LabVIEW CI Service

The LabVIEW Continuous Integration Server and Project Builder require LabVIEW 2014 or later.

# Setting up the CI Server

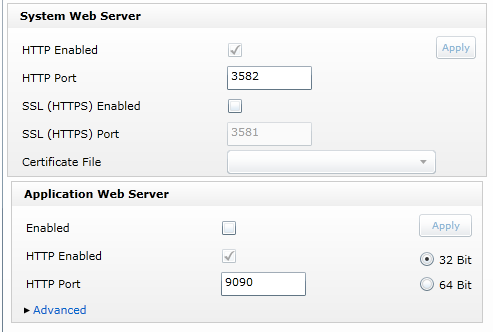
## Install LabVIEW

The CI Server needs a copy of the LabVIEW development environment to build applications. We assume you know how to do that.

However, you must change the LabVIEW Application Web Server port.

Jenkins and the LabVIEW Application Web Server use the same port. This is known to cause problems such as the Jenkins page not being able to load. To rectify this, the port of the LabVIEW Web Manager needs to be changed.

1. Open the LabVIEW Application Web Server in internet explorer window
2. Navigate to the “Web Server Configuration” Page
3. Under System Web Server and Application Web Server. Verify that the HTTP or HTTPS ports are **NOT** 8080. (They can be any other number not being used ex 9090)



**Important note:** The LabVIEW CI Server does **not** use the LabVIEW Application Web Server port. The only reason we are changing the Application Web Server’s port number is to get the server out of Jenkins’ way. The CI Server uses port 8002 instead (this is a hard-coded value). Any calls to the CI Server should use port 8002, and not the value you selected for the Application Web Service.

## Installing Jenkins

1. Navigate to <https://jenkins-ci.org/>
2. Download the “Latest and greatest” version.
3. Once downloaded run the installer

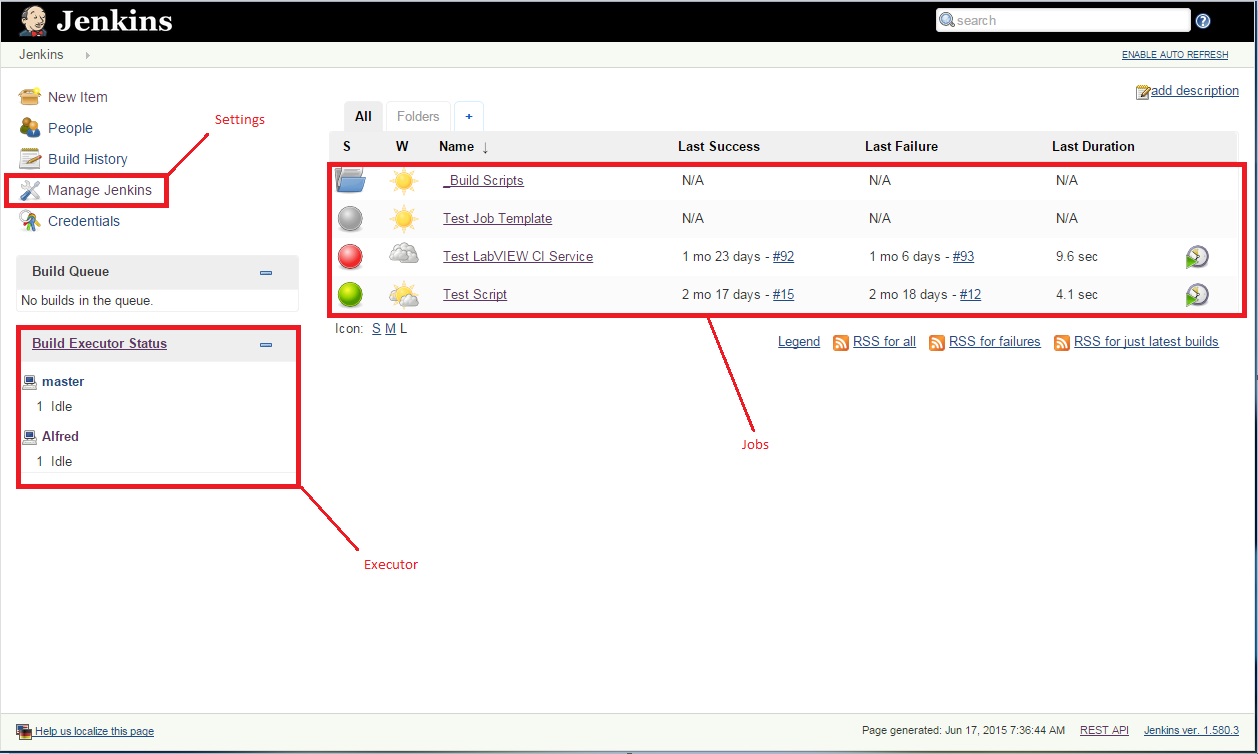
## Removing Read Only

1. Navigate to the Jenkins folder in Program files
2. Make Sure the folder is not set to Read Only and All users have Full Control over the folder.

