->

<New id="users" class="org.eclipse.jetty.plus.jndi.Resource">
  <Arg><Ref id="wac"/></Arg>
  <Arg>jdbc/users</Arg>
  <Arg>
    <New class="org.apache.commons.dbcp.BasicDataSource">
      <Set name="driverClassName">org.apache.derby.jdbc.EmbeddedDriver</Set>
      <Set name="url">jdbc:derby:<Property name="appserver.base" default="."/>/da
      <Set name="username">SA</Set>
      <Set name="password"></Set>
      <Set name="testOnBorrow">>true</Set>
      <Set name="validationQuery">SELECT count(*) FROM SYS.SYSTABLES</Set>
    </New>
  </Arg>
</New>

```

In addition, you must add the JDBC driver to the `$CONTINUUM_HOME/lib` directory, and update `$CONTINUUM_HOME/conf/wrapper.conf` to list the driver instead of Derby. For example:

```
wrapper.java.classpath.15=%REPO_DIR%/postgresql-8.4-702.jdbc3.jar
```

If you use MS SQL Server or Oracle, you will need to adjust the `$CONTINUUM_HOME/apps/continuum/WEB-INF/classes/META-INF/plexus/application.xml` file to change the `javax.jdo.option.Mapping` value from default to `mssql` or `oracle` respectively. Bear in mind that this change will need to be made again across release upgrades.

```

<otherProperties>
  <property>
    <name>javax.jdo.option.Mapping</name>
    <value>default</value>
    <!-- change this property for SQL Server or Oracle support - CONTINUUM-697 -->
    <!-- value>mssql</value -->
    <!-- value>oracle</value -->
  </property>

```

#### 46.1.2.2 Web Application

To use an external database with the Continuum web application, you should configure the `DataSource` in your container.

Make sure the JDBC driver is available in the expected location for your container.

If you use MS SQL Server or Oracle, you will need to adjust the `WEB-INF/classes/META-INF/plexus/application.xml` file in the unpacked web application to change the `javax.jdo.option.Mapping` value from `default` to `mssql` or `oracle` respectively. Bear in mind that this change will need to be made again across release upgrades.

