opentext™

Dimensions CM

Software version: 14.7

CM Bridge Guide

Product version: 14.7

Last updated: December 8, 2023

Table of Contents

Chapter 1
Introduction

About OpenText™ Dimensions CM Bridge

Dimensions CM Bridge is a connector between a Subversion-enabled client and a Dimensions CM server that enables:

- Your development tools to talk to a CM repository instead of Subversion.

- Development teams familiar with Subversion to continue using their existing IDEs and tools with the additional benefits of Dimensions CM.

- Dimensions CM to be used with IDEs that do not have a "native" integration with CM but do support Subversion.

For details about supported versions, see the Support Matrix in the <u>Dimensions CM online help</u>.

NOTE CM Bridge is automatically installed with a Dimensions CM server.

Dimensions CM Repository URL

A Dimensions CM repository is not structured as a single file and folder tree like Subversion, instead each stream has its own files and folders. When working with a Dimensions CM stream from a Subversion client you must specify the Dimensions CM *product* and *stream* names in uppercase as part of the URL. For example:

- URL that points to the root of the QLARIUS product:

  ```
  http(s)://<web server>:8080/cmbridge/QLARIUS/
  ```

- URL that points to the root folder of the JAVA_BRANCHA_STR stream in the QLARIUS product:

  ```
  http(s)://<web server>:8080/cmbridge/QLARIUS/JAVA_BRANCHA_STR
  ```

- URL that points to a folder and file in the JAVA_BRANCHA_STR stream:

  ```
  http(s)://<web server>:8080/cmbridge/QLARIUS/JAVA_BRANCHA_STR/
      Qlarius Underwriter/qlarius/interfaces/AutoQuote.java
  ```

  TIP Products and streams are treated as directories in a repository.

- In the example below there are two products, *APPS* and *QLARIUS*, and the streams are listed as child folders of the products. There are multiple folders and files under JAVA_BRANCHA_STR.

NOTE You can use a web browser, SVN-based clients, or the SVN command line to view the contents of a repository.
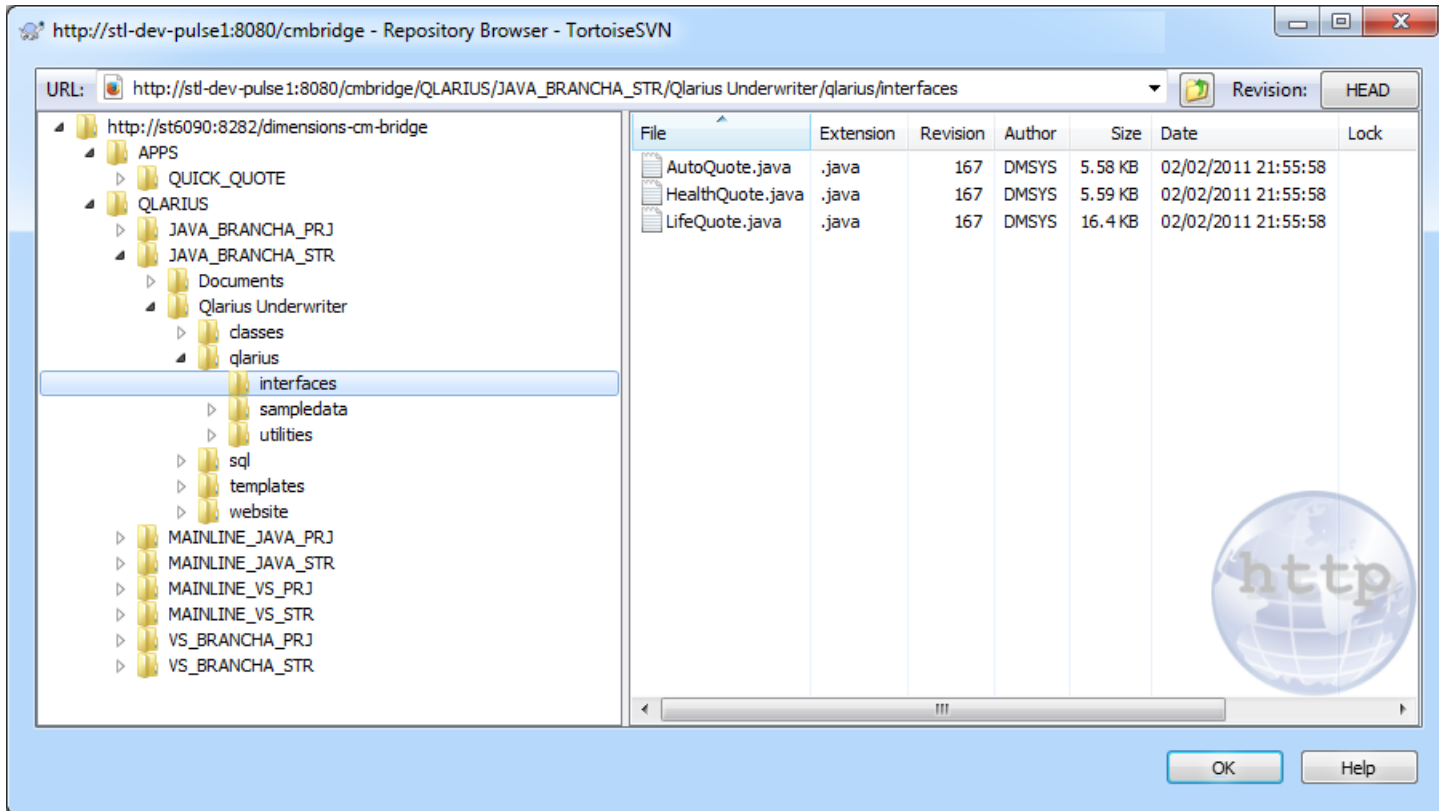
Specifying Multiple Base Databases