## Launching Jenkins

1. Open Jenkins by using the default URL: <http://localhost:8080>

Note: The address can be changed in the “Manage Jenkins” Settings page

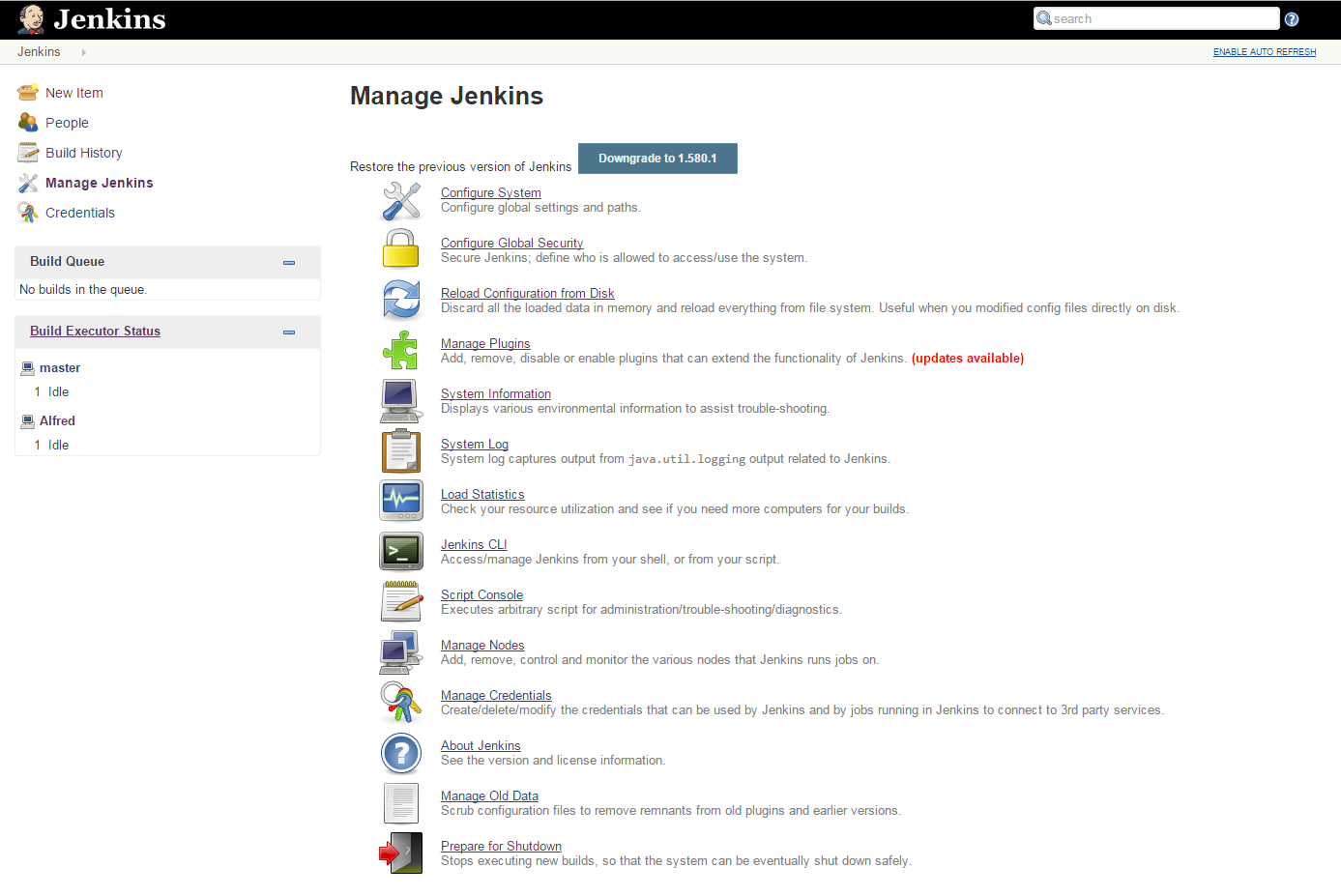


The “Jobs” section shows a list of all of the Jenkins jobs. If you use the Alfred tool, this section will show your jobs in folders.

