

Tutorial: Install, setup and use the Jenkins Plugin for Dimensions CM

Pre-requisites

Dimensions CM with Tomcat installed

Step 1 Install Jenkins

Go to: <http://jenkins-ci.org/> and download the latest version of the Java Web Archive (.war)



Rename the downloaded **jenkins.zip** to **Jenkins.war** and place it into %DM_ROOT%\..\..\common tools\tomcat\8.0\webapps¹

Tomcat will unpack the war file into %DM_ROOT%\..\..\common tools\tomcat\8.0\webapps\Jenkins, allow about 5 minutes for this to complete.

Copy the files **darius.jar**, **dmclient.jar**, **dmfile.jar**, **dmnet.jar** from %DM_ROOT%\cm\java_api\lib into the Jenkins Servlet library location

e.g. %DM_ROOT%\..\..\common tools\tomcat\8.0\webapps\Jenkins\WEB-INF\lib

Restart the Serena Common Tomcat service

¹ Note: For Tomcat 7 installations this will be: %DM_ROOT%\..\..\common tools\tomcat\7.0\webapps

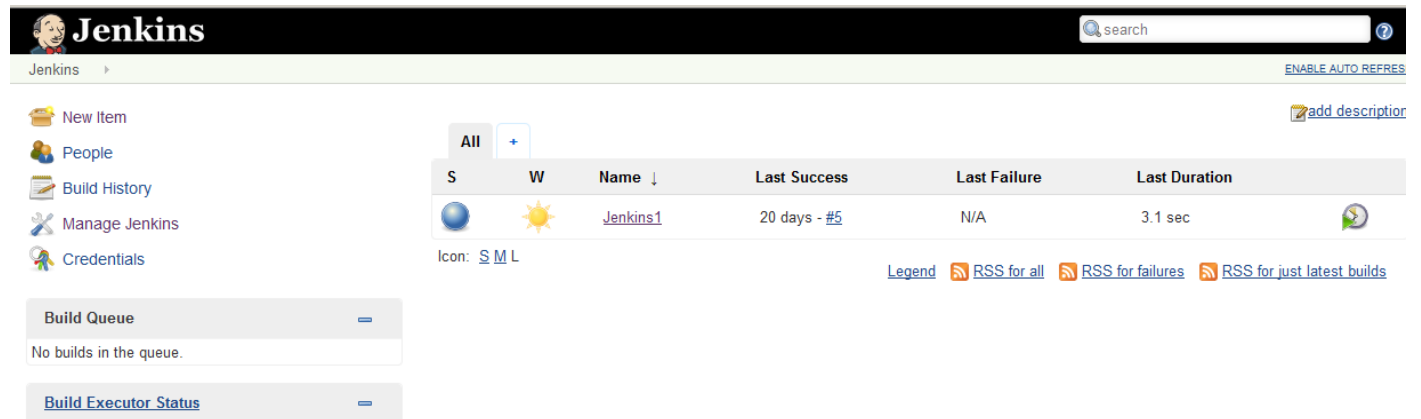
Step 2 Check Jenkins has installed

Start a support browser and browse to the URL:



`http://<SERVER>:<TOMCAT PORT>/Jenkins/`

NOTE: the Jenkins folder name detailed above must match the folder name under \$DM_ROOT which you have unpacked the .war file into.

A screen similar to the following should appear



The screenshot shows the Jenkins web interface. At the top, there is a search bar and a navigation menu. The main content area displays a table of build history for a job named 'Jenkins1'. The table has columns for 'S' (Success), 'W' (Warning), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. The 'Jenkins1' build is shown as successful, with a last success time of '20 days - #5' and a duration of '3.1 sec'. Below the table, there are links for 'Icon: S M L' and 'Legend'. On the left side, there is a 'Build Queue' section showing 'No builds in the queue.' and a 'Build Executor Status' section.

