

CollabNet Connector to IBM Rational ClearCase Version 1.1

User Manual

COLLABNET®

Copyright © 2006 CollabNet, Inc. All rights reserved. CollabNet is a trademark or registered trademark of CollabNet, Inc., in the U.S. and other countries. All other trademarks, brand names, or product names belong to their respective holders.

CollabNet makes no representation with respect to accuracy or completeness of document, and specifically disclaim any implied warranties for any purpose and shall in no event be liable for any loss of profit or any other commercial damage, including but not limited to special, incidental, consequential, or other damages.



8000 Marina Boulevard, Suite 600, Brisbane, California 94005 U.S.A.
Toll free: 888-778-9793 • Voice: 650-228-2500 • Fax: 650-228-2501

www.collab.net • E-mail: info@collab.net

Table of Contents

Overview

Connector workflow	3
System Requirements	6
Setting up the Connector	6

Synchronizing data

Export mode	7
Re-export	9
Import mode	10
Best Practices	13
Editing the Configuration file	14
Frequently Asked Questions (FAQ)	15
Troubleshooting Guide	16

Overview

The CollabNet Connector to IBM Rational ClearCase is designed to help you work simultaneously between two diverse version control systems: **IBM Rational ClearCase** and **CollabNet Subversion**. The CollabNet Connector bridges these two version control systems.

The requirement for this kind of co-existence arises when an organization which is using IBM Rational ClearCase wants to synchronize data in its ClearCase Versioned Object Base (VOB) to a Subversion repository hosted in CollabNet Enterprise Edition, and vice-versa. In such a case, it is possible that one set of users use CollabNet Subversion for their distributed development and other set of users use ClearCase for version control, build and release management capabilities.

Therefore, changes made in either of the version control systems need to be synchronized with the other in order to have identical and up to date content, and avoid loss of data.

Connector workflow

The Connector synchronizes changes between the IBM Rational ClearCase views and the CollabNet Subversion working copy. The Connector provides two modes to synchronize data (files and documents, source code, images, etc.) between the two version control systems, namely *Export* and *Import*.

Before proceeding to a detailed explanation of the two modes of synchronization, you need to know the following terms used in this document:

- **ClearCase VOB:** This repository is opaque to the Connector, but is the core element of the Connector. Modifications made in a ClearCase view should be checked in to the ClearCase VOB for the Connector to run in the export mode.
- **ClearCase - Reference View:** For the export mode to run, we need two views one of which can be known as the *ClearCase - Reference View*. This view must match the CollabNet Subversion working copy for the Connector to run in the export mode.
- **ClearCase - Current View:** For the export mode to run, apart from the *ClearCase - Reference View*, we need another view that should contain the most recent changes made to the ClearCase VOB. This view can be known as the *ClearCase - Current View*.
- **CollabNet Subversion working copy:** This is a path in your system in which the contents of the CollabNet Subversion repository are checked out. For the Connector to work in the export mode, contents here should match the *ClearCase - Reference View*. In the import mode, the contents of the CollabNet Subversion working copy will synchronize with a ClearCase view.
- **CollabNet Subversion repository:** When you export the difference between the two ClearCase views, modifications made (*except for file content modifications*) are automatically committed to the CollabNet Subversion repository. Changes made in the CollabNet Subversion working copy should be committed to the CollabNet Subversion repository for the Connector to run in the import mode.

Export: In this mode, the Connector synchronizes the changes made in the ClearCase view to the CollabNet Subversion working copy.

For the Connector to work in this mode, it is mandatory to have two ClearCase views to clearly distinguish between modifications made. The Connector gets the difference between the *ClearCase - Reference View* and the *ClearCase - Current View*, and synchronizes it with the *CollabNet Subversion working copy*.

- If the *ClearCase - Current View* is updated with files or folders added, deleted or renamed, then the difference between the *ClearCase - Reference View* and *ClearCase - Current View* will be synchronized with the CollabNet Subversion working copy and also directly committed to the CollabNet Subversion repository.
- If the *ClearCase - Current View* is updated with file content modifications, the Connector will synchronize this change **ONLY** with the CollabNet Subversion working copy. You need to manually commit this change in the CollabNet Subversion working copy to the CollabNet Subversion repository.