You can use multiple base databases with CM Bridge:

1 Open the `startup.properties` file located in:

- Windows: `C:\ProgramData\OpenText\Dimensions CM\Bridge\conf`

- Linux: `/opt/opentext/dimensions/<version>/bridge_data/conf/`

2 Add this parameter: `scm.multirepo=true`

3 Specify multiple base databases in the format `scm.url.<prefix>`. For example:

- `scm.url.typical=scm\:dimensions\://sun/cm_typical@sun14`

- `scm.url.mybasedb=scm\:dimensions\://sun/mybasedb@sun14`

4 Save the properties file.

5 Add the prefix to the CM repository URL:

`http://<host>:8080/cmbridge/`**`<prefix>`**`/<product>/<stream>`

For example:

`http://sun:8080/cmbridge/`**`cm_typical`**`/QLARIUS/JAVA_BRANCHA_STR`

Support for Subversion Properties

CM Bridge supports the storage and retrieval of Subversion properties. The following standard Subversion operations are supported:

- `propset`
- `propget`
- `propedit`

For details about using SVN externals with CM Bridge, see .

Performance Improvement Tips

To improve performance, limit the list of streams that is displayed to your favorite and recent streams.

1   Open the `startup.properties` file located in:
    - **Windows:** `C:\ProgramData\OpenText\Dimensions CM\Bridge\conf`
    - **Linux:** `/opt/opentext/dimensions/<version>/bridge_data/conf/`

2   Add this parameter: `streams.favrecent.filter=true`

    TIP: To specify the number of recent streams that is displayed, add this parameter:

    `streams.favrecent.recentcount=n`

Limitations and Known Issues

- Operations are only supported with Dimensions CM streams (not projects).
- The log output is empty for a file or folder that has not changed since a stream was created.
- When files are checked-out or exported, item header substitution is not performed.
- Dimensions CM directory items are not supported.
- All Subversion commands are supported except:
    - `merge`
    - `mergeinfo`
    - `copy` (you can only copy streams and projects in a single product).
    - `switch`
- SVN tagging is not supported.
- SVN branches are only supported at the product level.

Troubleshooting

When investigating issues with CM Bridge, first look in the log file:

`C:\ProgramData\OpenText\Dimensions CM\Bridge\_logs`

This log may include an error message or stack trace that can help Support to identify the issue.

TIP

- If you are having an issue in a particular client, try reproducing the problem from a Subversion command line as this can display a more detailed error message.

- Include logs, screenshots, error message text, and the steps you performed to reproduce a problem when you are reporting an issue to Support.