S	W	Name ↓	Last Success	Last Failure	Last Duration
		Jenkins1	20 days - #5	N/A	3.1 sec

Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

Step 3 Install the Jenkins Plugin

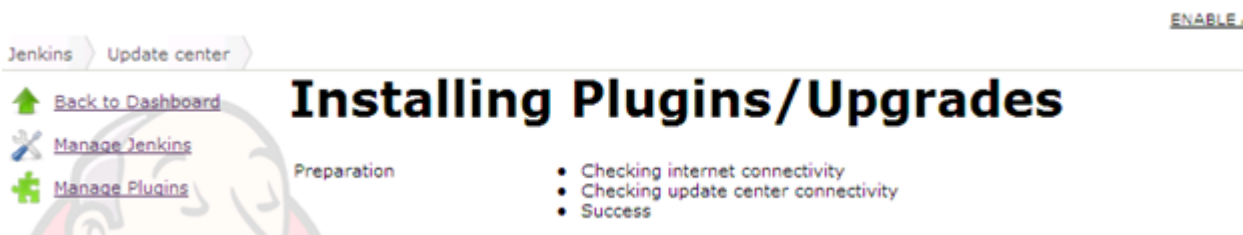
In the Jenkins dashboard click on **Manage Jenkins**, then **Manage Plugins**, then click on the **Available** tab and select the **Dimensions Plugin**.



Scroll to the bottom right-hand corner of the page and click on **Install without restart**.



The plugin should now automatically download and install, reporting



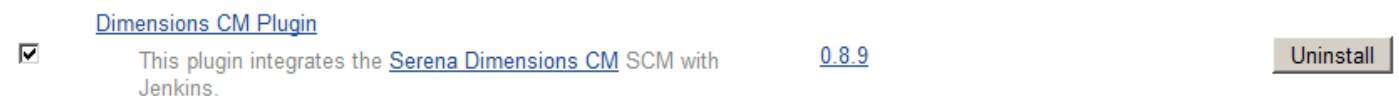
As reported on the Installation confirmation screen, Jenkins will need to be restarted. To do this,

Close down the browser

Restart the Serena Common Tomcat service

Restart the browser and return to the Jenkins URL

In the Jenkins dashboard click on **Manage Jenkins** then **Manage Plugins** then click on the **Installed** tab and the **Dimensions SCM Plugin** should now appear



2





² Note this version may change depending on the current version of the plugin

Step 4 Test the Jenkins Plugin

Click on **Back to Dashboard**, then **Manage Jenkins**, then **Configure System**, scroll down to the section entitled **Dimensions**.

Fill in the relevant login information

Dimensions

Login Name	<input type="text" value="dmsys"/>	
Password	<input type="password" value="••••"/>	
Server	<input type="text" value="STL-VM-PS-W2K8:671"/>	
Database	<input type="text" value="cm_typical@dim"/> <small>(the format used must be database@dsn)</small>	

Now click on **Check Connection...** which should respond with

Connection test succeeded!

Note: The installation details entered above will be used as the default Dimensions SCM repository for

Ensure the Option **Use update** is selected

Use update

If this option is checked, then the workspace will be automatically populated with the changed code from Dimensions. This option should only be used if you intend Hudson to populate the workspace for you.

Click on **Advanced...** and then specify GMT as the time zone.

Time zone |

Click on **Validate Time zone** which should respond with

Time zone

Timezone test succeeded!

Now click on the **Save** button at the very bottom-left of the page

Hudson URL

Step 5 Create a new Jenkins Job

At this point we will create a Jenkins project that builds independent of an SCM system.

In the Jenkins dashboard click on **Manage Jenkins** then **New Item**.

Specify an **Item name** e.g. Jenkins Project 1 and select the type as a **Freestyle project**. Then click on **OK**.

The screenshot shows the Jenkins 'New Item' configuration page. The 'Item name' field is set to 'Jenkins Project 1'. The 'Freestyle project' option is selected. The 'Build Queue' shows 'No builds in the queue.' and the 'Build Executor Status' shows '1 Idle' and '2 Idle'.

