

# Sablime<sup>®</sup> v6.1 and sfw1.1

## Eclipse\* Plug-in Guide April 2005

Revised 4/20/2005 – Added technical notes on page 7

Revised 6/21/2005 – Incidental change: Removed hyperlinks to sample and other non-browsable URLs.

### What this Guide Covers

This guide provides information about and instructions for installing and using the Sablime Eclipse plug-in. This plug-in supplies an interface between Sablime and the Eclipse IDE framework.

### Requirements

The Sablime Eclipse plug-in works with Web Sablime-supported platforms. Users must be enabled for Web Sablime in order to use the Eclipse plug-in.

The Sablime Eclipse plug-in works with both Eclipse 2.x and Eclipse 3.x, although there are some differences in functionality (as detailed below).

### Installation

The Eclipse plug-in is installed using the Eclipse Feature Update facility and the process differs between Eclipse 2.x and Eclipse 3.x.

#### Initial Installation in Eclipse 2.x

1. Start Eclipse, and on the "Help" menu, select "Software Updates -> Update Manager..."
2. On the "New Features" tab, right-click and select "New -> Site Bookmark..."
3. Enter the name "Sablime Eclipse Plug-in", and the URL "<http://www.bell-labs.com/project/sablime/eclipseplugin/update/>" and click "OK".
4. The created bookmark should have a diamond icon and, drilling down, should show "Other" and then "Sablime Eclipse Plug-in 1.1.0"
5. Click on "Sablime Eclipse Plug-in 1.1.0" to bring up the plug-in description in the Preview pane (upper right).
6. Scroll down and click on the "Install Now" button.
7. Click "Next"; accept the license agreement and click "Next"; then click "Finish". Note: the plug-in takes about 230KB disk space to install.
8. At the next dialog, click "Install" to install the plug-in.

9. Click "Yes" to restart Eclipse.

If the installation is successful, then on the "Help" menu, "About Eclipse Platform -> Plug-in Details" will show the Sablime plug-in at the bottom of the list.

### Initial Installation in Eclipse 3.x

1. Start Eclipse, and on the "Help" menu, select "Software Updates -> Find and Install..."
2. Select "Search for new features to install..." and click "Next"
3. Click "New Remote Site..."
4. Enter the name "Sablime Eclipse Plug-in", and the URL "http://www.bell-labs.com/project/sablime/eclipseplugin/update/" and click "OK".
5. Check the box for the Sablime Plug-in and click "Next".
6. Check the box for "Sablime Eclipse Plug-in 1.1.0" and click "Next".
7. Accept the license, and click "Next".
8. Click "Finish". Note: the plug-in takes about 230KB disk space to install.
9. Click "Install" to finish the installation.
10. Click "Yes" to restart Eclipse.

If the installation is successful, then on the "Help" menu, "About Eclipse Platform -> Plug-in Details" will show the Sablime plug-in at the bottom of the list.

### Plug-in Updates

If you are notified that there is an update available for the Sablime Eclipse plug-in, you can use the Eclipse Feature Update to install the update.

### Update Installation in Eclipse 2.x

1. Start Eclipse, and on the "Help" menu, select "Software Updates -> Update Manager..."
2. Drill down to the Site Bookmark you created in the initial installation. It should show "Sablime Eclipse Plug-in <currently installed version>".
3. Click on "Sablime Eclipse Plug-in <currently installed version>" to bring up the plug-in description in the Preview pane (upper right).
4. Scroll down and click on the "Update Now" button.
5. Click "Next"; accept the license agreement and click "Next"; then click "Finish".
6. At the next dialog, click "Install" to update the plug-in.
7. Click "Yes" to restart Eclipse.

## Update Installation in Eclipse 3.x

1. Start Eclipse, and on the "Help" menu, select "Software Updates -> Find and Install..."
2. Select "Search for updates of the currently installed features ..." and click "Next"
3. Check the box for the Sablime Eclipse Plug-in and click "Next".
4. Check the box for "Sablime Eclipse Plug-in <currently installed version>" and click "Next".
5. Accept the license, and click "Next".
6. Click "Finish".
7. Click "Install" to finish the installation.
8. Click "Yes" to restart Eclipse.

## Uninstall Sablime Eclipse Plug-in

### Eclipse 3.x

1. Start Eclipse, and on the "Help" menu, select "Software Updates -> Manage Configuration..."
2. Select "Sablime Eclipse Plug-in 1.1.0", and click "Uninstall" on the right-hand pane.