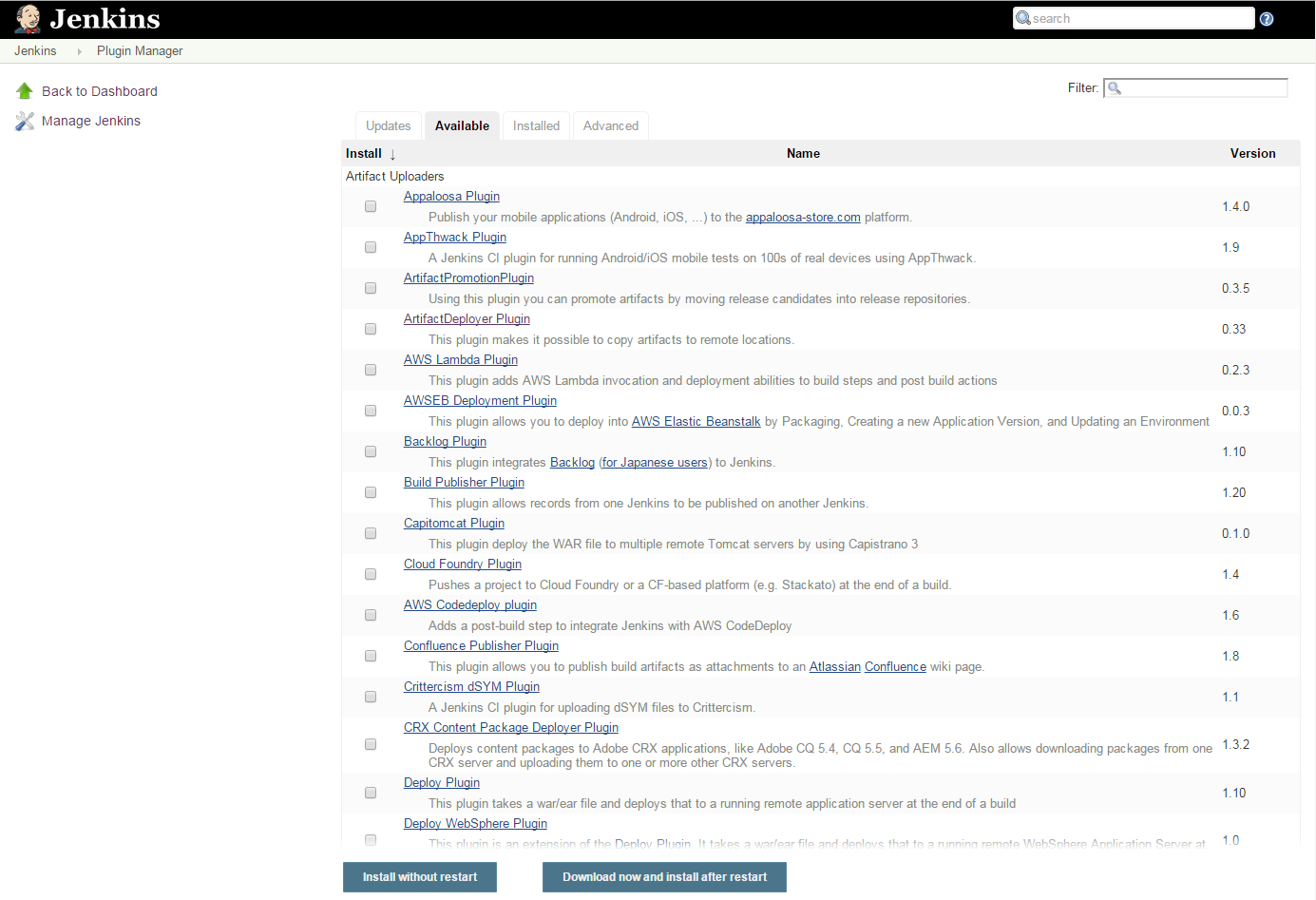
The “Executor” section shows the number of Executors and Nodes. A Jenkins job runs on a single executor within a node. Nodes can be on the same computer on or on a different computer. By default, there is one executor and one node, but this can be configured.

## Installing Plugins

1. Navigate to Manage Jenkins



1. Navigate to Manage Plugins



1. Click on the Available Tab
2. Search for and select:
   1. CloudBees Folders Plugin
   2. HTTP Request Plugin
   3. Job DSL Plugin
   4. Run Condition Extras Plugin
   5. Windows Slaves Plugin
   6. Workspace Cleanup Plugin
   7. (Whatever Source Code Control you use)
   8. Green Balls

Note: The Windows Slaves Plugin depends on one or more of these other plugins. You will not be able to find or select this plugin until you have installed the others, and restarted Jenkins.

1. Select “Download now and install after restart”

## Setting up Email Notification

1. Navigate to Manage Jenkins
2. Select Configure System
3. Scroll down to the Email Notification section and enter your company’s SMTP server and default user email suffix
4. Click Save

## Setting Up Slave Node for Alfred

Note: Setting this up is optional but highly recommended. Since the LabVIEW CI Project builder tool uses Jenkins scripts to build Jenkins jobs, these scripts run in Jenkins. If, you do not setup a slave node dedicated for Alfred, there may be a delay in the creation of jobs if another job is already running.

1. Follow these steps: <https://wiki.jenkins-ci.org/display/JENKINS/Step+by+step+guide+to+set+up+master+and+slave+machines>
2. Name the Slave node “Alfred”
3. After creating this node, make sure you have read and write access to the root folder you specified.

Note: It is recommended that you install the slave node as a Windows service. If you attempt to do so, and encounter an access error, stop the node and exit your browser. Run your browser again as an administrator, restart the slave node, and try again to run it as a service.

## Configuring Number of Executors

Change the number of executors to 1

Restart Jenkins before attempting to install the LabVIEW CI Server. Navigate to <http://localhost:8080/restart>.