Item name

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Build multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

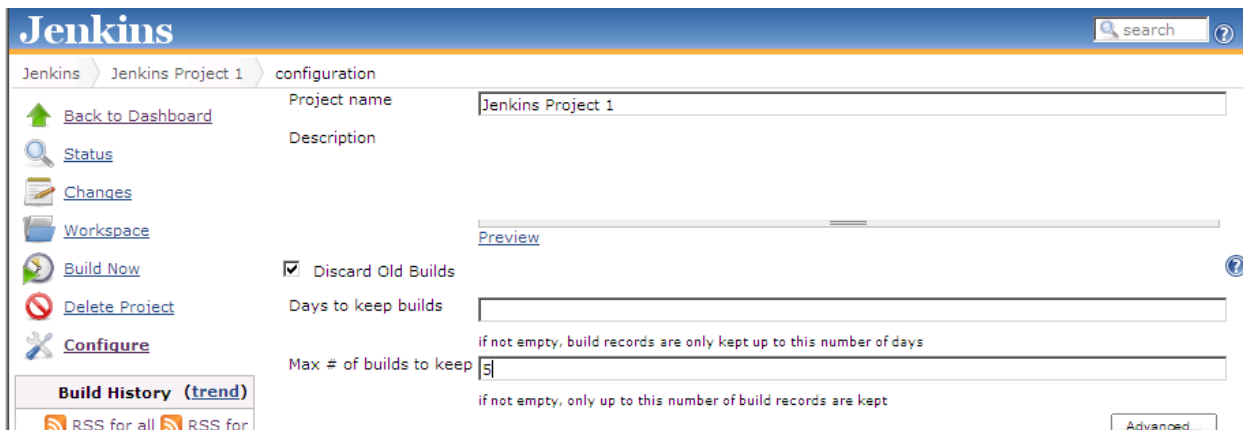
External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system. See [the documentation for more details](#).

Copy existing Item
Copy from

This will now present the project view

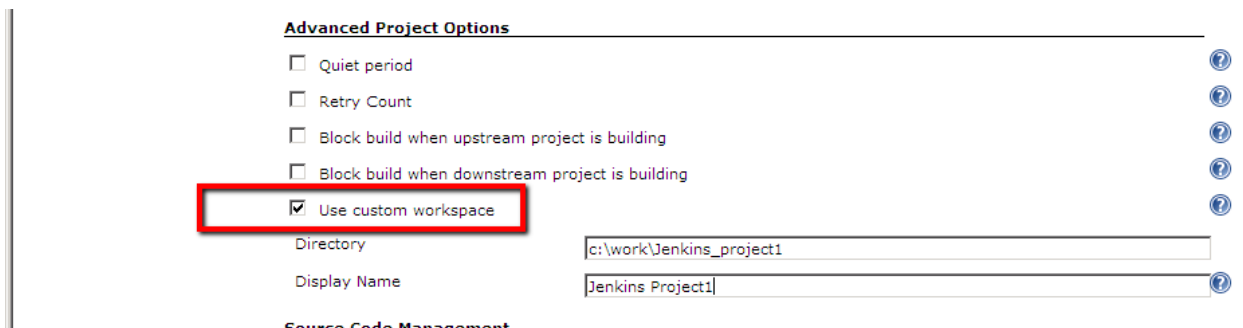
Activate the switch Discard Old Builds

Set Max # of builds to keep as 5



Jenkins Project 1 configuration page. The 'Discard Old Builds' checkbox is checked. The 'Max # of builds to keep' field is set to 5. The 'Days to keep builds' field is empty. The 'Advanced...' button is visible at the bottom right.

Expand **Advanced Project Options** and tick on **Use custom workspace**, specifying the directory as **c:\work\jenkins_project1**



Advanced Project Options section. The 'Use custom workspace' checkbox is checked and highlighted with a red box. The 'Directory' field is set to c:\work\jenkins_project1. The 'Display Name' field is set to Jenkins Project1.

Scroll down and under Build click on the Add build step button

Build

Add build step ▼

Post-build Actions

Select the option **Execute Windows batch command**, and specify the script detailed below



Execute Windows batch command configuration. The command field contains the following text:
cd c:\work\jenkins_project1
copy hello.c hello.exe

The click on the **Save** button at the bottom-left of the screen.