Chapter 2
Apple Xcode

Introduction

The following examples describe some of the ways that you can use CM Bridge with Apple Xcode.

Add an Xcode Project to Dimensions CM

You can add an existing Apple Xcode project to a stream in a Dimensions CM repository.

1 From a Terminal window change directory to the folder where the Xcode project is located.

2 Use the Subversion command line to import the folder into Dimensions CM. For example:

```
svn import Example http://<web server>:8080/cmbridge/QLARIUS/XCODE/
Example -m "Initial commit" --username <user ID> --password
<password>
```

Check Out an Xcode Project from Dimensions CM

You can check out an Xcode project from Dimensions CM.

1 Open Xcode.

2 Do one of the following:

- From the Welcome screen select **Check out an existing project**.

- From the Source control menu select **Check Out**.

The Check Out dialog box appears.

3 Do one of the following:

- Select a previously used repository.

- Enter a URL that identifies a Dimensions CM server and product in this format:

```
http://<web server>:8080/cmbridge/<CM product>
```

For example:

```
http://dev-pulse:8080/cmbridge/QLARIUS/
```

4 Click **Next**.

5 From the repository **Type** list select Subversion.

6 If prompted enter a CM user ID and password and click **Next**.

7 Open a CM product.

8 Select the project or stream to be checked out and click **Next**.

9 Select the local directory to which to check out the project or stream.

10 Click **Check Out.** The project or stream is checked out from Dimensions CM to the local directory.

Commit Changes in an Xcode Project to Dimensions CM

You can commit changes that you make in an Xcode project to a stream or project in Dimensions CM. You can also relate commits to Dimensions CM change requests.

1 From the **Source Control** menu select **Commit**.

2 In the **Messages** box:

- Enter a commit comment for your changes.

- Optionally enter the IDs of one or more Dimensions CM change requests as part of the comment in this format: [CR1,CR2]. For example:

    [QLARIUS_CR_21,QLARIUS_CR_36] Your comment

3 Click **Commit**.

Update an Xcode Project

You can update the local version of an Xcode project with changes committed by other developers in your team. From the **Source Control** menu, select **Update**.

Chapter 3
TortoiseSVN

Introduction

CM Bridge supports the majority of TortoiseSVN operations. The following examples describe some of the ways that you can use CM Bridge with TortoiseSVN.

Browse Files and Folders

You can use TortoiseSVN to browse the folder structure of a stream and view the contents of files.

1 From Windows Explorer open the TortoiseSVN repository browser.

2 Enter a URL to a Dimensions CM stream, for example:

```
http://<web server>:8080/cmbridge/QLARIUS/JAVA_BRANCHA_STR
```

