Data migration from Microsoft Visual SourceSafe to IBM Rational ClearCase and Unified Change Management

Vivek K Pandey

May 18, 2009

This article describes the procedure to migrate data kept in Microsoft® Visual SourceSafe® (VSS) repositories into IBM® Rational® ClearCase® repositories. It also shows you how to get the data in Unified Change Management (UCM) format, so that you can work with activities, streams, and the project structure.

This article describes how to migrate data (source code, documents, and such from Microsoft® Visual SourceSafe® (VSS) repositories into IBM® Rational® ClearCase® repositories, and from there to UCM format to work with activities, streams, and the project structure. Because you cannot import data directly into the Unified Change Management (UCM) versioned object base (VOB) structure, you must do the migration in three steps to get the data in UCM format:

1. Export the data from VSS repositories.
2. Import the data in ClearCase repositories in base ClearCase format.
3. Convert the base ClearCase directory and labels into UCM component directories and baselines.

However, before you go through these steps, it helps to understand how data is kept in VSS repositories. Also, you must determine what decisions you need to make before you can start to export data from VSS.

This article takes an example of an existing project structure in VSS, which was migrated to ClearCase and UCM in an actual project.

Visual SourceSafe directory structure

Data is kept in VSS under a single directory tree, and all of the project paths start with a forward slash (/). Therefore, the typical project structure in VSS would look like that shown in Figure 1. Rational_Sample_data is the product name, and it has different directories, which also include directories starting with the word "Branch." These directories are merely branched versions of the existing mainline directories.
Figure 1. Typical project structure

To elaborate, the Source directory is where mainline development takes place, and Branch_Source is the branch development of the Source directory. (This might be confusing if you are familiar with the ClearCase perspective, where you do not have separate physical directories for each branch.)

The Source directory and the Branch_Source directory may or may not contain different data, depending upon the work being done on both of the development lines. However, the Branch_Source directory will always exist from some version of the mainline development.

In this case, Branch_Source was created from Version 173 of the Source directory, as Figure 2 shows.

Figure 2. Sample development using branches

You can use this kind of branching to fix some bugs in the released version of the product, while working in parallel with the development of the next release of the software. In Figure 2, a label of Build.20 is applied to Version 173 of the Source directory, and a branch has been created to fix bugs on the Build.20-labeled version.

UCM directory structure

Before you export the data from the VSS, you need to decide upon the component (directory) structure that you ultimately want in UCM. If the directory structure has to be kept the same as it is in the VSS, then declare either the complete Source directory or each directory within the Source directory as individual UCM components.
There are certain advantages to having a proper component structure in place in UCM. The UCM user manual describes these advantages.

In this example, you declare the source directory as your UCM component (the next sections in this article show you how to do this). At this point, do not worry about the branched copy of the source directory, because you will also learn how to import it later in this article.

After you decide upon the directory structure in UCM, you can proceed to export the data from VSS. This can be imported in ClearCase format.

**Step 1. Export data from VSS**

Perform these steps to export data from VSS:

1. Open a command prompt and ensure that the PATH variable includes the path of your VSS installation directory.
2. Map the VSS repository to a network drive, using proper authentication.
3. Set the following environment variables:
   • set SSDIR=V:\ {mapped drive of the VSS repository}
   • set TMP=c:\temp
   • set SSUSER=<valid user name who has access to vss repository>
   • set SSPWD = <VSS password of above user-id>
   • Set the VSS project directory to this folder, which has to be exported: ss cp "/Article/DemoArticle1.0/Source"
   • Verify the current project using this command: ss cp
4. Export the VSS directories into a text file by using this command:
   
   clearexport_ssafe -r -o c:\datafiles\vss_export_Source.txt

   There are certain switches available with the clearexport_ssafe command, which you can use here according to the requirements:
   • -p <date-time>: Process only versions that have been modified since date-time was specified with new metadata.
   • -s <date-time>: Process only versions that have been modified since date-time was specified.
   • -l <date-time>: Process Important versions only, but include versions created since that specified time. A version is important only if:
     • It is the most recent version
     • It has a label
5. To keep the same branching structure from VSS to ClearCase (as depicted in Figure 1), use the following command:

   clearexport_ssafe -r -s <date-time> -b Branch_Source -v Build.20 -o c:\datafiles\vssexport_branch.

   Where:
   • Branch_Source is the Target branch in ClearCase
   • Build.20 is the Label from which the Branch_Source branch has to be taken out
• <date-time> is the Time when the Branch_Source branch was created in VSS, so that it processes only the required version on the branch
6. If there are no errors, the output file should have the export data after successful completion of the export command.