Now create the directory c:\work\jenkins_project1 on disc

In the directory create a simple text file called c:\work\jenkins_project1\hello.c

Step 6 Run the Jenkins job


In the project view click on the option **Build Now**



Once the Build has completed, under **Permalinks** click **Last Build**

Permalinks

- [Last build \(#5\), 20 days ago](#)
- [Last stable build \(#5\), 20 days ago](#)
- [Last successful build \(#5\), 20 days ago](#)

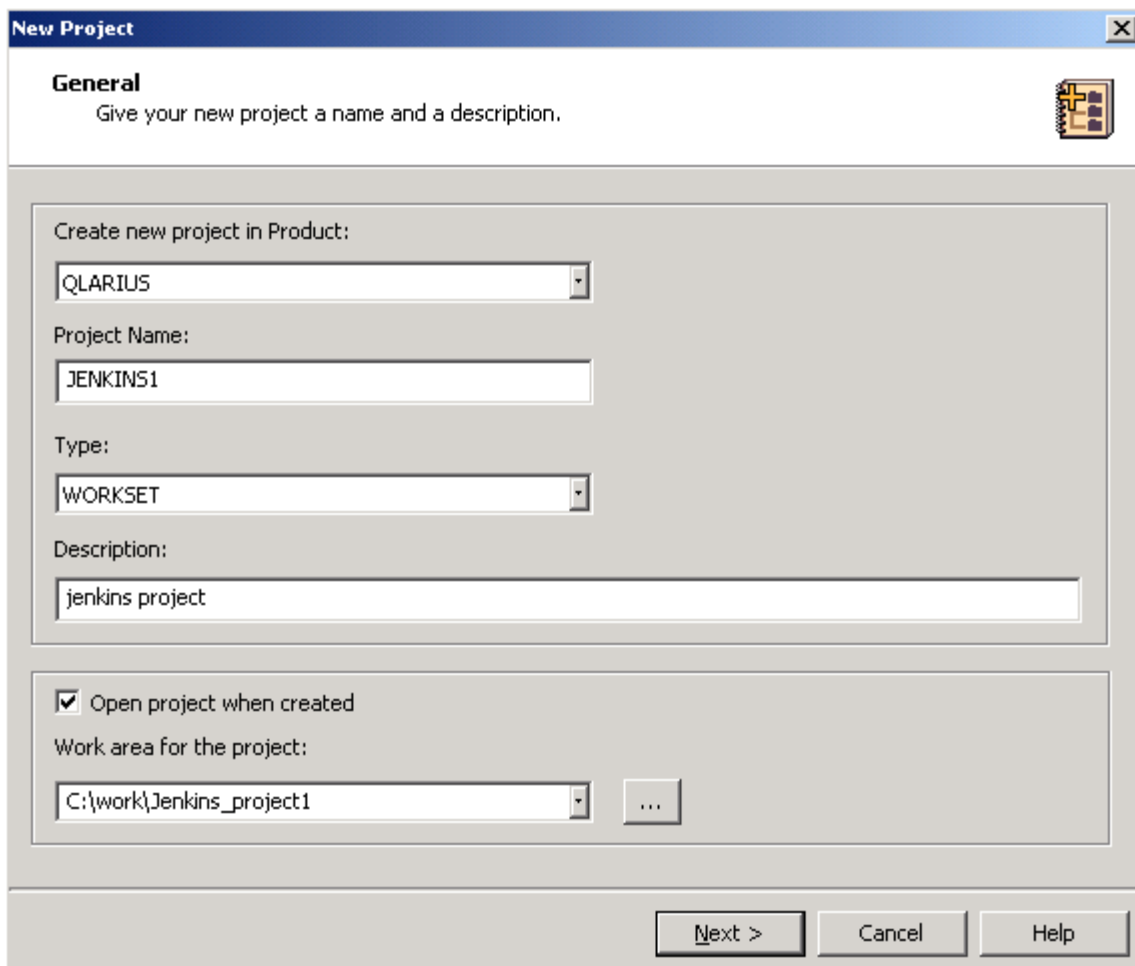
Then click on  **Console Output** to see the results of the build.

A successful Build should show Console output similar to the log below:



Step 7 Place the Build Sources into Dimensions CM

Create a new empty project named **Jenkins1** in Dimensions CM, using the default project settings.



New Project

General
Give your new project a name and a description.

