



## Upgrade Instructions

Upgrading from one version of CI Factory to another can be a smooth affair. There are some things that you can do to help make it a simple task. This guide will walk you through the step to prepare for and upgrade as well as perform the upgrade itself.

### Preparation

There are some things that you can do to organize your CI Factory installation that will make an upgrade easier. Please try to isolate customizations out of the Packages, limit yourself to configuration changes in the properties files. This will keep manual merge work simple and to a minimum. Customizations are best kept in the file Main.Build.xml. An even better practice is to create a new file, such as Custom.Targets.xml, and include and call it from Main.Build.xml.

The upgrade script can over write many files in your installation. It is not a perfect automation and will rely on you to clean up a few odds and ends. To help you to know where to look for those odds and ends ask your source control repository for a list of files that you have changed since you first uploaded your CI Factory installation into the repository. You will want to review the changes that the upgrade script makes to these files. You may need to merge some changes from the version in the repository back after the upgrade script completes. Having this list of files will lessen your workload, otherwise you would have to review every file the upgrade script touches.

It is a good idea to stop the ccnet server for the upgrade target before running the upgrade script. This way there are no accidents.

Lastly find a machine beside the build server to perform the upgrade on. If you already have a working copy of the workspace for this project please delete it and pull it fresh from the repository. The goal here is to have only files that are in the repository on your machine. This will make it easy to know what files to add to the repository after the upgrade script has completed, anything not already in the repository. Please be aware that some source control clients need a force command argument in this situation.

If you are working with a source control system that requires files to be checked-out before editing them you should checkout the entire project, from the Current, or trunk, directory on down.

### Upgrade

Delete or rename your current version of CI Factory folder. If you have made any changes to CI Factory itself you will need your old copy to merge those changes into the new version of CI Factory. Run the CI Factory self-extracting executable to place the new version of CIFactory onto your system.



At this point, if you have made changes to CI Factory use a diff tool to merge those changes into the new version of CI Factory.

If you are not using the default folder name of "c:Projects" as you meta-base folder then you will need to edit the ..CIFactoryInstall Scriptsupgrade.xml build file and set the "ProjectsDirectory" property accordingly.

Execute the upgrade.bat script in the CI Factory folder. You will be asked what project you wish to upgrade. Please fill in the name and click the button. You will most likely be asked some questions for information that the build script can not figure out on it's own. Some of these include:

**Which codeline do you wish to upgrade?**

This will be asked if there is more than on directory in the project root directory. Choose the codeline, or branch, that you with to upgrade.

**What is the name of the Build directory?**

This will be asked if your build directory is named something other than Build. Please supply the name of the directory.

**What is the version or label prefix for this build project?**

**When CI Factory first installed your project it was set to '1.0.0.' in the file ccnetproject.xml.**

This is the first 3 numbers of the product version where the 4th is incremented with every successful build.

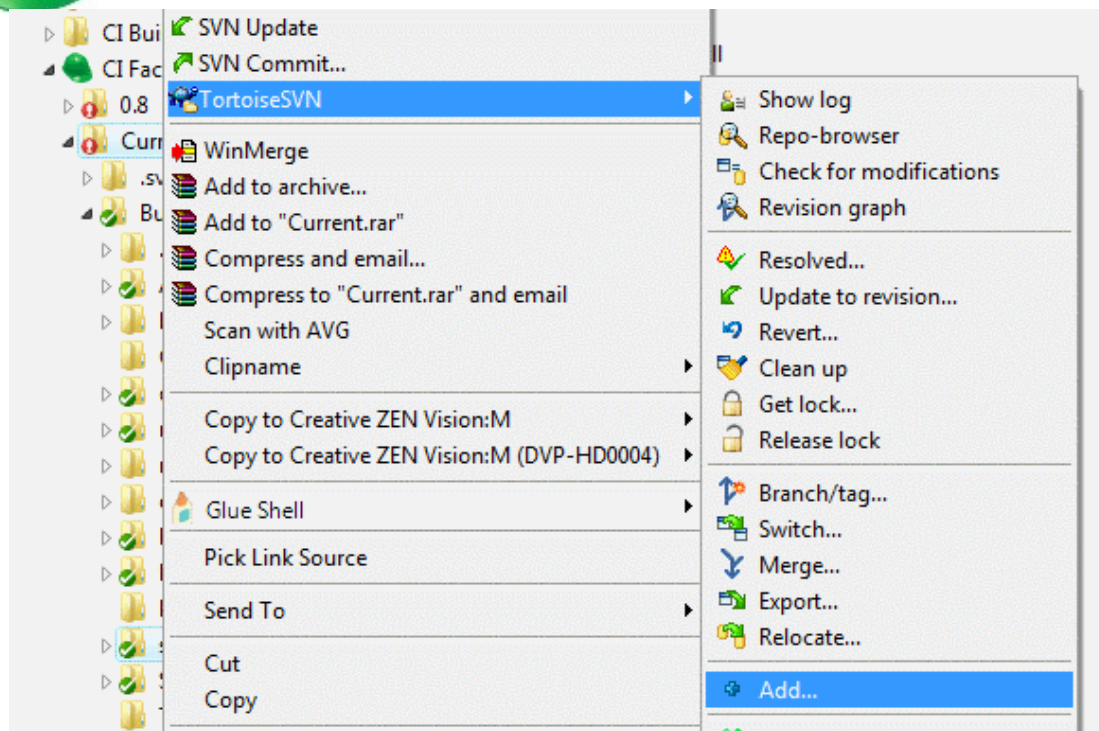
**I can't figure out what the port number is for the CCNet server. Please enter the port number.**

Well that is as simple as I know how to make it.

**I can't figure out what the host name for the CCNet server. Please enter the host name.**

This is the name of the machine that the build server runs on.

After the upgrade script has completed make sure to add all the new files to source control. All source control clients are different yet many allow you to execute their add command recursively on a folder already under version control. For example with TortoiseSVN:



After adding all the new files you may begin reviewing the changes made to the list of files you collected in your preparation.

Some source control clients don't offer a diff tool that allows you to merge (e.g. Perforce). I suggest you reconfigure your client to use the best diff/merge tool you can get your hands on; I prefer WinMerge myself.

For each of the files on your list execute the diff command against the latest version in the repository. Review the changes and make changes where needed.

After you have finished reviewing all the changes you are ready to commit/checkin the changes. Before committing the changes make sure that the ccnet server is not running.

After you finished committing the changes login to the build server and perform a manual update/get latest of the source. A manual update/get latest is required because ccnet can not update itself as it locks the assemblies.

Next you need to execute the NAnt script SetupIIS.xml. Open a command prompt and cd to the Build directory. Execute the following:

```
nAntbinNAnt.exe /f:SetupIIS.xml
```

Now navigate to the server folder in the Build directory. There are some files with the extension .state. The names of these files should be of the format <projectname>-<codelineame>.state.



Your codeline name is likely Current or trunk.

If the state file names are not formatted correctly correct them. You will also need to correct one value in the file as well. Please make sure to preserve the utf-16 encoding, edit with Visual Studio for example.

You are now ready to turn the ccnet server back on.

Done!