

Common Features

SmartSVN includes a set of common features and UI elements that are shared by various commands.

Recursive/Depth options

In directory mode, most commands can work *recursively* and *non-recursively*. By default, SmartSVN offers a basic option **Recurse into subdirectories** (or a similar name) which let's you operate either only on the directory itself, or on the directory and all contained files and subdirectories, i.e. recursively.

Alternatively, you can switch to *advanced* recursion options in the [Preferences](#). In this mode SmartSVN offers the Subversion *depth* levels:

- **Only this directory** only operates on the directory/file itself.
- **Only file children** operates on the directory and its directly contained files.
- **Immediate Children (files and directories)** operates on the directory, its directly contained files and subdirectories, but not on files or directories within these subdirectories.
- **Fully recursive** operates on the directory, contained files and subdirectories recursively.

Obviously, having **Recurse into subdirectories** selected is equivalent to depth **Fully recursive** while having **Recurse into subdirectories** deselected is equivalent to depth **Only this directory**.

Revision input fields

Most input fields for which you can enter a revision number, support a *browse* function, which can be accessed either by a **Select** button or by hitting `<Ctrl>+<Space>` keystroke.

A dialog displaying all revisions for the selected file/directory will come up. It shows all revisions for which the directory has actually been affected and additionally all revisions which correspond to a specific tag, see [Tags and Branches](#) for further details. The **Revision** column shows the revision number or the corresponding tag. The other columns display the revision's **Time**, **Commit Message** and **Author**, respectively. The **Path** column shows the revisions's root location.

The displayed revisions are taken from the Log Cache ([Log Cache](#)), so recent revisions might not be contained in the list. In this case you can use **Refresh** to update the Log Cache (and implicitly the displayed revisions) from the repository.

Browse Revisions at specifies the *peg* revision for the location to browse. In general **HEAD** should be sufficient for *alive* locations. Otherwise, you may select the corresponding **Peg Revision**.

Example

When [merging](#) revisions from *replaced* (and hence *dead*) branches, it will be necessary to enter the correct **Peg Revision** to identify the branch.

Repository path input fields

Most input fields for which you can enter a repository path, support a *browse* function, which can be accessed by the **Browse** or by hitting `<Ctrl>+<Space>` keystroke.

The Repository Browser ([Repository Browser](#)) will come up as a dialog. Depending on the command from which the browser has been invoked, you can either select a repository file and/or a repository directory.

For certain commands -- where necessary -- *peg*-revisions are supported. Peg-revisions specify the **URL** of a repository path. This can be helpful when working with paths which do not exist anymore in the repository. In SmartSVN, you can append a peg-revision to a path by prefixing it with a '@'.

Example

To specify a path '/project/path' at revision *91*, enter `/project/path@91`.

Tag input fields

Input fields, for which you can enter a tag, like when using Switch ([Switch](#)), support a *browse* function, which can be accessed by the **Browse** button or by hitting `<Ctrl>+<Space>` keystroke.

The Tag Browser ([Tag Browser](#)) will come up to let you select a branch or tag.

For certain commands -- where necessary -- *peg*-revisions are supported. For details refer to [Repository path input fields](#).

Example

To specify a tag 'my-tag' at revision *91*, enter `my-tag@91`.

File/directory input fields

Input fields, for which you can enter a path to a file or directory, support a *browse* function, which can be accessed by selecting the **Choose** button or by hitting `<Ctrl>+<Space>` keystroke.