Create new project in Product:
QLARIUS

Project Name:
JENKINS1

Type:
WORKSET

Description:
jenkins project

Open project when created

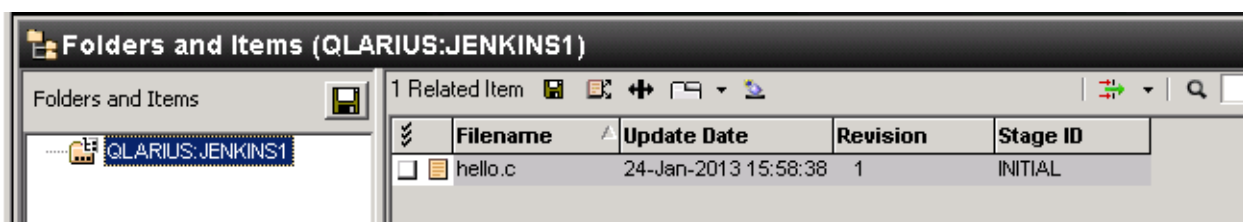
Work area for the project:
C:\work\Jenkins_project1

Next > Cancel Help

Open the project and set the Project working directory to be: c:\work\Jenkins_project1

Delete the file: `del c:\work\Jenkins_project1\hello.exe`

Now deliver the hello.c file into the project



Folders and Items (QLARIUS:JENKINS1)

1 Related Item

Filename	Update Date	Revision	Stage ID
hello.c	24-Jan-2013 15:58:38	1	INITIAL

Step 8 Configure the Jenkins Project to use Dimensions CM

Now click on **Back to Project** on the left-hand side of the screen, and then click on **Configure**. Under **Source Code Management** select **Dimensions**, specify the project **QLARIUS:JENKINS1**.

Tick the option to **Clear the contents of the workspace**.

Source Code Management

CVS

Dimensions

Project Name

QLARIUS:JENKINS1

(the format used must be *PRODUCT_ID:PROJECT_NAME* even for streams)

Folders To Monitor

Folder

Add folder...

Clear the contents of the workspace

Always force a build to occur

Overwrite any locally modified files in workspace

Save

Apply

Click on the **Advanced** button for this section and specify the relevant Dimensions login details, click on the **Check Connection** button to ensure the login works.

Login Name	jov. dmsys
Password
Server	stl-vm-ps-w2k8
Database	cm_typical@dim (the format used must be <i>database@dsn</i>)
Connection test succeeded!	Check Connection...

After testing the connection tick the box **Use Update** directly below the connection details entered.

Use Update

If this option is checked, then the workspace will be automatically populated with the changed code from Dimensions using the UPDATE or DOWNLOAD command. This option should only be used if you intend Hudson to populate the workspace for you.

Scroll further down to the **Time Zone** field, specify the time zone as GMT, and click on **Validate Time zone** to confirm.

Time zone

GMT

Timezone test succeeded!

Validate Timezone...

Scroll to the **Execute Windows batch command** and remove the CD command³

Execute Windows batch command

Command `copy hello.c hello.exe`

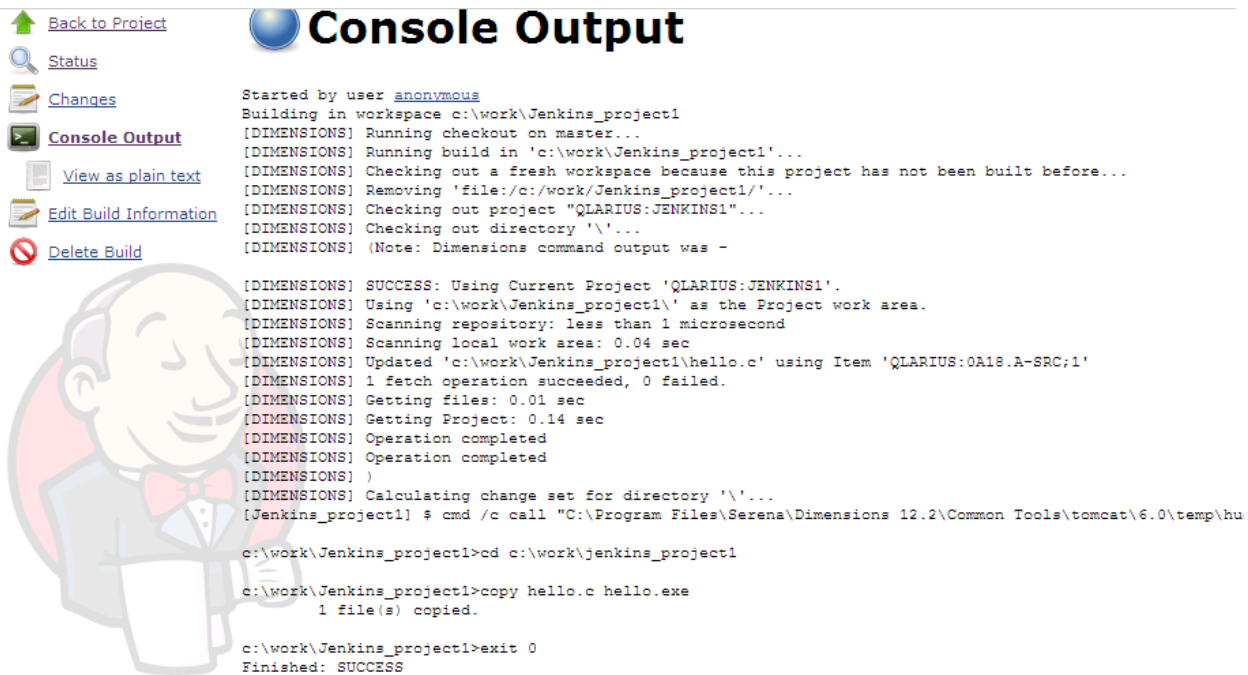
Scroll to the bottom of the screen and click on the **Save** button, to save all the changes made.