```

<otherProperties>
  <property>
    <name>javax.jdo.option.Mapping</name>
    <value>default</value>
    <!-- change this property for SQL Server or Oracle support - CONTINUUM-697 -->
    <!-- value>mssql</value -->
    <!-- value>oracle</value -->
  </property>

```

#### 46.1.3 Shutdown Procedure

When using an external database, it is very important to stop Continuum prior to stopping or restarting the database. Continuum may exhibit unpredictable behavior if the database disappears while it is running, and may not recover once the database comes back up. If this happens, you must re-start Continuum.

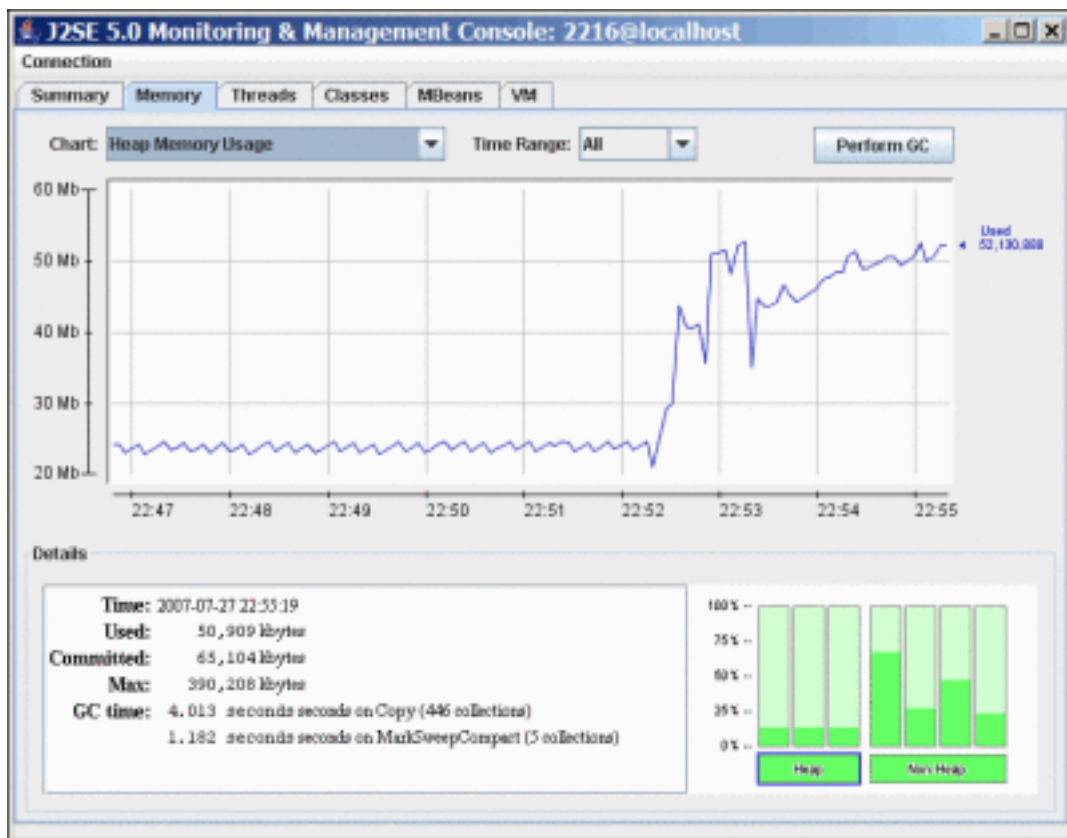
## 47 Monitoring Continuum

### 47.1 Monitoring a Continuum instance

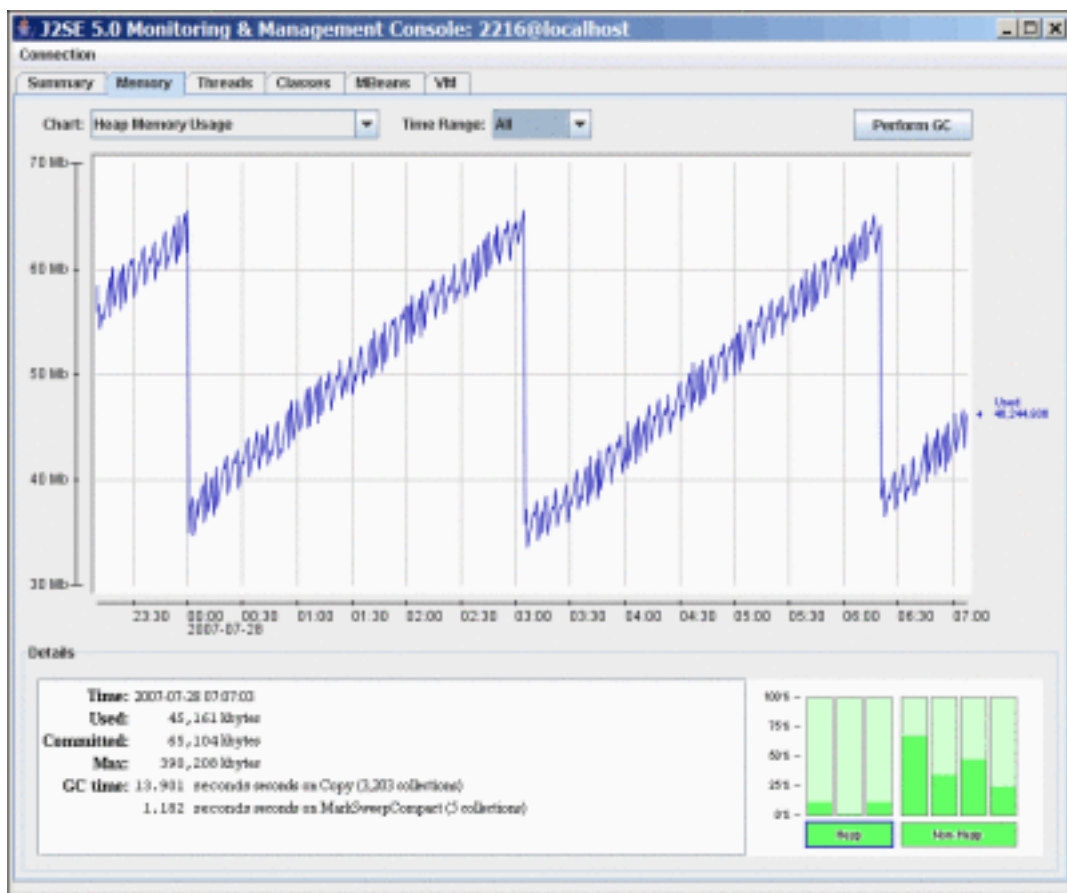
#### 47.1.1 JConsole

- Edit `$CONTINUUM_HOME/bin/[platform]/wrapper.conf` to add:  
`wrapper.java.additional.9=-Dcom.sun.management.jmxremote`
- Start Continuum
- Start jconsole, (included with Sun JDK 1.5 and later,) select the 'WrapperSimpleApp' line, and click Connect

The following image shows starting Continuum 1.1, adding a project group, then adding and building the trunk of the Shale Framework (14 modules):



The following image shows the same Continuum instance running for several hours:



### 47.1.2 Garbage Collection and Heap Size

Edit `$CONTINUUM_HOME/bin/[platform]/wrapper.conf` to add:

