

Shell Integration

SmartSVN offers a *shell integration* to have the SVN functionality of SmartSVN also present in certain parts of *GUI* shells, like in *file dialogs*. The shell integration is currently present on *Microsoft Windows* and *Apple Mac OS X*. It is only available when SmartSVN is running (except the one on Mac OS X 10.6).

Commands (Windows and OS X 10.5)

From the shell's context menu, there are the most important SVN commands available for locally versioned files and directories. Performing commands from the shell's context menu results in the same dialogs and windows as if performing the commands from the [Project Window](#) . For details regarding the commands refer to [Commands](#).

For commands performed from the shell, the same environmental settings are used as when performing them from the Project Window. This especially implies the [Project Settings](#) , if for the current working copy directory a corresponding project exists. If no matching [Project](#) can be found, SmartSVN will use the [Default Settings](#) .

From the context menu, use **Open Project** (or **Open SmartSVN** if no file/directory is selected) to launch the [Project Window](#) and open the corresponding project.

Tip

For the command icons, the icon files within `lib/icons` in the installation directory of SmartSVN are used. The names are corresponding to the command names. For every command, there is a default icon and a *grayed* version, which has an additional `-g` in its name. If you prefer, you can replace these icons.

Commands (OS X 10.6)

Unfortunately, Apple has dropped the Finder integration API with OS X 10.6. Hence, SmartSVN only can provide a very simple alternative using so-called services. From the Finder's context menu three commands are available if files or directories are selected: **Update from SVN**, **Commit to SVN** and **Open in SmartSVN**. Note, that because of the limited services API these commands are available independent of the SVN state of these files or directories. They are even available for items which are not SVN-controlled. In contrast with the shell integration on Windows and OS X 10.5, SmartSVN does not need to be running to be able to invoke the commands. If necessary, SmartSVN will start automatically.

Output Window

All commands invoked from the shell integration will be executed in a special *output* window. You may select **Close automatically on success** to have the window closed automatically after all currently running operations have been completed successfully.

File menu

- Use **Show Changes** on a selected file/directory to see what has been changed locally by executing the command.
- Use **Log** on a selected file/directory to see the corresponding [Log](#).
- Use **Close** to close the frame.

Edit menu

- Use **Stop** on one or more selected commands to cancel them. If no command has been selected, you will be asked whether to cancel all currently running commands.
- Use **Customize** to customize accelerators.

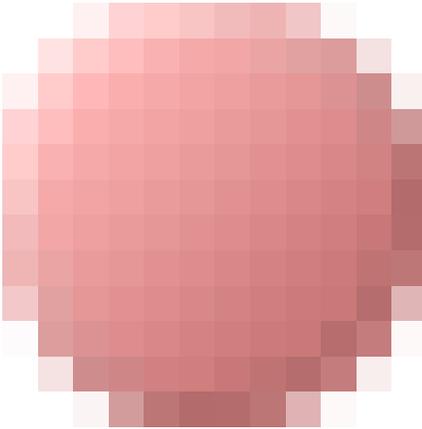
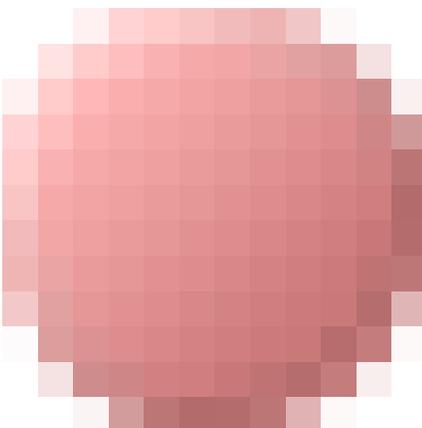
Window menu