# LabVIEW CI Installation

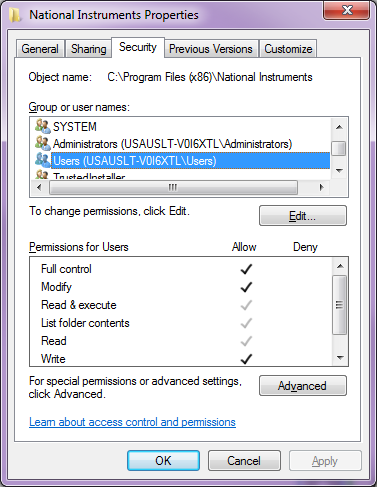
The LabVIEW CI Installation has two components. The LabVIEW CI Web Service runs on the build service machine, and processes build step requests from Jenkins. The Alfred tool is installed in your LabVIEW tools menu, and runs from your local development environment.

The components are distributed as VI Packages.

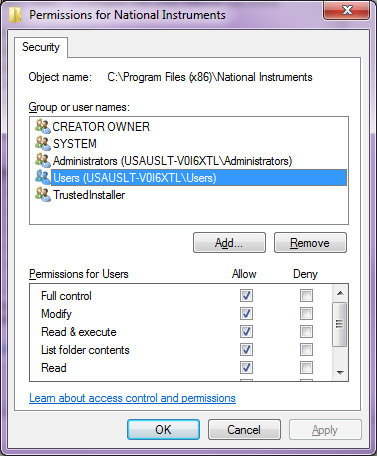
## Installing the LabVIEW CI Web Service

**Before you begin:** Windows’ security features can block the successful installation and building of your CI Server. We strongly recommend that, prior to installation, you give yourself full rights to your <applications folder>\National Instruments folder. (If this causes you concern, you may instead manually create the folder <applications folder>\National Instruments\LabVIEW Continuous Integration, and give yourself full rights to that folder instead.)

To do so, navigate to your applications folder, right-click on the National Instruments folder, and select **Properties.** Click on the Security tab.



Select Users, and inspect Permissions for Users. If you do not see checkmarks for “Full control”, “Modify”, and “Write”, as shown above, click the **Edit…** button.



Highlight Users, and click the appropriate check boxes, as shown above. Click OK.

Click OK again. You now have the necessary access rights to install the CI Server.

You will need to install two VI Packages on your CI server computer. Depending on how your server’s security is configured, it may be necessary to run VI Package Manager as an administrator.

Download and install the NI Launch Remote Actor package. You can find it here: <https://decibel.ni.com/content/message/46210#46210>

Install the package ni\_lib\_labview\_ci\_server\_source package.

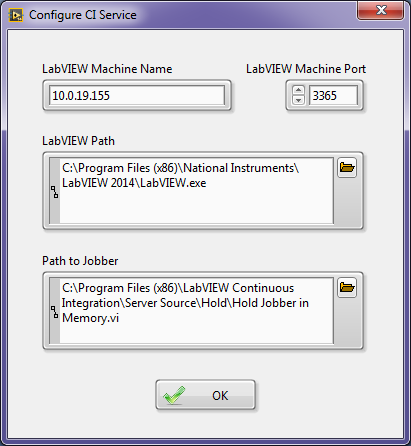
Once installed, navigate to the installation folder, which can be found at <applications folder>\National Instruments\LabVIEW Continuous Integration. Make sure you have read and write access to this folder.

Open this folder, navigate to the Server Source folder, and open LabVIEW Continuous Integration.lvproj.

Open and run Build Server.vi. This VI builds the various components of the CI Server.

Build Server.vi will automatically launch the installer for the CI Web Service. Follow the instructions in the installer dialog.

After the web service has been installed, you will be presented with a setup dialog, as shown.



Make sure the information shown is correct, and click OK.

You should have an icon on your desktop for the LabVIEW CI Web Service. Launch the service. Depending on your security configuration, you may need to launch the service as an administrator.

Restart Jenkins, as described above.

Your CI Server is now installed and ready, but you must define a Jenkins project to perform builds. This is done using the LabVIEW CI Project Builder.

## Installing and Using the LabVIEW CI Project Builder

The CI Project Builder is an addition to the tools menu of your LabVIEW development environment.

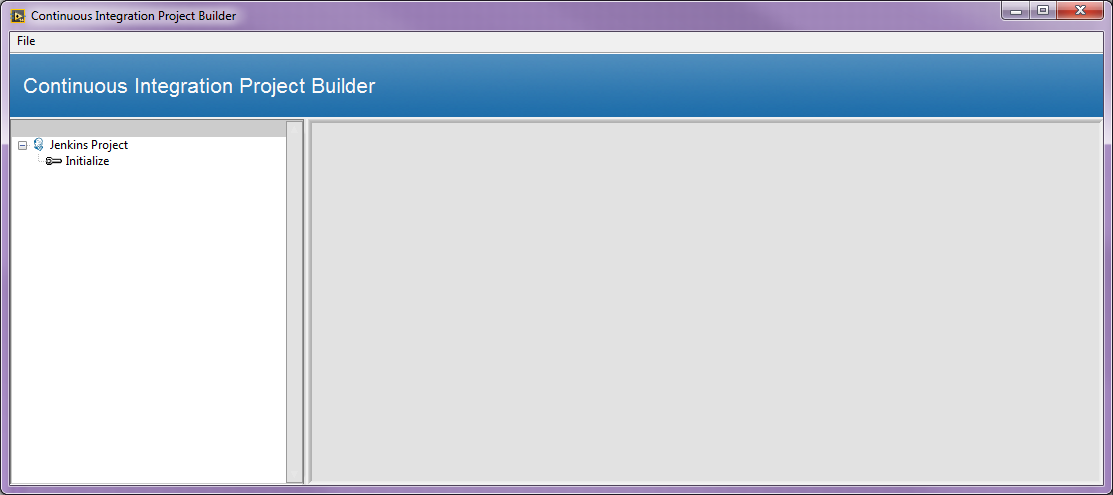
While you *can* install this tool on the CI Server’s copy of LabVIEW, it is intended to be installed and used on your developers’ machines.