```
wrapper.java.additional.10=-verbose:gc
```

or

```
wrapper.java.additional.10=-Xloggc:gc.txt
```

The first sends to the console (and wrapper log) while the second sends to the filename that you tell it to.

Example output:



```
420.568: [GC 62459K->58474K(65104K), 0.0074933 secs]
420.778: [GC 62634K->58733K(65104K), 0.0060762 secs]
420.919: [GC 62893K->58832K(65104K), 0.0034433 secs]
421.015: [GC 62992K->58723K(65104K), 0.0026420 secs]
421.067: [GC 62883K->60885K(65104K), 0.0107248 secs]
421.140: [GC 65045K->62114K(66384K), 0.0122194 secs]
421.153: [Full GC 62114K->32136K(66384K), 0.3591207 secs]
421.655: [GC 36296K->32242K(65104K), 0.0037377 secs]
421.754: [GC 36402K->33061K(65104K), 0.0037182 secs]
421.835: [GC 37221K->33120K(65104K), 0.0028722 secs]
421.930: [GC 37280K->34582K(65104K), 0.0056601 secs]
422.074: [GC 38742K->35150K(65104K), 0.0072451 secs]
```

before->after (total), time

This shows the heap size before and after garbage collection as well as the total heap size and the time for the GC run.

### 47.1.3 References

- <http://java.sun.com/developer/technicalArticles/J2SE/jconsole.html>

## 48 Log Files

---

### 48.1 Log Files

To keep track of the Continuum performance and problems, log files are created during runtime. These files can be found in the `logs/` directory.

- **continuum.log** - contains all the start-up information for Continuum.
- **continuum-audit.log** - contains information regarding the project operations and configurations being modified. For example, projects added are logged here, with date and timestamp, `userId` of who performed the deploy, and the project that was built. For security measure, passwords are masked behind asterisks.
- **continuum-security-audit.log** - contains information regarding Continuum's security. For example, a successful login of a user or a user account is created.

## 49 Audit Logs

---

### 49.1 Audit Logs

Continuum's `logs` directory contains an audit log file named `continuum-audit.log`, which tracks events that occur in the server. For security measure, passwords are masked behind asterisks.

A typical record looks like this:

```
2009-07-22 12:27:58 - admin - BUILD_SCHEDULE BI_HOURLY - Added Build Schedule
2009-07-22 12:28:23 - admin - BUILD_SCHEDULE BI_HOURLY - Modified Build Schedule
2009-07-22 12:29:56 - admin - PROJECT http://svn.apache.org/repos/asf/continuum/trunk/p
2009-07-22 12:31:00 - admin - PROJECT Project Group id=6 - Forced Project Build
```

The space delimited records are:

- date and time (server local time)
- user that enacted the change (or guest if none)
- the project or configurations affected
- the event that occurred

Currently, the following events are logged:

- add/delete of projects
- forced project builds
- release prepare/perform/rollback
- add/edit/delete of schedules
- add/edit/delete of build definition templates

## 50 Continuum Logs

---

### 50.1 Continuum Logs

Continuum's `logs` directory contains an continuum log file named `continuum.log`, which logs all the startup information of Continuum.

A typical record looks like this:

```
2009-07-22 11:57:48,397 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:48,397 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:48,912 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,081 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,082 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,082 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,083 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,083 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,083 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,083 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,083 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,083 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,083 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,083 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,083 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,084 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,084 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,084 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.Default
2009-07-22 11:57:49,084 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.initial
Continuum initializer running ...
2009-07-22 11:57:49,177 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.initial
Default Project Group exists
2009-07-22 11:57:49,177 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.initial
Continuum initializer end running ...
2009-07-22 11:57:49,186 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.build.s
Activating schedules ...
2009-07-22 11:57:49,236 [WrapperSimpleAppMain] INFO org.apache.maven.continuum.build.s
DEFAULT_SCHEDULE: next fire time ->Wed Jul 22 12:00:00 PHT 2009
```

## 51 Security Logs

---

### 51.1 Security Logs

Continuum's `logs` directory contains a security log file named `continuum-security-audit.log`, which keeps track of all the security operations.

A typical record looks like this:

```
2009-07-22 12:32:00 - admin - User Created: continuum
2009-07-22 12:32:10 - admin - Role Assigned to user continuum: Continuum Group Project
2009-07-22 12:32:17 - admin - User Modified: continuum
2009-07-22 12:32:34 - - Logged Out user admin
2009-07-22 12:32:47 - - Successful Login for user continuum
```

The hyphen delimited records are:

- date and time (server local time)
- current user performing the operation
- the operation performed

Currently, the following events are logged:

- user creation/modification/deletion
- user log in/out
- assigning roles to a user

## 52 Appearance Configuration

---

### 52.1 Configure Appearance

#### 52.1.1 Banner

You can configure the right logo of the banner including title and link on the image. You have to add some information in a POM (like a company POM) (coming from org.apache:apache in this example):

```
<organization>
  <name>The Apache Software Foundation</name>
  <url>http://www.apache.org/</url>
</organization>
<properties>
  <organization.logo>http://www.apache.org/images/asf_logo_wide.gif</organization
</properties>
```

From the menu, choose the **Appearance** entry :

### Appearance

#### Company Details

The logo in the top right of the screen is controlled by your selected 'company POM'.

You have not yet specified a company POM. [Select a Company POM](#)

*Appearance*

Use the **Select a Company POM** link :

### Appearance

#### Company Details

Enter the details of the company super POM below. If it exists, the organization name, URL and logo will be read from it.

Group ID:

Artifact ID:

*'Select a Company POM'*

The result will be displayed (using org.apache:apache in this example):



Lamy (admin) - Editer les Détails - Se déconnecter

Continuum | Maven | Apache

## Appearance

### Company Details

[Edit](#)

The logo in the top right of the screen is controlled by your selected 'company POM'.

Your selected company POM is below. If you would like to change the organization name, url or logo, you can [edit the POM](#).

Group ID:	org.apache
Artifact ID:	apache
Version:	4

[Edit Company POM](#)

### POM Information

Name	The Apache Software Foundation
URL	<a href="http://www.apache.org/">http://www.apache.org/</a>
Logo URL	<a href="http://www.apache.org/images/asf_logo_wide.gif">http://www.apache.org/images/asf_logo_wide.gif</a>

*'Selected a Company POM'*

**Note:** the POM is searched for in the central repository and in the repositories available for the active profiles from your `$ user.home/.m2/settings.xml`.

### 52.1.2 Footer

You can configure the footer by putting your own html content in the **Appearance** entry of the menu.

#### footer content

HTML Content:	<code>&lt;div class="xright"&gt;Copyright &amp;copy; 2005-2007&amp;nbsp;The Apache Software Foundation&lt;/div&gt; &lt;div class="clear"&gt;&lt;hr/&gt;&lt;/div&gt;</code>
<input type="button" value="save"/> <input type="button" value="Annuler"/>	

*'Configure footer'*

The default value is :