3 If prompted, enter a CM user ID and password. The structure of the repository is displayed in the repository browser, and the stream that you specified is expanded:
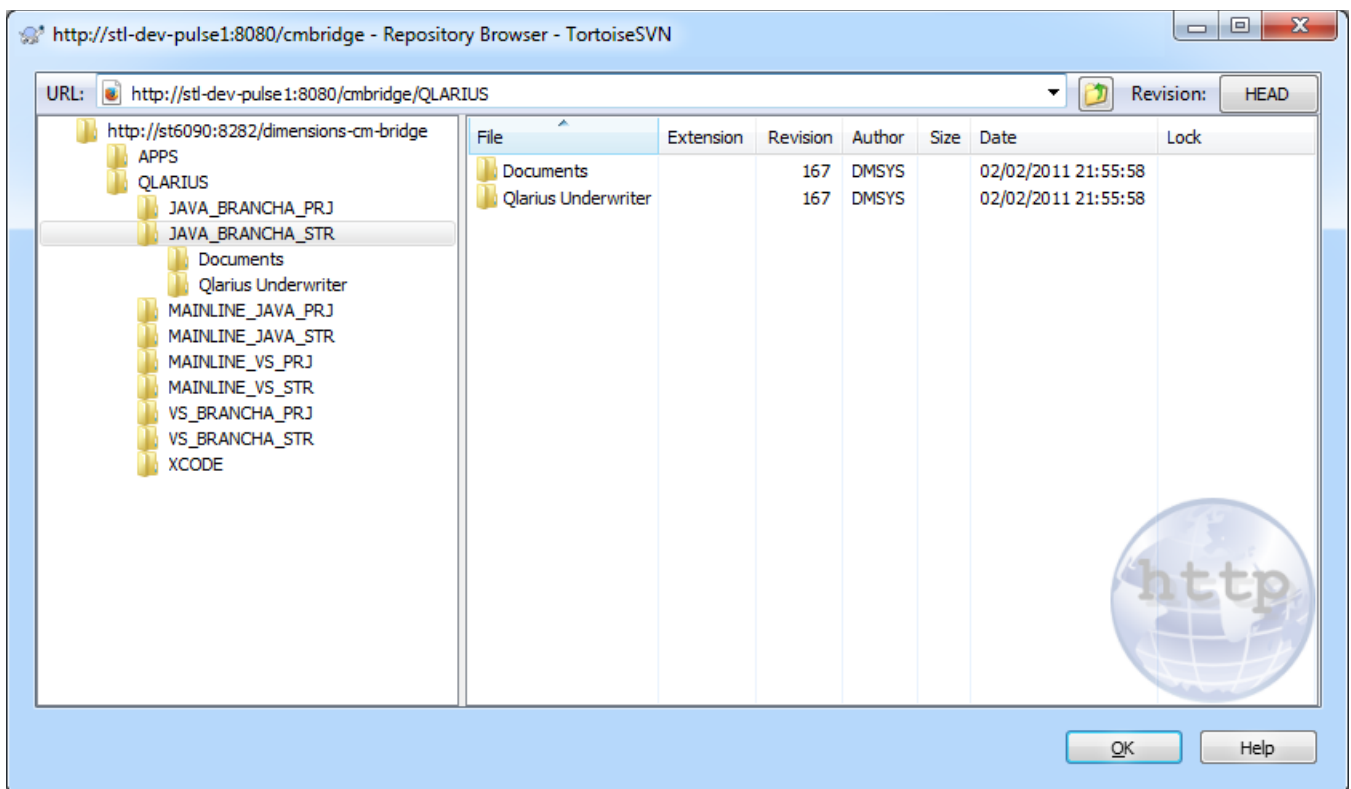
4 To browse a folder's contents, select it.

5 To browse a file, right-click it and select **Open**. The file is downloaded from Dimensions CM and opened in the default application for the file type.

View Log Messages

You can use TortoiseSVN to view the history of any object and see what changes were delivered to it.

Right-click a folder and select **Show Log**. The details of all the changesets delivered to that folder are displayed including the files modified in each commit.
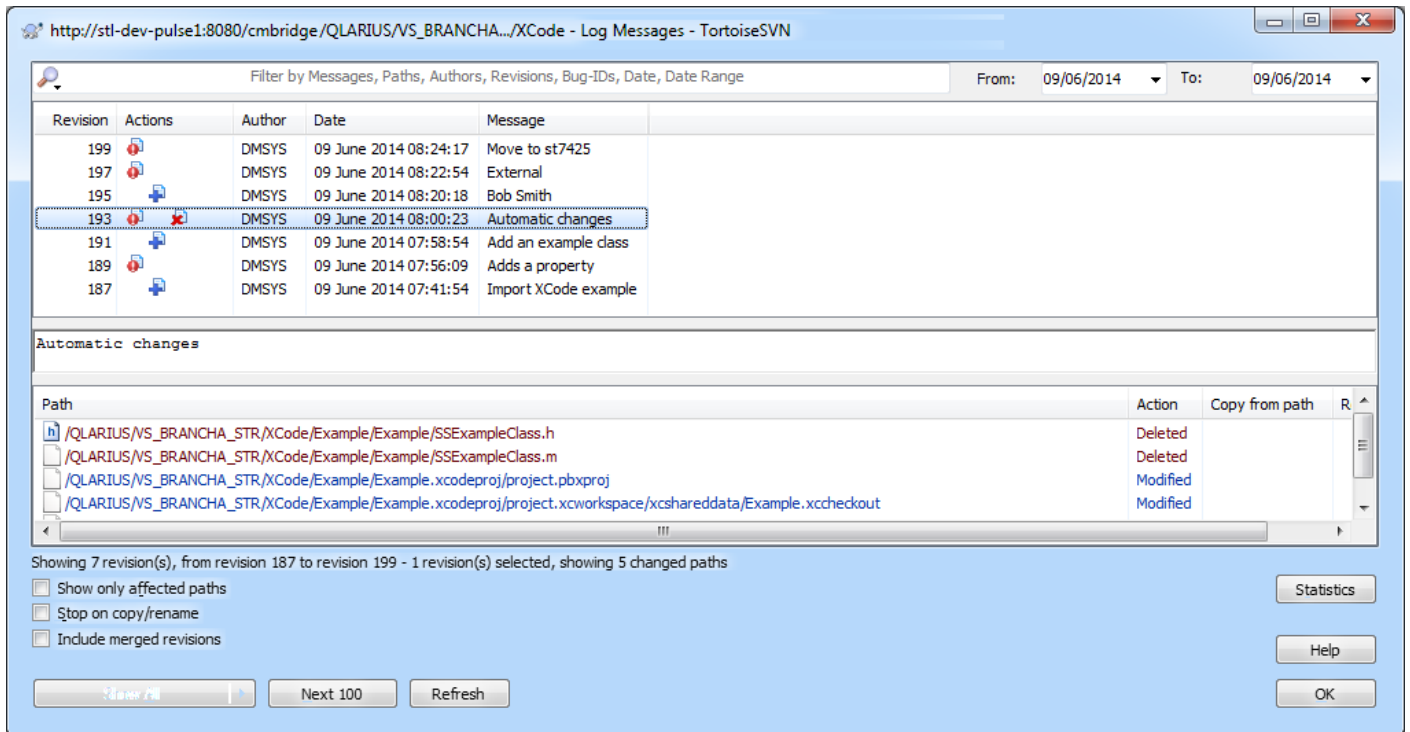