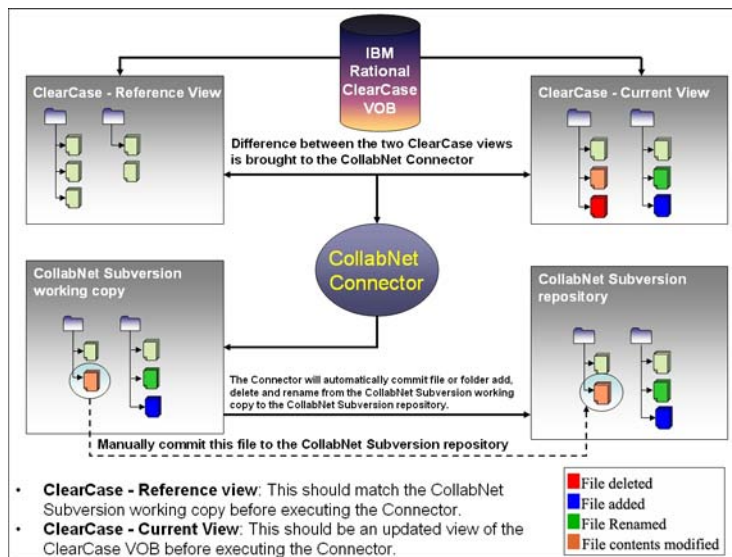


Figure1: The preceding illustration shows a *ClearCase VOB*, two ClearCase views: a *Reference View* and a *Current View*, a *CollabNet Subversion working copy* and a *CollabNet Subversion repository*.

Once you execute the Connector in the export mode, the following will happen (refer Figure 1):

- The Connector will detect the difference in data between the two ClearCase views.
- The file deleted (*red*), file added (*blue*) and file renamed (*green*) will be exported to the CollabNet Subversion working copy and also directly committed to the CollabNet Subversion repository.
- The file whose content has been modified (*Orange*) will be updated only to the CollabNet Subversion working copy. You need to manually commit it to the CollabNet Subversion repository to synchronize all data.

Import: In this mode, the Connector brings in the changes made in the CollabNet Subversion working copy to a ClearCase view of your choice.

Once the modifications are imported to the ClearCase view specified, you need to checkin the changes made in this view to the ClearCase VOB.

Note: Unlike the export mode, there is no auto-commit of data directly to the ClearCase VOB.

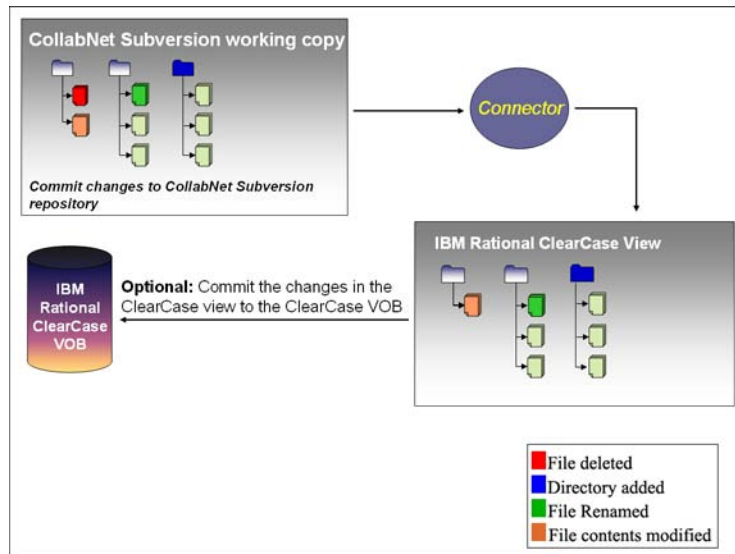


Figure 2: The preceding illustration shows a *ClearCase view*, a *CollabNet Subversion working copy* and a *ClearCase VOB*. Modifications made in the CollabNet Subversion working copy should be first committed to the *CollabNet Subversion repository*.

On executing the Connector in the import mode, the following will happen (refer Figure 2):

The file deleted (*red*), directory added (*blue*), file renamed (*green*) and file whose content is modified (*orange*) in the CollabNet Subversion working copy will be imported to the ClearCase view. You need to manually checkin the changes made in this view to the ClearCase VOB.