```
<div class="xright">Copyright &copy; 2005- $\{currentYear\}$ &nbsp;The Apache Software F
```

## 53 Build Definition Template

### 53.1 Build Definition Template

From the menu, choose the 'Build Definition Templates' entry

Build Definitions Templates

*Build Definition Templates*

You will see all build definition templates and the build definitions which can be added in templates. In this screen you can add/edit/update/delete a build definition ( **the Continuum defaults can be deleted**)

Available Templates

Name	
default ant template	
default maven1 template	
default maven2 template	
default shell template	

Add

Available Build Definitions

Goals	Arguments	Build File	Schedule	Profile	Is Build Fresh ?	Default	Description	Type	
		build.xml	DEFAULT_SCHEDULE		false	true	default ant buildDefinition	ant	
clean:clean jar:install		project.xml	DEFAULT_SCHEDULE		false	true	default maven1 buildDefinition	maven-1	
clean install	--batch-mode --non-recursive	pom.xml	DEFAULT_SCHEDULE		false	true	default maven2 buildDefinition	maven2	
			DEFAULT_SCHEDULE		false	true	default shell buildDefinition	shell	

Add

*Build Definition Templates*

In the build definition template edit screen, you can easily one or more build definitions.

#### Build Definition Template

Name\*: hourly build midnight site build

Configure the used Build Definitions:

--- Available Build Definitions ---

- default ant buildDefinition
- default maven1 buildDefinition
- default shell buildDefinition

<-

->

<<--

-->>

<\*>

--- Used Build Definitions ---

- default maven2 buildDefinition
- site build

Save Cancel

*Build Definition Template Edit*



## 54 Shutdown Continuum

---

### 54.1 Shutdown Continuum

Shutting down Continuum while a project is building or queued is NOT recommended. The build queues page should be used to cancel any currently running and queued builds before shutting down.

#### 54.1.1 Queues view

From the menu, choose the 'Queues' link



*Queues*

Then you'll get the Queues page

Current Build		
Project Name	Build Definition	
Continuum Project	clean install	

Continuum - Build Queue		
<input type="checkbox"/> Project Name	Build Definition	
<input type="checkbox"/> Continuum Configuration	clean install	
<input type="checkbox"/> Continuum Model	clean install	
<input type="checkbox"/> Continuum API	clean install	
<input type="checkbox"/> Continuum Test	clean install	
<input type="checkbox"/> Continuum Reports	clean install	
<input type="checkbox"/> Continuum Store	clean install	
<input type="checkbox"/> Continuum Commons	clean install	
<input type="checkbox"/> Continuum Security	clean install	
<input type="checkbox"/> Continuum Release System	clean install	
<input type="checkbox"/> Continuum Notifiers	clean install	

*Queues*

In this page you can see the current build, projects in the build queue, the current checkout and projects in the checkout queue. For each line, you can cancel the process

#### 54.1.2 Shutdown

To shutdown Continuum, the queue must be empty, so when you want to shutdown, you can either wait until all builds are complete or cancel them.

If you don't want to have new projects added in the build queue before you shutdown Continuum, you can disable all schedules. You'll need to re-enable them on the next startup. (In the future, you'll have an action to automate the disable/enable schedules process.)

## 55 Understanding Distributed Builds

---

### 55.1 Understanding Distributed Builds

#### 55.1.1 Introduction

The Distributed Builds is an extension to the base Continuum functionalities that gives us the ability to process multiple independent builds beyond the capacity of a single server's processing power. It also enables us to execute builds on multiple different platforms while retaining a unified view of all project builds.

#### 55.1.2 Architecture

Continuum follows a Client-Server model using XML-RPC as the protocol. However, since it uses a bi-directional XML-RPC implementation, we instead distinguish the components by calling them Master and Build Agent.

The Master is a Continuum instance that has the ability to delegate the builds to registered Build Agents.

The Build Agent is a standalone Jetty-bundled webapp that listens for any build requests from the Master it is assigned to.

There is a one-to-many relationship between the Master and the Build Agents. A Master may have many Build Agents, but each Build Agent can only have one Master.

#### 55.1.3 Behavior

Distributed Builds happen at the project group level of Continuum. When the entire project group is built in the Master, independent projects (single project or multi-module project) are distributed to a single registered Build Agent.

In a project group containing a mix of projects, the distribution of work goes through the following steps:

1. In the Master, a build in the project group is triggered.