Check Out a Folder

You can use TortoiseSVN to get the contents of a stream to a local area:

1 From Windows Explorer open the TortoiseSVN repository browser.

2 Enter a URL to a Dimensions CM stream, for example:

```
http://<web server>:8080/cmbridge/QLARIUS/JAVA_BRANCHA_STR
```

3 If prompted, enter a CM user ID and password. The structure of your repository is displayed and the stream you specified is expanded.

4 Right-click the stream and select **Checkout**.

5 Enter the location where you want to check out the files.

6 Click **OK**. The files and folders are copied from the Dimensions CM stream to your machine.

Commit Changes to Dimensions CM

You can use TortoiseSVN to commit changes that you have made locally back to a Dimensions CM repository. You can also relate commits to Dimensions CM change requests.

1 If you have added a new file, to register the addition right-click it and from the **TortoiseSVN** menu select **Add**.

2 To commit all your changes, navigate to the stream's parent folder, right-click, and select **SVN Commit**.

3 In the **Commit** box:

- Enter a commit comment for your changes.

- Optionally enter the IDs of one or more Dimensions CM change requests as part of the comment in this format: [CR1,CR2]. For example:

  ```
  [QLARIUS_CR_21,QLARIUS_CR_36] Your comment
  ```

4 Click **OK**.

Chapter 4
Subversion Command Line

Introduction

CM Bridge supports the majority of Subversion commands. The following examples describe some of the ways that you can use CM Bridge with the Subversion command line.

List Files and Folders

Use the Subversion *ls* command to browse the folder structure of a stream and view its details, for example:

```
svn ls "http://<web server>:8080/cmbridge/QLARIUS/JAVA_BRANCHA_STR"
    --username <user ID> --password <password>
```

Use the *ls* verbose option to list additional details about the files and folders and list the subfolders, for example:

```
svn ls -v "http://<web server>:8080/cmbridge/QLARIUS/JAVA_BRANCHA_STR/
    Qlarius Underwriter/qlarius/interfaces" --username <user ID> --
    password <password>
```

The verbose directory listing contains additional details about each file or folder, such as:

- The repository version number when it was last modified.

- The user who last modified it.

- The date and time of the last modification.