System Requirements

The Connector requires the following:

- Operating Systems supported:
 - Microsoft Windows XP with Service Pack 2
 - Red Hat Linux 7.2
- IBM Rational ClearCase version 2003.06.00
- CollabNet Enterprise Edition v. 4.0.0, v. 4.1.0, v. 4.1.5
- A Subversion command-line client for Windows:
 - **Subversion 1.2.1** (lowest version supported by this Connector)
To download, click: <http://subversion.tigris.org/files/documents/15/24487/svn-1.2.1-setup-2.exe>
(or)
 - **Subversion 1.3.0** (current version)
To download, click: <http://subversion.tigris.org/files/documents/15/29065/svn-1.3.0-setup.exe>
- Free disk space: 100 MB
- System memory (RAM): 256MB

Note: To install and configure this Connector on a Linux platform, contact the CollabNet Services Team. The functionality of the Connector is the same for the Linux platform, the only difference being the usage of lower case for filenames in Linux.

Setting up the Connector

To setup the Connector, unzip the '*ClearCase_versioning_Connector.zip*' file on your system and launch the '*Connector_Installer*' executable. The installer will scan your system and check if **Subversion v1.2.1** or above and **IBM Rational ClearCase 2003.06** are installed.

If the installer does not detect any one of the required command-line tools, an error message will be displayed, and you will be forced to quit the installer. If the required conditions are met, you can proceed with the installation process.

Note: If you are overwriting the contents of the ClearSVN Connector v1.0 with this version of the Connector, then ensure to take a backup of the configuration file (*clearsvn.ini*) if you have values specified for the import and export arguments. Once the Connector installer executes successfully, replace the new configuration installed with the configuration file you saved.

Some essential files installed (for example, under the *C:\Program Files\ClearSVN\dist* directory) include:

- **clearsvn.exe** - An executable file to run the Connector.
- **clearsvn.ini** - A user editable configuration file which can be used to set commonly used arguments with this Connector.
- **Documentation** folder - A folder that contains the User Manual for this Connector.

Synchronizing data

You can transfer data between *IBM Rational ClearCase* and the *CollabNet Subversion* version control systems using the *Export* and *Import* mode. This chapter discusses the procedure to execute the Connector in these two modes, arguments to be passed while executing the commands, and editing the configuration file.

Export mode

This mode transfers data modifications made in the IBM Rational ClearCase views to the CollabNet Subversion working copy.

For the export to work, you need to have:

- A ClearCase VOB. For example, let us term this VOB as '*Components*'.
- Two ClearCase views for this ClearCase VOB to clearly distinguish between the modifications:
 - **PC_view1** (*Reference View*): Let us term this view as **V1**. The contents in this view should match the CollabNet Subversion working copy before executing the Connector in the export mode. Let us assume that the data in this view is stored on your system in the following path: `C:\PC_view1\components\hardware` where:
 - *Components* is the ClearCase VOB name and
 - *hardware* is the directory name that needs to synchronize with the CollabNet Subversion working copy (*refer to the image on Page 8 to know the contents under this directory*)
 - **PC_view2** (*Current View*): Let us term this view as **V2**. Let us assume that the data in this view is stored on your system in the following path:
`C:\PC_view2\components\hardware`
- A *CollabNet Subversion working copy* in your system. For example, let us assume that the *trunk* folder of the CollabNet Subversion repository is been checked out in the following path: `C:\hardware`

There are numerous ways by which this Connector can be used to export data from the two ClearCase views to the CollabNet Subversion working copy. Let us consider one scenario:

STEP 1 From the command prompt, navigate to **V2**.

STEP 2 Update **V2** using the *cleartool update* command. Now, **V2** will be synchronized with the most recent changes made to the ClearCase VOB. For example, let us assume that **V2** has been updated with the following files (*refer to the image on Page 8*):

- File added - *WiFiSupport.pdf*
- File deleted - *clockrate.doc*
- File renamed - '*overview.html*' to '*QuickStart.html*'
- File contents modified - *FSB.xls*