The tool is distributed as a VI Package, and installs in the LabVIEW\Projects folder.

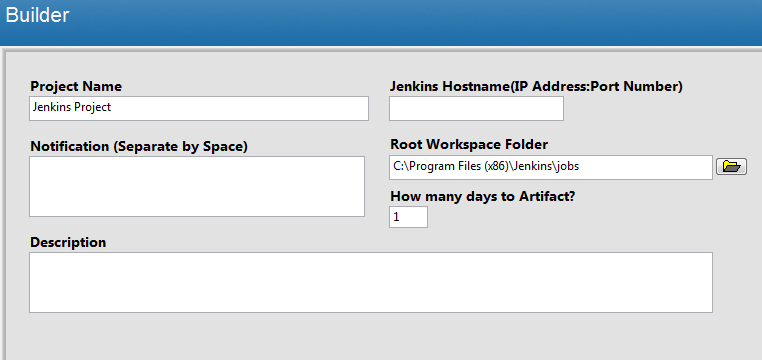
From VI Package Manager, install the package ni\_lib\_labview\_ci\_client\_source. The CI Project Builder is ready for immediate use.

To launch the CI Builder, select Tools » Continuous Integration Project Builder… from the LabVIEW menu.

You will be presented with the following dialog box:

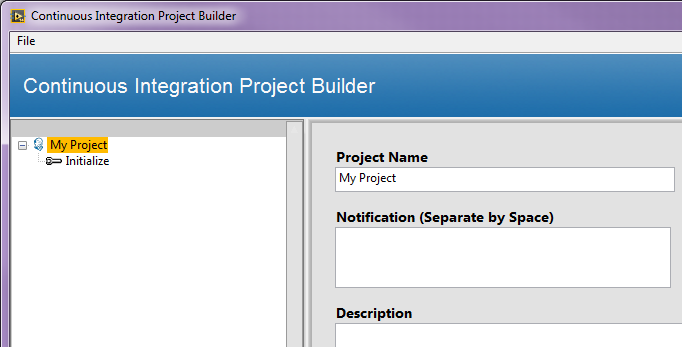


Click on “Jenkins Project.” The right pane will fill in with the following:



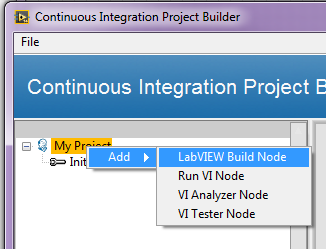
Fill in the indicated fields to define your CI project.

Updating the Project Name field will update the tree view to the left:

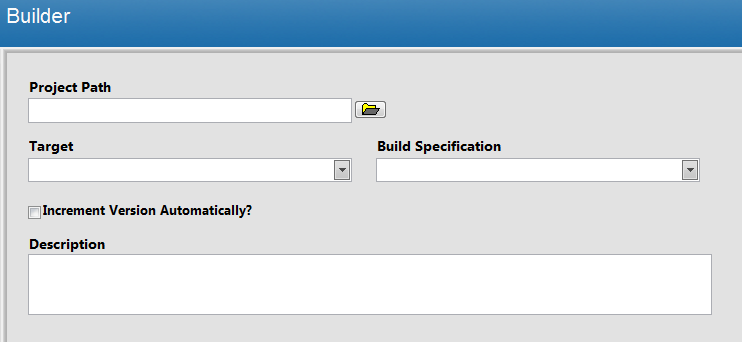


Important Note: The dialog box auto-populates the Jenkins Hostname field with your local IP address. Presumably, you will want to use a CI Server on a different machine. To do so, update this field with the IP address of your CI Server.

You can define as many build jobs as you wish. LabVIEW Build and Run VI ship with the CI Project Builder. Other build jobs are available on the Continuous Integration forum. To add steps, right click on your project’s name in the tree view, and select the step you wish to add.

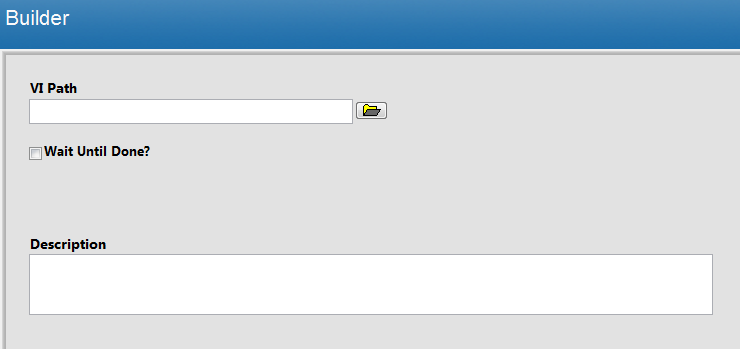


If you add a LabVIEW Build job and click on the job, you will see the following fields in the right hand pane:



Selecting a project path populates the Target pulldown menu. Selecting Target populates the Build Spec pulldown menu.