³ This is because Jenkins will use its own temporary area for receiving incoming Dimensions CM files and use that workspace to perform any builds.

Step 9 Re-run the Build

Click on the **Build Now** option on the left-hand side of the screen, a new job should be generated and once it has completed click on **Latest Console output**, which should produce something similar to the log below:



The screenshot displays the Jenkins 'Console Output' page. On the left, there is a navigation menu with options: 'Back to Project', 'Status', 'Changes', 'Console Output' (highlighted), 'View as plain text', 'Edit Build Information', and 'Delete Build'. The main area shows the build log for a job named 'Jenkins_project1'. The log starts with 'Started by user anonymous' and 'Building in workspace c:\work\Jenkins_project1'. It details the checkout process, including scanning the repository and local work area, and updating the file 'hello.c'. The build concludes with 'Finished: SUCCESS'. A cartoon character of a man in a suit is overlaid on the left side of the log.

```
Started by user anonymous
Building in workspace c:\work\Jenkins_project1
[DIMENSIONS] Running checkout on master...
[DIMENSIONS] Running build in 'c:\work\Jenkins_project1'...
[DIMENSIONS] Checking out a fresh workspace because this project has not been built before...
[DIMENSIONS] Removing 'file:/c:/work/Jenkins_project1/'...
[DIMENSIONS] Checking out project "QLARIUS:JENKINS1"...
[DIMENSIONS] Checking out directory '\'.
[DIMENSIONS] (Note: Dimensions command output was -

[DIMENSIONS] SUCCESS: Using Current Project 'QLARIUS:JENKINS1'.
[DIMENSIONS] Using 'c:\work\Jenkins_project1\' as the Project work area.
[DIMENSIONS] Scanning repository: less than 1 microsecond
[DIMENSIONS] Scanning local work area: 0.04 sec
[DIMENSIONS] Updated 'c:\work\Jenkins_project1\hello.c' using Item 'QLARIUS:0A18.A-SRC;1'
[DIMENSIONS] 1 fetch operation succeeded, 0 failed.
[DIMENSIONS] Getting files: 0.01 sec
[DIMENSIONS] Getting Project: 0.14 sec
[DIMENSIONS] Operation completed
[DIMENSIONS] Operation completed
[DIMENSIONS] )
[DIMENSIONS] Calculating change set for directory '\'.
[Jenkins_project1] $ cmd /c call "C:\Program Files\Serena\Dimensions 12.2\Common Tools\tomcat\6.0\temp\hu

c:\work\Jenkins_project1>cd c:\work\jenkins_project1


c:\work\Jenkins_project1>copy hello.c hello.exe
        1 file(s) copied.

c:\work\Jenkins_project1>exit 0
Finished: SUCCESS
```

Step 10 Extend the Dimensions Plugin to capture the built artefact and create a baseline

Click on **Back to Project** and then select **Configure**, scroll down and under **Build Environment** check the box to **Lock Dimensions project...**

Build Environment

Lock Dimensions project while the build is in progress 

Under **Post-build Actions** check the option to **Load any build artifacts to the Dimensions repository** and define the **Patterns to use** as below


Load any build artifacts into the Dimensions repository

Patterns to use

Pattern



Now check the next tab below entitled **Tag successful builds in Dimensions as a baseline** and then click on the **Advanced...** button. Define the options as they appear below

Tag successful builds in Dimensions as a baseline 

Owning part for baseline

Baseline Template

Baseline Type

Baseline Scope



(Note - The above options can be used to configure the type of baseline that is used to tag this build. You should only change these options if you are a knowledgeable Dimensions user).