2. Every independent project within the project group is identified, whether as a single project or a multi-module project. Projects with inter-dependencies cannot be distributed separately, so multi-module projects are delegated to a Build Agent as one build.
3. For each independent project, the Master iterates over the list of registered Build Agents and queries each if available. The query is an XML-RPC `ping()` followed by a `getBuildSizeOfAgent()` invocation.
4. If there is a Build Agent available, the Master collects the information necessary for the build (SCM URL, project id, etc.) and passes it when invoking `buildProjects()` to the Build Agent with the smallest number of tasks in its queue. The Master also passes the name of the local repository of the project's project group. Make sure that a local repository with the same name is configured in the Build Agent's configuration file since that is what will be used when the project is built in the build agent. Otherwise, it will use the local repository set in the build agent's `settings.xml` file.
5. In the Build Agent, the build request is processed: the build is queued and executed. Upon execution, the Build Agent first performs an SCM checkout or an SCM update followed by an SCM changelog to get the latest update date, then the actual build follows.

6. At this point, when the build is running, the Master can invoke `cancelBuild()` which returns a transient build result, and `getBuildResult()` that updates the build output viewed in the Master.
7. After the build, the Build Agent returns the complete build result to the Master by invoking the callback method `returnBuildResult()`, which the Master aggregates to provide a unified view of projects.

A `ping()` is always called before each XML-RPC invocation to check if the agent is available. If it's not, the agent will be disabled.

#### 55.1.4 Setup

- [Enable](#) the Distributed Builds option and set the Shared Secret Password in the General Configuration.
- [Install and Configure](#) one or more Build Agents.
- [Add](#) your Build Agents to the Continuum Master.
- [Add](#) your Build Agents to a Build Agent Group.
- [Add](#) your Build Agent Group to a Build Environment.
- Configure a project to use that Build Environment.
- Make sure that the [local repository](#) of the [project group](#) is properly configured. In this case, it should point to the path of the repository where the Build Agent is installed/running.

**Warning:** You need to have a central remote repository to store the artifacts created from the Build Agent so that other agents will be able to use the new artifacts.

#### 55.1.5 Limitations

- Only system administrator can enable/disable distributed builds
- Credentials (i.e. svn credentials) are passed along if specified, but if server cache is used it will need to be done individually on the Build Agents
- There is no tracking of SCM changes
- The Build Agent needs a configuration web interface
- All projects in a project group will be distributed to the same Build Agent

#### 55.1.6 Future Enhancements

- Remote builders
  - Builders can be installed on remote machines, a Continuum manager will send actions to run to builders. An action can be something to run on all builders, on some of them or eventually only to an available builder if we don't want to run more than one build. Actions can be sent with JMS and builders can apply some filters if they don't want to receive all actions. With that, we can do some parallel builds but the dependency tree must be respected for the build order. To work correctly with dependencies, each builder must use a central local repository. Maybe we can use an internal Archiva.
  - With Continuum builders configured to receive all commands, users can run multi-platform build for each build definition execution.
  - With Continuum builders configured to receive only some project types, users can use a different builder by project group. In this case, the build of all projects will be done quickly because commands are balanced on several servers.
  - With Continuum builders configured to build something when it is available, users can install builders on several machine to balance the charge. In this case, it will be possible to run some parallel builds.

- When the builders work is done, a message will be sent to the manager to notify the end of the process.
- With JMS used for the communication, we can add some listeners to create reports/statistics, log some information.
- Policy-based distribution
  - Next available
  - Load balanced
  - Targeted environment matching

## 56 WebDAV for Agent Working Copies

---

### 56.1 WebDAV support for working copies

The WebDAV support is added in the continuum build agent only for retrieving the working copies of projects.

#### 56.1.1 Example using Sardine

You can use [Sardine](#) for the WebDAV client.

The first sample will retrieve the directory listing for the project with id of 1.