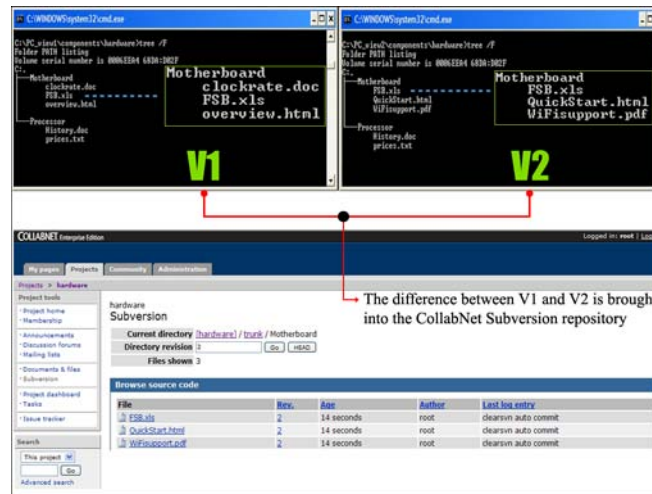
Important: DO NOT UPDATE V1. Before proceeding to Step 3, you need to ensure that **V1** and the *CollabNet Subversion working copy* match with the same content. Data that needs to be transferred has to be done from the root (parent folder), a branch or tag whichever is applicable.

STEP 3 Execute the following from the command prompt:

```
clearsvn export -b c:\PC_view1\components\hardware -c c:\PC_view2\components\hardware
-s c:\hardware
```

In other words:

```
clearsvn export -b <directory path of V1> -c <directory path of V2> -s <directory path of the
CollabNet Subversion working copy>
```



If the export is successful:

- The following files will be exported to the CollabNet Subversion working copy and automatically committed to the CollabNet Subversion repository:
 - WiFisupport.pdf* (new file added)
 - clockrate.doc* (file deleted)
 - QuickStart.html* (file renamed from 'overview.html')
- The 'FSB.xls' file whose content was modified will be synchronized only with the CollabNet Subversion working copy. You need to manually commit this file to the CollabNet Subversion repository.
- Two files suffixed with a timestamp are generated in the directory where the command was executed, for example:
 - svnexport.ini.2006-01-17 10-03-08** - This file contains the command-line arguments passed while executing the command.
 - svnexport.session.2006-01-17 10-03-08** - This file confirms that the export operation was successful.
- A log file is generated in the directory from where you execute the command, or in the path you specify using the **-l** argument. This file captures background operation information.

Once data in the *CollabNet Subversion working copy* and **V2** match exactly, you can update **V1**. This is a recommended step if you want to use the same two views in the next export operation.

Re-export

If you abort an export command or it terminates abruptly, the re-export mode can be used to complete the export operation. Data will not be synchronized with the CollabNet Subversion working copy or repository if the export is terminated or aborted.

If the export command terminates, two files without a timestamp are generated in the directory from where you execute the export command. For example:

- ***svnexport.ini*** - This file contains the command-line arguments given while executing the command.
- ***svnexport.session*** - This file will contain the status of the export command.

Once you execute the re-export command, the Connector picks up the required information from these two files and starts the session again.

Execute the following command to re-export the contents:

clearsvn export -r (or) **clearsvn export --re-export**

Note: If you prefer not to use re-export command and want to perform a fresh export, you can delete the two files (*svnexport.ini* and *svnexport.session*) created from the respective directory and run the export command again.

Table 1: Mandatory arguments to be passed while executing the Export command

Short option	Long option	Description
-b	--ccref-view-root=DIR	Here, DIR is the path of the <i>ClearCase - Reference View</i> . Modifications should not be made in this ClearCase view. This is done in order to get a difference between the <i>ClearCase - Current View</i> and the <i>ClearCase - Reference View</i> .
-c	--ccview-root=DIR	Here, DIR is the path of the <i>ClearCase - Current View</i> which has the most recent changes to the <i>ClearCase VOB</i> . Modifications made here should be synchronized with the CollabNet Subversion repository.
-s	--svn-client-root=DIR	Here, DIR is the path of the CollabNet Subversion working copy to which the Connector brings in changes from the ClearCase views.

Note: To list the arguments available with this Connector, execute one of these commands:

- ***clearsvn import -h*** (*-h* or *--help*): Help information on the import mode
- ***clearsvn export -h*** (*-h* or *--help*): Help information on the export mode

Import mode

This mode transfers data modifications made in the CollabNet Subversion working copy to the IBM Rational ClearCase view.