List Previous Revisions of Files and Folders

Use the Subversion *log* command to browse the history of a stream's folder structure in a repository and view details of previous revisions of files and folders, for example:

```
svn log "http://<server>:8080/cmbridge/QLARIUS/JAVA_BRANCHA_STR" --
    username <user ID> --password <password>
```

Output:
```
------------------------------------------------------------------------
    -
r199 | EMMA  | 2014-06-09 08:24:17 +0100 (Mon, 09 Jun 2014) | 1 line
Delivered some changes for the CM Bridge preview
------------------------------------------------------------------------
    -
r197 | DMSYS | 2014-06-09 08:22:54 +0100 (Mon, 09 Jun 2014) | 1 line
External test
------------------------------------------------------------------------
    -
r195 | USER1 | 2014-06-09 08:20:18 +0100 (Mon, 09 Jun 2014) | 1 line
Initial delivery
```

In this example, there have been three commits creating repository versions r199, r197, and r195.

Use the *ls* command -r option to list the contents of a specific version of a stream, for example:

```
svn ls –v –R –r 195 "http://<web server>:8080/cmbridge/QLARIUS/
    JAVA_BRANCHA_STR" --username <user ID> --password <password>
```

This command lists the contents of the stream at revision 195.

Check Out a Stream

Use the Subversion *checkout* command to check out an entire stream to a local disk, for example:

```
svn checkout "http://<web server>:8080/cmbridge/QLARIUS/
    JAVA_BRANCHA_STR" --username <user ID> --password <password>
```

Commit Changes to Dimensions CM

Use the Subversion *add* command to register files that have been added and the *commit* command to deliver changes made on disk back to a Dimensions CM stream. You can also relate commits to Dimensions CM change requests.

1   Change directory to the stream's parent folder.

2   Enter the `commit` command. Optionally enter the IDs of one or more Dimensions CM change requests as part of a comment in this format: [CR1,CR2]. For example:

```
svn commit –m " [QLARIUS_CR_21,QLARIUS_CR_36] your comment " --
    username <user ID> --password <password>
```

View File Annotations

Use the Subversion *blame* command to view an annotated copy of a file and display the users who modified each line of code and when, for example:

```
svn blame "http://<web server>:8080/cmbridge/QLARIUS/JAVA_BRANCHA_STR/
    MyNewFile.txt" --username <user ID> --password <password>
```

The command lists which repository version last changed each line and the user who made the change, for example:

```
211      DMSYS This is the first line of the file
213      EMMA Emma added this line
213      EMMA and this line too
211      DMSYS this is the last line
```

Use SVN Properties

You can use SVN properties to create a link between folders so that the work area for one stream can use content from another stream.

1 Use the `checkout` command to get the content of the first stream:

```
svn checkout "http://<web server>:8080/cmbridge/QLARIUS/
    JAVA_BRANCHA_STR" --username <user ID> --password <password>
```

2 Use the `propset` command to set the `svn:externals` property on a folder in the first stream to point to a folder in the second stream:

```
svn propset svn:externals "<URL of shared folder > <folder name in
    work area>" "<folder location in work area>"
```

For example:

```
svn propset svn:externals "http://<web server>:8080/cmbridge/APPS/
    QUICK_QUOTE/Shared MyApps" "Qlarius_Underwriter"
```

This command links the folder *Shared* from the *APPS:QUICK_QUOTE* stream to the subfolder MyApps in the folder `Qlarius_Underwriter` in the checked-out stream.

3 Use the `commit` command to deliver the property change:

```
svn commit -w "Linked the Shared folder"
```

4 Update your working copy or checkout the stream to an empty folder, for example:

```
svn checkout "http://<web server>:8080/cmbridge/QLARIUS/
    JAVA_BRANCHA_STR" --username <user ID> --password <password>
```

The folder `Qlarius_Underwriter` is checked out and the externals are fetched to the correct location on disk.

You can also use SVN properties to set and modify the properties for any file or folder from the Subversion command line.

1 Use the `checkout` command to get the content of the stream.

2 Use the `propset` command to set a custom property on a file in the stream:

```
svn propset <property> "<value>" "<URL of file>"
```

For example, to set the custom property 'complexity' to 'high' on the file MyNewFile.txt in the stream JAVA_BRANCHA_STR:

```
svn propset complexity "high" http://<web server>:8080/cmbridge/
    QLARIUS/JAVA_BRANCHA_STR/MyNewFile.txt
```

3 Use the `commit` command to deliver the change:

```
svn commit –m "Set Complexity to High" --username <user ID> --
    password <password>
```