```
Sardine sardine = SardineFactory.begin();  
List<DavResource> resources = sardine.getResources( "http://buildagenturl/workingcopy/1
```

The next sample will get an InputStream reference to a remote file.

```
Sardine sardine = SardineFactory.begin();  
InputStream is = sardine.getInputStream( "http://buildagenturl/workingcopy/1/pom.xml" )
```

## 57 Developer's Guides

---

### 57.1 Developer's Guides

- [Building Continuum](#)
- [XML-RPC](#)

## 58 Building Continuum

---

### 58.1 Guide to building Continuum

#### 58.1.1 Why would I want to build Continuum?

You might want to build Continuum yourself for one of two reasons:

- To build the source release of Continuum
- to try out a bleeding edge feature or bugfix (issues can be found in [JIRA](#)), but you can try our SNAPSHOTs: <http://vmbuild.apache.org/>
- to fix a problem you are having and submit a patch to the development team.

Note that for releases, you may also make use of pre-built binaries.

#### 58.1.2 Checking out the sources

All of the source code for Continuum and its related libraries is in a Subversion repository. You can also [browse the repository](#), or checkout specific modules directly.

All SVN instructions are available on the [Source Repository page](#).

#### 58.1.3 Building the sources

##### 58.1.3.1 Prerequisites

- JDK 5 or greater
- Maven 2.0.7 or greater

##### 58.1.3.2 Building

To build Continuum, you run this command from the top (trunk) directory:

```
mvn clean install
```

## 59 XML-RPC

---

### 59.1 Guide to use XML-RPC with Continuum

#### 59.1.1 Introduction

In this section, you'll learn how to connect to a Continuum instance and how to do some action on projects.

#### 59.1.2 Requirements

To connect to a Continuum instance, you must use the `continuum-xmlrpc-client` jar.

This library have some others jars as dependencies, so the best way to start the development of a Continuum xmlrpc client is to create a maven2 project with the following dependencies:

```
<dependency>
  <groupId>org.apache.continuum</groupId>
  <artifactId>continuum-xmlrpc-client</artifactId>
  <version>YOUR_CONTINUUM_VERSION</version>
</dependency>
```

#### 59.1.3 Connection to Continuum

To connect to your Continuum with the client API, you must use the `ContinuumXmlRpcClient` class.

The constructor use 3 parameters:

- url, the url of the xmlrpc listener that is `http://host:port/continuum/xmlrpc`
- user, a Continuum user
- password, the user's password

```
URL url = new URL( "http://localhost:8080/continuum/xmlrpc" );
ContinuumXmlRpcClient client = new ContinuumXmlRpcClient( url, username, password );
```

#### 59.1.4 Getting project groups list

You have two ways to get the project groups list. The first is to get only a summary of groups and the second returns groups with details. If you don't need all informations, we recommend to use the first way, so you'll save time to get datas and memory on the server.

```
List<ProjectGroupSummary> pgs = client.getAllProjectGroups();
List<ProjectGroup> pgs = client.getAllProjectGroupsWithProjects();
```

#### 59.1.5 Getting projects in a group

```
List<ProjectSummary> ps = client.getProjects( projectId );
```

If you already have a `ProjectGroup` or `ProjectGroupSummary` object, you can access to the project group id with this:

```
int projectId = pg.getId();
```



### 59.1.6 Building all projects in a group

- With the default build definition

```
client.buildGroup( projectId );
```

- With a build definition

```
client.buildGroup( projectId, buildDefinitionId );
```

### 59.1.7 Building a project

- With the default build definition

```
client.buildProject( project.getId() );
```

- With a build definition

```
client.buildProject( project.getId(), buildDefinitionId );
```

**Note:** When you start a build, the project is put in the Continuum queue and will be built when all projects added previously in the queue will be built.

### 59.1.8 Triggering a build

In some case, users want to use the push build technique with a hook in their SCM, so when a developer will commit some files, a build will be triggered. To do this, you can write a simple xmlrpc client that will use a project id as parameter and you'll use the `buildProject(...)` method described above.

This method will start a forced build.

### 59.1.9 Removing a project

A project can be removed by supplying the project id.

```
List<ProjectSummary> projects = client.getProjects( projectId );
...
client.removeProject( ps.getId() );
```

### 59.1.10 Removing build results

Currently, to remove a build result, you need the entire build result (rather than just the summary).

```
List<BuildResultSummary> results = client.getBuildResultsForProject( ps.getId() );
...
BuildResult br = client.getBuildResult( ps.getId(), brs.getId() );
client.removeBuildResult( br );
```

### 59.1.11 Editing build queue

You can edit all projects which are in the build queue or check if one project is currently in the build queue.

```
List<BuildProjectTask> prjsInBuildQueue = client.getProjectsInBuildQueue();
client.isProjectInBuildingQueue( int projectId );
```

### **59.1.12 Backup**

With the Continuum xmlrpc client, you can backup a full Continuum instance (without users database, for the moment).