Baseline Name Template



Now click on the Save button to update the configuration.

Test the changes by running the steps described in Step 9. The following output should now appear.

```
Started by user anonymous
Building in workspace c:\work\Jenkins_project1
[DIMENSIONS] Running checkout on master...
[DIMENSIONS] Running build in 'c:\work\Jenkins_project1'...
[DIMENSIONS] Removing 'file:/c:/work/Jenkins_project1/'...
[DIMENSIONS] Checking out project "QLARIUS:JENKINS1"...
[DIMENSIONS] Checking out directory '\'.
[DIMENSIONS] (Note: Dimensions command output was -

[DIMENSIONS] SUCCESS: Using Current Project 'QLARIUS:JENKINS1'.
[DIMENSIONS] Using 'c:\work\Jenkins_project1\' as the Project work area.
[DIMENSIONS] Scanning repository: less than 1 microsecond
[DIMENSIONS] Scanning local work area: less than 1 microsecond
[DIMENSIONS] Updated 'c:\work\Jenkins_project1\hello.c' using Item 'QLARIUS:0A18.A-SRC;1'
[DIMENSIONS] 1 fetch operation succeeded, 0 failed.
[DIMENSIONS] Getting files: 0.01 sec
[DIMENSIONS] Getting Project: 0.03 sec
[DIMENSIONS] Operation completed
[DIMENSIONS] Operation completed
[DIMENSIONS] )
[DIMENSIONS] Calculating change set for directory '\'.
[DIMENSIONS] Dimensions project was successfully locked
[Jenkins_project1] $ cmd /c call "C:\Program Files\Serena\Dimensions 12.2\Common Tools\tomcat\6.0\temp\hu

c:\work\Jenkins_project1>cd c:\work\jenkins_project1

c:\work\Jenkins_project1>copy hello.c hello.exe
1 file(s) copied.

c:\work\Jenkins_project1>exit 0
[DIMENSIONS] Dimensions project was successfully unlocked
[DIMENSIONS] Running checkin on master...
[DIMENSIONS] Running build in 'c:\work\Jenkins_project1'...
[DIMENSIONS] Scanning workspace for files to be saved into Dimensions...
[DIMENSIONS] Loading files into Dimensions project "QLARIUS:JENKINS1"...
[DIMENSIONS] Build artifacts were successfully loaded into Dimensions
[DIMENSIONS] (SUCCESS: Using Project 'QLARIUS:JENKINS1'.
[DIMENSIONS] Using 'c:\work\Jenkins_project1\' as the Project work area.

[DIMENSIONS] Scanning repository: less than 1 microsecond
[DIMENSIONS] Scanning local work area: less than 1 microsecond
[DIMENSIONS] Creating new item revision for 'c:\work\Jenkins_project1\hello.exe'
[DIMENSIONS] Item QLARIUS:HELLO EXE-0X96739457X3728X0.A-EXE;1 created
[DIMENSIONS] Preserved 'c:\work\Jenkins_project1\hello.exe' as Item "QLARIUS:HELLO EXE-0X96739457X3728X0.A
[DIMENSIONS] Updating files: 0.34 sec
[DIMENSIONS] Uploading Project: 0.36 sec
[DIMENSIONS] Operation completed
[DIMENSIONS] Operation completed)
[DIMENSIONS] Build was successfully tagged in Dimensions as a baseline
[DIMENSIONS] (SUCCESS: Baseline QLARIUS:JENKINS1_JENKINS PROJECT 1_6 has been forwarded to DMSYS
[DIMENSIONS] Operation completed
[DIMENSIONS] Operation completed)
Finished: SUCCESS
```

References

<http://wiki.Jenkins-ci.org/display/JENKINS/Dimensions+Plugin>