For the import mode to work, you need to have:

- A ClearCase view: Synchronization of data from the CollabNet Subversion working copy can be done to only one ClearCase view.
- A CollabNet Subversion working copy in your system. In the example used, it is `c:\hardware`.

Follow these steps to synchronize data between the CollabNet Subversion working copy and the ClearCase view:

STEP 1 Update the CollabNet Subversion working copy to ensure that it is synchronized with the CollabNet Subversion repository. In this case, it is `c:\hardware`.

STEP 2 In the CollabNet Subversion working copy, you can add, delete, rename files or directories and modify contents of files. For example, let us assume that the 'QuickStart.html' file is deleted from the `c:\hardware\Motherboard` folder.

STEP 3 Commit these changes to the CollabNet Subversion repository.

STEP 4 Update your CollabNet Subversion working copy once again before proceeding to Step 5. This ensures that the CollabNet Subversion working copy has the most recent changes made to the CollabNet Subversion repository and has no local modifications. For example, let us assume that the last revision number of the CollabNet Subversion repository is 10.

STEP 5 Execute the following from the command prompt:

```
clearsvn import -s c:\hardware\Motherboard -t \trunk\Motherboard -c
C:\PC_view1\components\hardware\Motherboard -r 5:10
```

In other words:

```
clearsvn import -s <CollabNet Subversion working copy directory path> -t <relative path from the
root of the CollabNet Subversion repository from which the CollabNet Subversion working copy is
checked out> -c <directory path of the ClearCase view> -r N:M (range of revisions)
```

When using the **-t** argument to synchronize changes, consider these tips:

- If you are synchronizing changes made in the *trunk* folder of the CollabNet Subversion working copy (in the example being used i.e. *c:\hardware*), then the **-t** argument in the command is:

```
clearsvn import -s c:\hardware -t \trunk -c c:\PC_view1\components\hardware -r 5:10
```

- If you are not sure which directory to synchronize, then the **-t** argument in the command is:

```
clearsvn import -s c:\hardware -t \ -c c:\PC_view1\components\hardware -r 5:10
```

This command will track all the changes made from the root (parent) folder and import them to the ClearCase view.

If the import is successful:

- The ClearCase view will contain the changes from the CollabNet Subversion working copy. Unlike the export mode, there is no auto-commit of files or folders directly to the ClearCase VOB in the import mode.
- All kinds of modifications (*file or directory add, delete, rename, copy-move and file content modifications*) made in the CollabNet Subversion working copy will be synchronized with the ClearCase view.
- A log file is generated in the default location of your system, or in the path you specify during the import command (**-l** argument). Unlike the export operation, no session files are created in the import mode.

You can check in the changes in the ClearCase view to the ClearCase VOB, to ensure that data is now synchronized.

Note: There is no re-import command to start a terminated or aborted import command. No unversioned files are created in the ClearCase view in case of an aborted or terminated session, and you can execute the import command again.

Table 2: Arguments to be passed while executing the Import command

Short option	Long option	Description	Mandatory / Optional
-c	--ccview-root=DIR	Here, DIR is the path of the ClearCase view where you want to synchronize changes.	Mandatory
-r	-r N:M	N and M are the revision numbers of the CollabNet Subversion repository. Changes made within this range inclusive of N and M will be synchronized to the ClearCase view. M should match the CollabNet Subversion working copy revision number. Note: N should be less than M. If you provide only a single value, then that revision number should be equal to the current CollabNet Subversion working copy revision. Refer to the <i>Troubleshooting Guide</i> section on page 16 for more on this option.	Mandatory
-s	--svn-client-root=DIR	This is the Subversion working copy directory from which the Connector consolidates the changes to be transferred to the ClearCase view.	Mandatory
-t	--tree-path=DIR	Here, DIR is the relative path from the root of the CollabNet Subversion repository from which the CollabNet Subversion working copy has been checked out.	Mandatory
-p	--post-import-scan=	This argument will check if the import command has been executed successfully and data synchronized from CollabNet Subversion to the ClearCase view. The two options to this argument are 'always' and 'onwarn'. If 'always' is specified, then the post-import-scan will be executed after import is complete. If 'onwarn' is specified, the scan will execute only if there are specific errors during the import command. Post import scans will be executed only if the -n argument option is not specified. Command: <code>clearsvn import <paths> -p onwarn</code> (or) <code>clearsvn import <paths> -p always</code>	Optional
-q	--quiet	In this mode, the Connector will operate in silent mode without the user's intervention.	Optional

