Log

The **Log** window shows the history of a versioned file or directory ('entry'). A Log is typically invoked via **Query|Log** from the Project Window , but there are several other ways to open the Log window in SmartSVN.

The central component of the **Log** window is the **Revisions** table, which shows the found revisions together with their attributes. You can filter out certain revisions by using the **Revision Filter** field on the top right of the **Revisions** table. To the right of the **Revisions** table, the detailed **Revision Info** of the currently selected revision is displayed.

The lower part of the window shows the **Directories/Files** view for the selected revision. The displayed structure is restricted to the files and directories that are children of the *log context root*; all other files/directories which have been modified within this revision are skipped.

The *log context root* depends on the context from which the log has been invoked. For example, the log context root for logs performed by **Query|Log** from the Project Window is either the corresponding project root directory, or the Externals root directory.

> **Note**
>
> For repositories in Subversion 1.6 format, the received log data contains information on whether a changed entry is of file or directory type. Unfortunately this information is not present for older servers, hence SmartSVN tries to detect the entry types itself. The more log information is present, the better the results of this detection process. However, without complete log information SmartSVN may still be wrong. In this case, the entry is assumed to be a file (although it might actually be a directory).

If *merged* revisions are loaded via **Log|Load Merged Revisions**, they are added in a tree-like manner to their parent revision, which can then be *expanded* or *collapsed*. Because merged revisions have no direct link to the logged revisions themselves, various commands subsequently listed will not be applicable for these revisions. The context root for merged revisions is the corresponding repository root.

At all times, exactly one of the four views is 'active', which is indicated by its highlighted tab title. Menu and toolbar actions are always performed on the currently active view.

Log menu

Use **Load Properties** to fetch all properties for all displayed revisions from the repository (only available for files). The **Revisions** table will be extended by corresponding table columns, one for each property. This command is only available for file Logs. The upper limit of columns to be added can be configured through the system properties .

Use **Load Merged Revisions** to load the originating revisions for revisions that have been merged. This option recursively descends into merged revisions and, depending on the number of merges that have affected the file/directory, this may cause a large or even huge number of revisions to be loaded.

View menu

Select **Skip Unchanged Revisions** to skip revisions for which the logged entry has not actually been changed, but has only been reported due to a copy operation of one of its parents. For example, when creating a Tag of the project root, the log for every entry of that tag will contain this tag-revision.

Select **Stop for Copied Revisions** to stop logging once a revision is encountered for which the logged entry itself has been copied.

Select **Show Files Recursively** to toggle recursive file listing in the **Files** view. If the file listing is recursive, the **Files** view will display not only all files in the directory selected in the **Directories** view, but also all files in subdirectories of the selected directory.

Select **Show Only Entries Below Selected Directory** to restrict the **Directories/Files** view to only those directories and files which are actually children of the logged directory (only available for directories).

Modify menu

Use **Change Commit Message** to edit the commit message of the selected revision. Note that this functionality requires certain server-side hooks to be present and hence might not work.

Use **Merge Revision** to merge the selected revision to your working copy. **Merge Revision** will only be available if the logged branch is different from your working copy branch.

Use **Rollback Revision** to rollback the selected revision to your working copy. **Rollback Revision** will "reverse merge" the selected revision and possibly result in conflicts. After resolving these possible conflicts you may commit the rollback. Be sure to use a meaningful commit message which should probably mention something like "Rollback of revision #...".

File Export

You can export log data in various formats to a file using **Log|Export to File**.

Select either to export **All revisions**, independent of the selection or only the **Selected revisions**. Specify the **Output File** to which the log information will be written. If **Include changed paths** is selected, not only the main revision information but also the details on which files /directories have been changed will be exported.

Specify the file **Format** which shall be used for the export. **XML** will export in raw XML format, as used by `svn log --xml`. **HTML** will give a basic *HTML* output. **Plain text** will give a simply formatted plain text file. **Custom** maybe used to export in an arbitrary format, by performing a style sheet transformation on the raw XML data. In this case, enter the path of the stylesheet for **XSLT-File**.