### Eclipse 2.x (or if above fails on Eclipse 3.x)

1. Stop Eclipse
2. Navigate to the directory where Eclipse is installed.
3. In the features sub-directory, delete the com.lucent.sablime\_1.1.0 directory.
4. In the plugins sub-directory, delete the com.lucent.sablime\_1.1.0 directory.
5. In workspace\plugins, delete any com.lucent.sablime\_1.1.0 directory

## Usage

Users must be enabled for Web Sablime before using the Eclipse plug-in. The user should log into the Web Sablime URL at least once (to establish a default preferences file) before using the plug-in.

### Share an Existing Eclipse Project with Sablime

1. Right-click on the project and select "Team -> Share Project..."
2. Select "Sablime" from the Repository Type list
3. Enter your Web Sablime server URL (e.g.,

<http://mymachine.mydomain.com:6060>); user name and password; click "Next".

4. Select the Sablime product and generic from the drop-down lists.
5. Select an MR from the drop-down list of MRs assigned to you, or create a new MR with the "New MR" button. This MR becomes the default MR for the project and is used to "addsrc" files from the project into Sablime.
6. If the "Automatically add current project files" check box is unchecked, the MR is optional.
7. Select the parent directory for the project from the drop-down list of existing directories in Sablime (If the desired parent directory does not currently exist in Sablime, it must be created on the server using the "adddirs" command). Next enter the name of the sub-directory that the project will be stored in. The default name is the name of the project in Eclipse.
8. The plug-in will not share a project into an existing directory unless "Automatically add current project files" check box is unchecked. Note: this feature can be used to "re-attach" an Eclipse project that has been accidentally deconfigured.
9. If the "Automatically add current project files" check box is checked, all files in the project, except those matching patterns in "Window -> Preferences -> Team -> Ignored Resources", will be added.
10. Click "Next" or "Finish"
11. The optional last page of the dialog contains default settings for the project for various Sablime commands.
12. Click "Finish" to start the share operation.

### Import an Eclipse Project (Stored in Sablime) into an Eclipse Client

1. From the "File" menu, select "Import ..." and pick "Sablime Project" from the list.
2. Enter your Web Sablime server URL (e.g., <http://mymachine.mydomain.com:6060>); user name and password; and click "Next"
3. Select the Sablime product and generic from the drop-down lists.
4. Optionally, select an MR from the drop-down list of MRs assigned to you, or create a new MR with the "New MR" button. This MR becomes the default MR for the project.
5. Select whether you would like to choose from a list of directories in the generic that appear to be Eclipse projects (i.e. they have a .project file) or from all directories in the generic. Click "Next".
6. Select a directory to import.
7. Enter the name of the project to create. The default is the last component of the directory name.
8. Select a local directory for the project. The default is a directory with the same name as the project, in the current workspace.

9. If you uncheck "Use default value", you may create the project in any directory outside of the workspace (use "Browse" to create a new directory).
10. If the "Download File Content" check box is checked, all files in the Sablime directory will be downloaded into the new project.
11. If the box is unchecked, the project will show a view of the Sablime directory with all the files tagged with the "[Remote]" tag. The files may be made local by using the "Check Out" action or the "Replace With..." actions. This feature is useful to only download some files of a project or directory containing a large number of files.
12. Note: Some Eclipse functions or plug-ins will not work properly with "remote" files. In particular, the JDT complains bitterly in the log file about not being able to read the .classpath file.
13. The .project file is always downloaded if it exists regardless of the "Download File Content" setting.
14. Click "Next" or "Finish".
15. The optional last page of the dialog contains default settings for the project for various Sablime commands.
16. Click "Finish" to start the import operation.

Note: The "perspective" does not always automatically switch to the preferred perspective for projects that have "natures".

## Stop Sharing an Eclipse Project with Sablime

1. Right-click on the project and select "Team -> Deconfigure".

## Sablime Team Menu Actions

Right-click on a file, folder or project and select "Team" to bring up the Team menu. The currently supported actions are:

**Refresh Status** (update current status of file from Sablime database): File status is not periodically updated, so this command can be used to manually refresh the status. The project or folder form of this action discovers files that have been added or deleted in Sablime outside the current Eclipse instance. The new file will show up with the "[Remote]" tag; you must use either "Check Out" or "Replace With..." to retrieve the file's contents. Files deleted in Sablime are not deleted in Eclipse (unless they were remote to begin with).

**Add to Sablime** (add a new file to the Sablime database)

**Check Out** (edget)

**Check in** (edput)

**Undo Check out** (unedput)

**Reserve/Unreserve** (reserve or unreserve a checked out file)

**Merge/Marked as Merged** (used to resolve Check in conflicts)

Right-click on a file and select "Replace With -> Latest from Official Branch" or "Replace With -> Latest from MR Branch" to replace local version of a file with the latest version from the official or MR branch in the Sablime database.

Right-click on a file and select "Compare With->Latest from Official Branch" or "Compare With->Latest from MR Branch" to compare local version of a file with the latest version from the official or MR branch in the Sablime database.