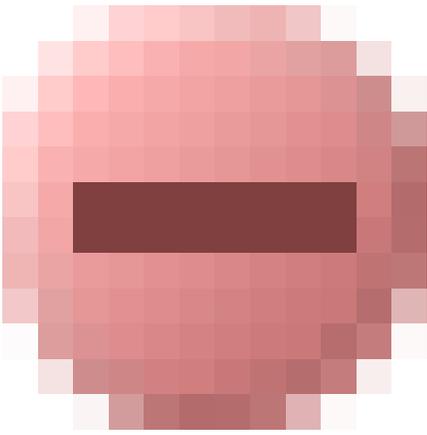
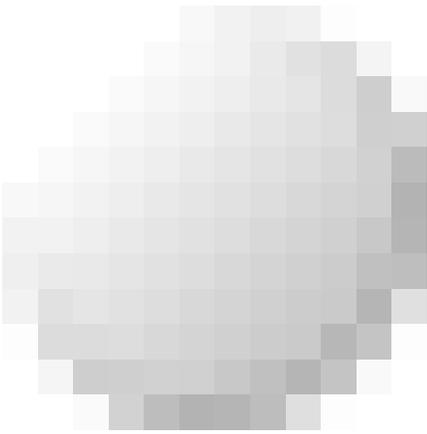
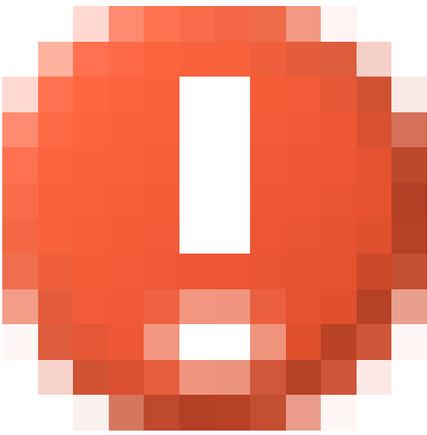
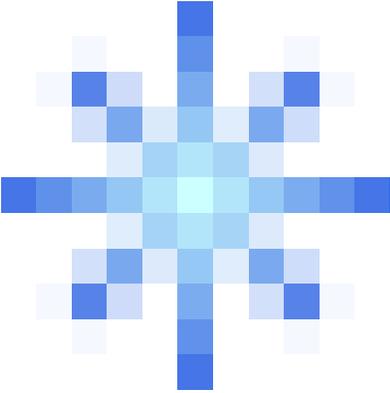
Refer to [Window](#) for more details.

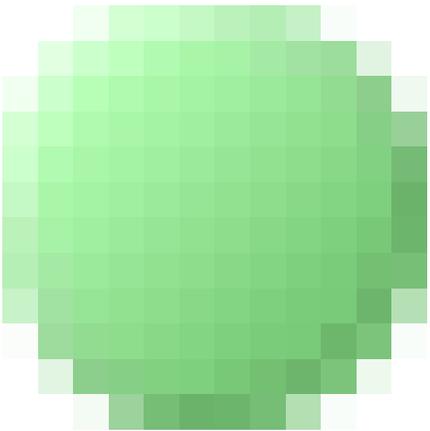
Overlay Icons

The *overlay icons* show the *SVN states* for the corresponding files and directories. Currently, overlay icons are only present on *Windows*. Because the number of possible overlay icons is limited by the operating system, only the most important SVN states have a special overlay icon, see [Overlay Icons](#) for details. Versioned, but unchanged files and directories do not have a special overlay icon. For all other SVN states, the *modified* icon is used.

Overlay Icons

	Modified	File/directory is modified in contents /properties.
	Modified recursively	Directory itself or some file/subdirectory is modified (requires the Status Cache service running).
	Added	File/directory is scheduled for addition.

	Removed	File/directory is scheduled for removal.
	Ignored	File/directory is not under version control (exists only locally) and is marked to be ignored.
	Conflicted	An updating command lead to conflicting changes either in content or properties.
	Unversioned	File/directory is not under version control, but only exists locally.

	Root	Directory is a working root and is not modified.
---	------	--

Tip

For the *overlay icons*, the icon files within `lib/icons` in the installation directory of SmartSVN are used. The names are corresponding to the *States* used in [Overlay Icons](#). If you prefer, you can replace these icons.

The availability of overlay icons as well as commands can be configured in the [Preferences](#).

Note

On Windows, for technical reasons no icon overlays for files within your profile directory `%USERPROFILE%` are shown (except of sub-directory `My Documents`).

Server Mode

To provide the *shell integration* without requiring SmartSVN actually being *open*, SmartSVN can be started with the `--server-mode` argument, for details refer to [Command line arguments](#).

Windows Shell Integration

The shell integration adds *overlay icons* to directory and file views of Windows and SVN commands to the context menu for directories and files. You will especially see them for the *Windows Explorer*, but also for other software which e.g. uses the native file dialogs of Windows.

Installation

You can choose to enable the shell integration for the installation of SmartSVN when using the installer bundle. It's also recommended to have SmartSVN automatically be started with the system startup, so the shell integration is available immediately. The installer offer a corresponding option which will add SmartSVN to the *Autostart* section, starting SmartSVN in [server mode](#).

Note

Windows supports only a quiet small number of overlay icons. On Windows 10, for example, by default too much overlay handlers are already installed and hence you will see a confirmation dialog that will ask whether to put the SmartSVN overlay icons before the other overlay icons. If you choose not to do that, some overlay icons, e.g. for conflicted files, will **not** be shown.

Uninstallation

The shell integration will be uninstalled together with SmartSVN. You can also uninstall the shell integration independently from the *Control Panel, Software*, using *Repair* there.

Configuration

In the [Preferences](#), you can configure for which drive types and in which range of functions the shell integration shall be applicable. For every drive type you can choose whether to show **Icon Overlays** (and the context menu) or only the **Context Menu** or have the shell integration be completely **Disabled**. If necessary, specify further **Paths** for which the shell integration will only be applicable with a limited range of functions, either only the **Context Menu** or completely **Disabled**. Use only plain paths, like `c:\temp` or `n:`, but no patterns here.