Table 3: Optional arguments that are common to the Export and Import commands

Short option	Long option	Description
-f	<code>--force</code>	<p>This option forces the Connector to run, even if specific conditions are not met. For example, in the export mode, the Connector checks whether the <i>Reference View</i> and the <i>CollabNet Subversion working copy</i> match exactly. This force mode overrides the result of this check.</p> <p>In the import mode, the uppermost revision specified in the command line should match the CollabNet Subversion working copy revision. This option overrides this check in the import mode.</p>
-l	<code>--log-file=FILE</code>	By default, a log file is created in current directory from where you execute the Connector. By giving the <code>-l</code> argument, you can specify the path in your system or on the local network to store the log.
-n	n/a	This is the non-operational mode during import and export. When this option is specified, the Connector does not perform any modifications to the CollabNet Subversion working copy or ClearCase view, but displays what it would have done if this option had not been specified in the corresponding mode.
-v	<code>--verbosity=MODE</code>	<p>Depending on the option you set, the log file generated will contain a specific set of information on the background operations taking place when the Connector is executed. The three options with this argument are <i>'high'</i>, <i>'med'</i> and <i>'low'</i>. The default value for this argument is <i>'low'</i>.</p> <p>For example, if you set the option to <i>'high'</i>, the log file will contain in-depth information on each operation taking place, and if set to <i>'low'</i>, the log file will contain comparatively lesser information. It is not mandatory to provide a verbosity level everytime when executing the Connector. The connector will throw an error if any other value is specified for this option.</p>

Best Practices

Following are some best practices that you can follow for the Connector to operate smoothly:

- **Export:** Before executing the export command, you should have two ClearCase views. One view should be a *Reference View* where no modifications are made. The other view should be a *Current View* that should bring in the most recent changes made to the ClearCase VOB. Although you can make changes to the *Current View*, we recommend that this view to be an update of the ClearCase VOB.
- **Import:** Update your CollabNet Subversion working copy before executing the import command, as it will bring in the most recent changes made to the CollabNet Subversion repository.

Editing the Configuration file

The Connector installer provides a user-editable configuration file (**clearsvn.ini**) to bypass the method of passing arguments in the command-line. This configuration file has the necessary arguments in the form of fieldnames. By default, the values of these fieldnames are empty.

You can provide the necessary paths for the export and import modes. This configuration file is useful if you have a set of paths for each of the version control tools where you want to synchronize data.

Arguments in the import mode include the following:

- *ccview-root* = <This is the path of the ClearCase view you want to import the changes to>
- *svn-tree-path* = <This is the path equivalent to the root of the repository. This is the **-t** argument. For more information on this, refer to the table on [page 12](#)>
- *svn-range* = <This can be a single revision number or a range of revisions, for example 2:5>
- *quiet-mode* = <This mode can be enabled by setting the value to 1. When set to 1, the Connector will operate in silent mode without the user's intervention>
- *import-force-mode* = <This mode can be enabled by setting the value to 1. When set to 1, the Connector will proceed even if the uppermost revision in the synchronization range does not match the *CollabNet Subversion working copy* revision>
- *svn-client-root* = <This is the path of the CollabNet Subversion working copy. In the example used, it is *C:\hardware*>
- *verbosity-mode* = <You can set the value to *low*, *med*, or *high*. For more details on these options, refer to the table on [page 13](#)>
- *noop-mode* = <You can enable this mode by setting the value to *True*. If set to *True*, the Connector will check for file and directory changes made, but will not make any modifications to the ClearCase view>
- *post-import-scan* = <should be either '*onwarn*' or '*always*' if set>
- *log-file-name* = <This can be a path in your system where you want to store the log file, for example *C:\clearsvn\logs\abc.txt*>

Arguments in the export mode include the following:

- *ccview-root* = <This is the path of the *ClearCase - Current View*. In the example used, it is *C:\PC_view2\components\hardware*>
- *ccref-view-root* = <This is the path of the *ClearCase - Reference View*. In the example used, it is *C:\PC_view1\components\hardware*>
- *force-mode* = <You can enable this mode by setting the value to 1. When set to 1, the Connector will proceed even if the *ClearCase - Reference View* and the *CollabNet Subversion working copy* do not match>
- *svn-client-root* = <This is the path of the CollabNet Subversion working copy. In the example used, it is *C:\hardware*>
- *verbosity-mode* = <similar to the import option>