Step 2. Import data into ClearCase

In this step, you will import the data into ClearCase repositories with the help of the output data file that you created in Step 1.

Note: If you are not using UCM as your target ClearCase setup, then this may be the final step of your migration process. However, if this import is for a UCM setup, then you must complete Step 3, as well. Also, the clearimport command can not be used in UCM view context, so the import must be performed only in base clearcase view context.

Now you can start the import process:

1. Create the base ClearCase VOB cc_Migration_VOB (if it does not exist already).
2. Create the base ClearCase view cc_Migration_View (if it does not exist already).
3. Open the command prompt.
4. Set the view context created in Step 2:
   ```
cleartool startview ClearCase_Migration_View
   net use z:\ \view\ClearCase_Migration_View
   ```
5. Browse through the VOB until you find the target location, and then execute the following command:
   ```
cd ClearCase_Migration_VOB
clearimport c:\datafiles\vss_export_Source.txt
   ```
6. The import output should be similar to that in Listing 1.

Listing 1. Sample output of the clearimport command

```
Validating label types.
Validating directories and symbolic links.
Validating elements.
Creating element "\bugfix/add.sql".
   version "1"
   version "2"
   version "3"
Creating element "\bugfix/mod_empl.c".
   version "1"
   version "2"
   version "3"
   version "4"
   ...
Closing directories.
```

After you complete this import step, your data should be in the VOB that you created in Step 1 (cc_Migration_VOB), which you can access hereafter by using the valid ClearCase views. (Edit the config-spec of the view if you have imported it on the branch.)
7. (Optional) You can also label the imported version to identify the base version for further development of your source code. To apply a label to all of the imported source code using the above view, execute the following command:

```
cleartool mklbtype -c "Identify imported versions from VSS" VSS_to_CC_Migrated
```

8. Use the Apply Label wizard to apply the "VSS_to_CC_Migrated" label to all of the imported elements. **Important:** Make sure that you lock the label to prevent moving labels to any other version.

If you are interested in setting up the UCM use model for your configuration management, proceed to the next section.

**Step 3. Convert the base ClearCase directory and labels into UCM component directories and baselines**

UCM introduces the concept of two kinds of VOBs: the Project VOB and the Component VOB.

1. Create the UCM Project VOB. (Follow instructions in the ClearCase user manual for creating a Project VOB.)
2. Next, use the imported data from the previous section, according to UCM.
3. Import the VOB directory (created in Step 2 here as a result of the import) as a UCM component. Right-click **Components** and select **Import > VOB Directory as Component**, as shown in Figure 3.

![Figure 3. Import the VOB directory](image)

4. Specify the component details as follows (also see Figure 4):
   - **View:** test_view_vivek
   - **VOB:** BASE_CC_VOB
   - **Root Directory:** Calculator
   - **Component Name:** Calculator
In this instance, `test_view_vivek` is the base ClearCase view by which the component directory is visible. `Calculator` is the component directory.

5. Import the label created in Step 2 as a UCM baseline for components created in the previous Step 4. Right-click `Component` and select `Import > Label as Baseline`, as shown in Figure 5.

6. Given the new baseline, select the components that use the label that you want to import, as shown in Figure 6.
7. Specify the label that you created in Step 2, which is a base ClearCase label that has to be converted to a UCM baseline, as shown next in Figure 7.

**Figure 7. Import Label – Step 2 of 2**

8. Create a UCM project named `Migration_PVOB` in the previously created Project VOB (you can find the steps for creating the UCM project in the ClearCase user manual.) The integration stream for the project should be named `Migration_Project_Int`.

9. To start, right-click `Migration_PVOB` and select **New > Project**, as shown in Figure 8.

**Figure 8. Create a new project**

10. Add the component baselines (which you just imported), as shown in Figure 9.
11. Create the integration view for the project named Migration_Project, which points at the Integration stream named Migration_Project_Int.

If you start working with ClearCase, the integration stream going forward will be created from the baseline (created as a result of importing the label), as shown in Figure 10.

Figure 10. The “ClearCase Version Tree Browser” view

You are now ready to use your migrated source code with Rational ClearCase.
Future migrations

You can use the information in this article to help plan a migration from VSS to ClearCase (and possibly from ClearCase to UCM). Before the actual migration occurs, clearly define what will be migrated from VSS and in what shape (that is, which baseline and codes from VSS will be put into ClearCase and what stream structure and component structure you will use henceforth).

© Copyright IBM Corporation 2009
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)