Adding a Run VI job presents these fields:



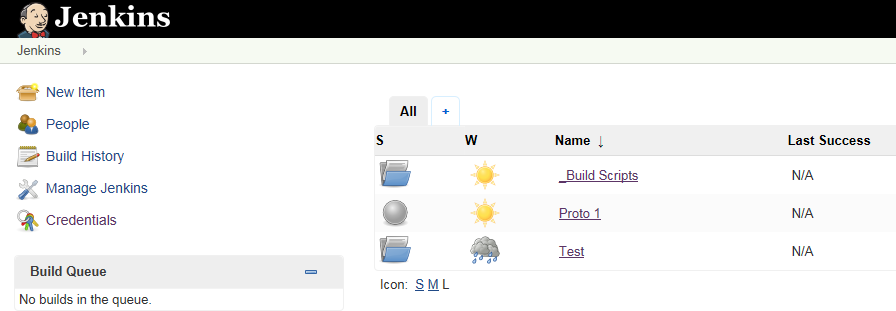
For both of these jobs, **the path field must point to** **the path where the project or VI resides on the CI Server**. For convenience, we strongly recommend that the local copy of your source code reside in the same folder structures on both the developer’s machine and the CI Server machine. Virtual drives are handy for this.

Save your configuration by clicking “Save.” You can have more than one saved configuration, and you can retrieve them by clicking “Load.”

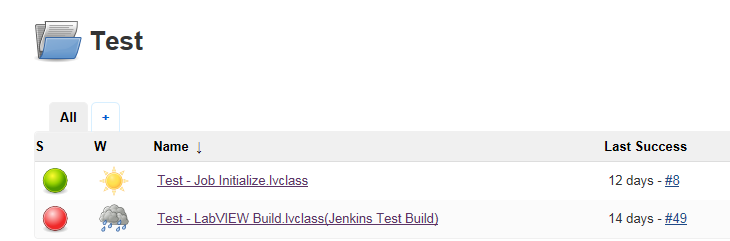
Click “Build” to configure this project in Jenkins.

## Configuring Source Code Control

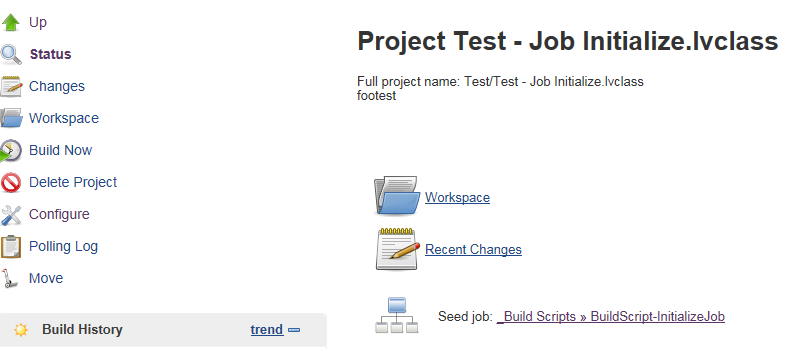
When the CI Project Builder tool builds your Jenkins project, it creates a folder on the Jenkins main page, as shown here.



In this illustration, the CI Project Builder created the “Test” folder. The new folder will contain all of the build steps you defined, plus a Job Initialize step, as shown here.



Job Initialize is the step that performs source code synchronization. You must customize this step in Jenkins to interact with your source code control system. Click on the step name to open the step, and then click on Configure.



A future release of the build tool may allow you to customize this step from within the tool.

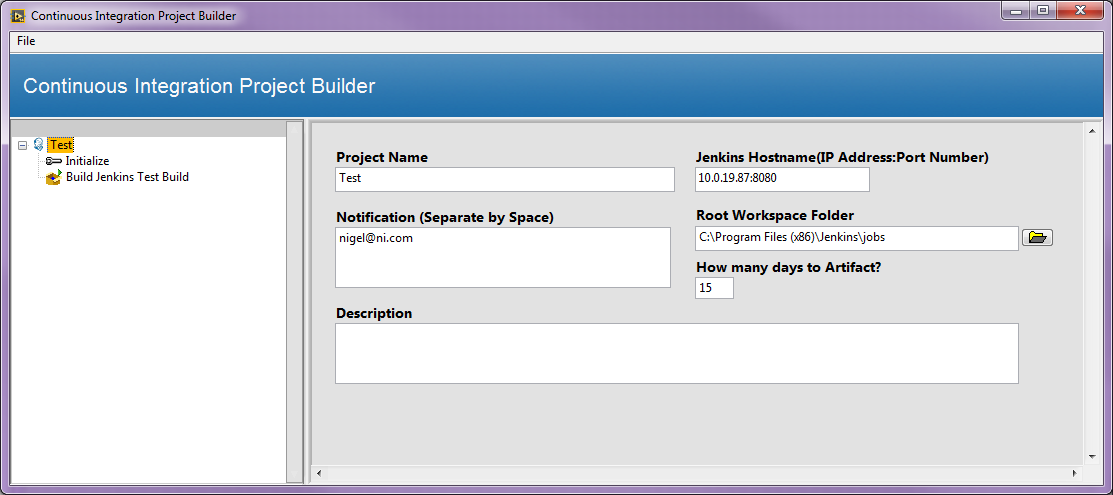
We intend to provide examples showing how to configure this step for common source code control tools at a future date.