- *noop-mode* = <You can enable this mode by setting the value to *True*. If set to *True*, the Connector will check for file and directory changes made, but will not make any modifications to the CollabNet Subversion working copy>
- *log-file-name* = <similar to the import option>

Once you set the values for the respective fields, run the following command:

clearsvn export (or) **clearsvn import**

Note: If you assign values for the fields in the configuration file, but give a different value when you execute the command; the Connector will take the latter. For example, in the configuration file, if you set the value of the *verbosity-mode* field (import mode) to '*med*', and execute the import command with the **-v** argument value as '*high*', the Connector will take '*high*' as the value.

Note: For arguments whose value can be set to 'True' or '1', it does not necessarily mean that if their values are set to 'False' or '0', they will be disabled. In order to disable them, you can leave them blank.

Important: Only the values for the argument field can be edited, and not the field name itself.

Frequently Asked Questions (FAQ)

Q. *Will this Connector work with a GUI based Subversion client, for example Tortoise SVN?*

A. No. This Connector supports only the Subversion command-line client.

Q. *Will this Connector work outside a CollabNet Enterprise Edition environment?*

A. No, this Connector works alongside the Subversion tool in CollabNet Enterprise Edition and cannot be used with any other Subversion tool.

Q. *Will I encounter any authentication issues when executing the Connector?*

A. No, you will not face any authentication issues when executing the Connector. If you have set the Subversion authentication credentials to 'Yes' in the Subversion configuration file, the user-name and password will get cached.

The Subversion configuration file '*config*' is stored on your system in the following path:

C:\Documents and Settings\[Windows login name]\Application Data\Subversion

Q. *How do I revert back the changes if I have performed a wrong export or import operation?*

A. In case you have performed a wrong operation, you have to manually revert back the changes from the working copy or repository of the respective tool. For example, if you have synchronized data with a wrong directory:

- *Export* - Revert the changes made to the CollabNet Subversion working copy and repository.
- *Import* - Uncheckout the file or folder in the ClearCase view to which the modifications was made.

Q. *What will happen if a file or directory in a ClearCase - Current view has been moved prior to the export operation?*

A. This is considered a *Delete-Add* function where a file or directory is deleted from a certain location and added in another location.

Troubleshooting Guide

Problem: If I try to export modifications made in the *ClearCase - Current View* to the CollabNet Subversion working copy, the Connector displays a 'View private file' error. What does this error mean?

Solution: Prior to executing the export command, you should not make modifications in the *ClearCase - Current View* which contains the most recent version of the *ClearCase VOB*.

If you modify a file in this view and do not check it in, you will receive the following error: 'View private files found in <ClearCase - Current View directory path>. This ought not be the case. Export aborted!' Ideally, make sure to check in the changes made in this view to the ClearCase VOB.

Problem: I try to import changes made in the CollabNet Subversion working copy to the ClearCase view without performing an update of the ClearCase view. What will be the result?

Solution: During the import operation, it is not mandatory that the ClearCase view be an updated copy of the ClearCase VOB. The operation will be successful.

Problem: If I execute the export command without performing an update of the CollabNet Subversion working copy, I get an error. What does this mean?

Solution: The Connector will check if the *ClearCase - ReferenceView* matches the *CollabNet Subversion working copy*. If it does not match, an error will be displayed.

It will inform that you can use the force mode option (-f) to avoid checking if the *ClearCase - ReferenceView* matches the *CollabNet Subversion working copy*.

Problem: When I try to import a range of revisions from CollabNet Subversion to a ClearCase view, an error is displayed. Why?

Solution: If the upper range does not match the current CollabNet Subversion working copy revision, the Connector will display an error. For example, if the version of the CollabNet Subversion working copy is 10, and you want to import only revisions 3:7, the Connector will display an error. Here, you can use the force mode (-f) option to avoid the error.

Problem: When I run an export or import command, I get an 'IOError: [Errno 28] No space left on device' error. Why?

Solution: This error will appear if your system does not have sufficient disk space to transfer data. This error is not related to the free disk space required to install this Connector.