Note

In general it's recommended to have **Icon Overlays** only enabled for **Fixed Drives**, because the display of the overlays may slow down your machine due to access to the *working copy metadata* (.svn directory).

When having working copies located on fast network shares, **Icon Overlays** should work well here, too. In case you have a mix of fast network shares and, for example, slow VPN-tunneled shares, you may exclude the latter ones by the **Paths** input field.

Mac OS X Finder integration

The Finder integration lets you update or commit files from the Finder or just open the working copy in SmartSVN. You can find the menu items **Update from SVN**, **Commit to SVN** and **Open in SmartSVN** in the Finder's context menu.

Tray Icon

By default, SmartSVN keeps running even when all frames have been closed. To have SmartSVN still accessible, a *tray icon* is used. It's available for *Microsoft Windows*, most *Linux* desktop managers and other operating systems for which tray icons are supported.

From the context menu of the tray icon, use **New Project Window** to open a new [Project Window](#), **New Repository Browser** to open a new [Repository Browser](#) or **Show Transactions** to open the [Transactions frame](#). Open the **Preferences** or information **About SmartSVN**. To exit SmartSVN, use **Exit SmartSVN**.

Note

On Mac OS SmartSVN is permanently available when SmartSVN is running, even when all frames are closed. In this case it has a reduced menu bar, including the **Window** menu.

The tray icon shows the progress of currently processing SVN operations which have been invoked from the shell extensions. It also shows the presence of *new* revisions for the [Transactions frame](#); the tooltip gives more information on which repositories have new transactions.

You can disable the tray icon in the [Preferences](#), **Low-level Properties**, by setting property `ui.systemtray.enabled` to `false`. In this case, SmartSVN will exit once the last frame has been closed.

Note

`ui.systemtray.enabled` property is not regarded when starting SmartSVN in [server mode](#).

Status Cache

The Status Cache is an optional *Windows service* which manages *SVN status* information for your working copies. It's primarily used to display the *recursively modified* state for directories, which is denoting that some files/subdirectories are modified. Also, the [initial scanning/refresh](#) accesses Status Cache information to quickly give a preview of the working copy.

To avoid unnecessary system load, the root directories which will be served by the Status Cache have to be explicitly configured. SmartSVN will ask you to do so for the first command which you perform through the Shell Integration. The Status Cache can be reconfigured any time in the [Preferences](#):

Cache Roots specifies the file system roots which will be served by the Status Cache. Enter every root directory on a new line, wildcards are *not allowed* here. Optionally you can reset the Status Cache by **Clear all cached status information**. Selecting this option is only recommended if you definitely want to get rid of cached status information for a certain root directory as cached information is not discarded by simply removing this root directory from the **Cache Roots** list.

Performance considerations

You should carefully determine which root directories should be served by the Status Cache, as the Status Cache will introduce a certain overhead to your system's load. This overhead comes more apparent the slower the file system to cache is. In general you should:

- Only configure to cache local harddisks
- Avoid caching of possible temporary directories which might receive temporary working copies
- Don't create a too detailed list of individual directories to cache

So for instance, if all of your working copies are located at a separate drive `D:`, it will be perfect to have the Status Cache configured for this single root directory `D:` and nothing else.

Uninstallation

If you are only rarely working with the Shell Integration and additional *recursively modified* state is not important to you, you may completely uninstall the service. This can be done via the *Control Panel/Add or Remove Programs*, selecting the *SmartSVN* installer, **Change** and within the installer using **Change** again.

Settings Directory

The Status Cache settings are stored in `%COMMONAPPDATA%\SmartSVN\statuscache-1`, which is usually `C:\ProgramData\SmartSVN\statuscache-1`. System properties can be configured in `statuscache.properties`.

Known Issues

The SmartSVN Status Cache service runs under the system account and hence does not know about your SVN `config` file. Hence, `global-ignores` specified in this file will **usually** not be honored. There has been a workaround implemented for the common case where there is only one regular user account on your machine: in this case, the Status Cache will use this account's SVN `config` file instead. If this won't work for you, you may specify a custom SVN configuration directory in `statuscache.properties`, using system property `statuscache.svnConfigDirectory`. For example:

```
statuscache.svnConfigDirectory=C:/Users/marc/AppData/Roaming/Subversion
```

System properties

System properties affecting the Status Cache will be configured in `statuscache.properties`. Following system properties are available:

`statuscache.refresh.background`

Usually a file system refresh will only be triggered once status information is requested from the Explorer integration – this is done for performance reasons. "Recursive modification" information may become outdated, though. To avoid this, set this property to `true`. Be aware that this will increase refreshing activity!

`statuscache.refresh.onSvnWcDbChange`

For performance reasons, external modifications to the `.svn/wc.db` file will be ignored. This may result in outdated status information when working with another SVN client, like the command line client. If you want to still force SmartSVN to refresh in this case, set this property to `true`.

This may result in a significant rise of refreshing activity.