## Example

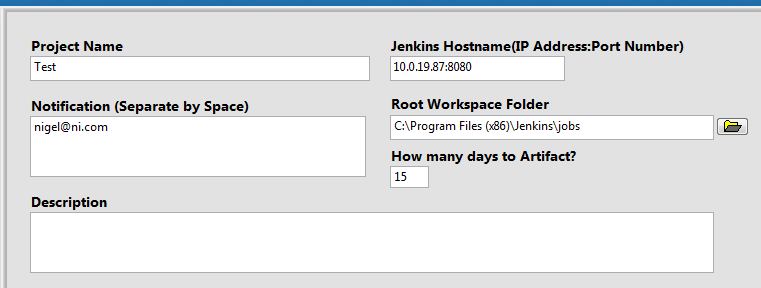
We have provided a sample project to test your installation. You can find it in the folder <applications>\National Instruments\LabVIEW Continuous Integration\Sever Source\Jenkins Sample Project.

The first step is to build the CI Project. Open LabVIEW, and select Tools » Continuous Integration Project Builder…

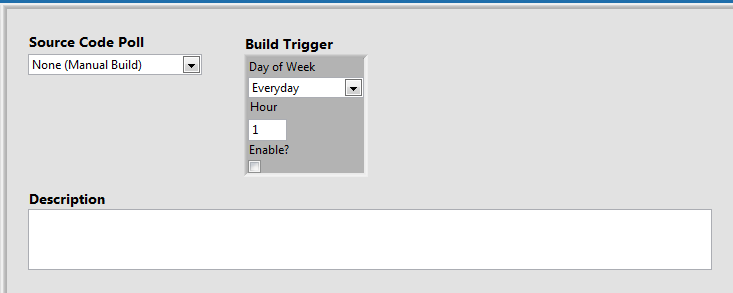
Complete the dialog box as shown, using the IP address of your Jenkins server.



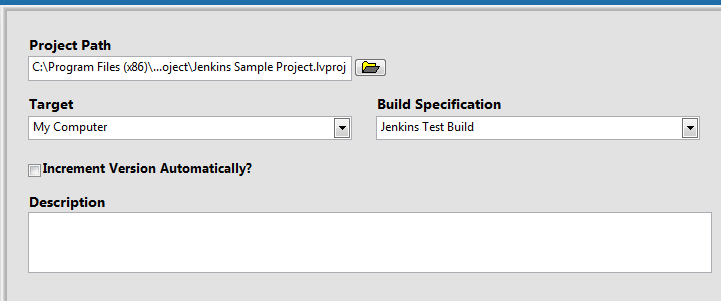
Here is a closeup of the main project page:



The Initialize page looks like this:



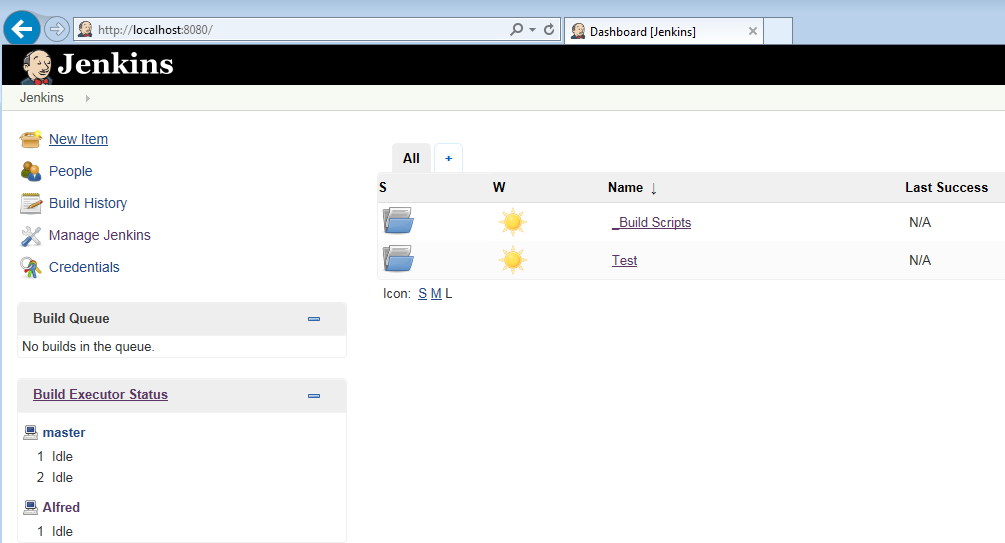
Finally, the LabVIEW Build page looks like this:



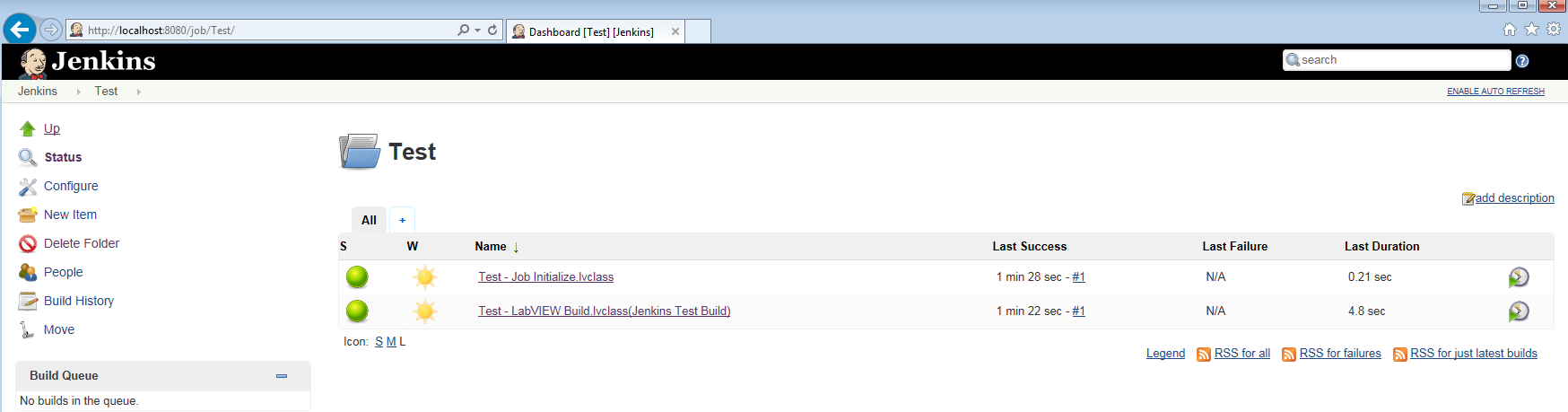
For the project path, select the Jenkins Sample Project. Be sure to use the path to the project located **on the CI server.**

Save and Build the CI project. Close this dialog box after the project has been built.

The Jenkins main page should now look like this.



“Test” is the project folder for the sample project. Click on “Test” to open the Test folder.

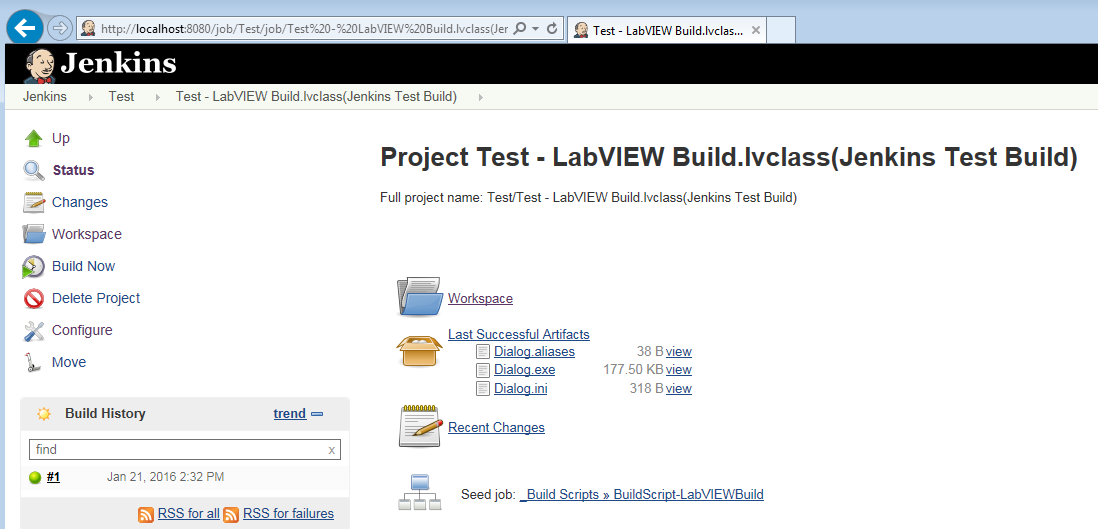


Click the run button to the far right of the “Test – Job Initialize.lvclass” jobs.

You should see the LabVIEW Build job appear in the queue, and then execute. LabVIEW should open and display the main panel for the LabVIEW Build job. After a short period, LabVIEW will exit, and the Build Executor status should be Idle.

Refresh the display to see the final status of these jobs.

To retrieve your artifacts, click on “Test – LabVIEW Build.lvclass (Jenkins Test Build).” The job page will contain links to your last successful build artifacts. Click to download the artifacts and test your results.



## Common Issues:

1. If you did not setup a slave node, go into the Build Scripts configuration and Uncheck “Restrict where this project can run