An action on a folder or the project applies the action to all eligible files and subfolders (recursively) under the folder or project. In addition, multiple files may be selected for an action (using CTRL-left click). Actions that operate on multiple files bring up a confirmation dialog showing the file list so that files can be deselected if necessary.

Most actions bring up a confirmation box in which options can be configured (see action notes below).

## Item Icons when Under Sablime Control

**Project:** The project icon will have a database symbol (🗄) overlaid on the lower right, and "[sablime @ <Web Sablime Server URL> (<user>)]" appended to its text label.

**Folder:** The folder icon will have a database symbol (🗄) overlaid on the lower right.

**File (Checked in):** The file icon will have a database symbol (🗄) overlaid on lower right.

**File (Checked out)** (by this Eclipse client): The file icon will have a checkmark symbol (☑) overlaid on the lower right, and "[<mr>]", or "[<mr> (reserved)]" appended to its text label (where <mr> is the MR used for the check out).

**File (Checked out)** (elsewhere): The file icon will not have a check mark symbol, but "[OtherCheckouts]" or "[OtherCheckouts (reserved)]" will be appended to its text label. Note: this information will only be current as of the most recent "Refresh Status".

**File (Remote):** If a file's text label has the "[Remote]" tag, the file has no local content and cannot be viewed until the content is retrieved from Sablime using either the "Check Out" or "Replace With..." actions.

## Notes on Individual Sablime Actions

**Add to Sablime:** A different MR may be selected from the "Sablime Properties" of the confirmation dialog. Addsrc options may be set (e.g., set version control tool to SBCS) from the "Sablime Options" tab.

**Check out:** A different MR may be selected from the "Sablime Properties" of the confirmation dialog. Edget options may be set (e.g., reserved edget) from the "Sablime Options" tab.

**Check in:** A check in comment may be entered using the "Comments" tab of the confirmation dialog. Edput options may be set (e.g., file-level dependency) from the "Sablime Options" tab.

**Merging:** If conflicts are detected on a check in of a file (because other changes have been edput to the file by another MR outside of the Eclipse instance), then the file enters Merge mode (indicated by the "(MergeNeeded)" tag in the file's label).

The workbench will show a 3-way comparison between the last common

ancestor (top pane but initially hidden), the latest version in Sablime (right pane) and the current local version (left pane). If the comparison view is dismissed, it can be brought back selecting "Team -> Merge..."

You can make changes to the local version (in the comparison editor) to resolve the conflict. When the conflict is resolved, right-click on the file and select "Team -> Mark as merged". Right-click on the file again and select "Team -> Check in" to complete the check in.

## Property Pages

There are Sablime-specific property pages (i.e., right-click and select properties) for files, folders and projects. The default MR for a project can be set on the sablime page of the project's properties.

The username and password used to access Web Sablime may be changed as well.

In addition, Sablime plug-in defaults can be configuration through the property dialog at "Window -> Preferences -> Team -> Sablime".

## Technical Notes

**What gets added when an Eclipse project is added to Sablime:** All the files in the project are addsrc'd, excluding those that match some pattern in the list on the "Window -> Preferences -> Team -> Ignored Resources" page. Various plug-ins (e.g., the Java IDE, the C++ IDE, etc) have files that aren't source files and shouldn't be added to Sablime; unfortunately not all of these files are on the default ignore list. Users should add these patterns to the ignore list:

- \*.class (From the Java IDE - no idea why this isn't in the default ignore list)

- build.xml (generated build script)

We should also note that there may be other files you will want to add to the ignore list at your discretion.

**Consequences of adding project metadata files such as .project and .classpath to Sablime:** We're confident that adding these files to Sablime is the right way to go since these files contain important information about the project (dependencies, build paths, etc). Users should be aware, though, that when the project is saved, the project properties are saved too, so it's important to ensure that those property values are as developer independent as possible. In the case of the Java IDE, that means that "Project -> Properties -> Java Build Path" shouldn't contain any absolute paths.

**Consequences of metadata files being read-only:** Most metafiles are edited indirectly by using a properties page or special editor. It's the editor's responsibility to make the call that triggers the hook that Sablime uses to check out a file if necessary. Unfortunately, sometimes the editor doesn't do this and the operation fails because the underlying file isn't writable. This seems to be much less of a problem in Eclipse 3.x. The workaround is to check out the file manually before invoking the editor.

The .project and .classpath files are a bit of a special case since they are somewhat hidden and an inexperienced user may not realize the connection

between them and the "Project -> Properties" page. You need to use the "Navigator" View ("Window -> Show View -> Navigator") to see them. In Eclipse 3.x there shouldn't be any problem editing these files via "Project -> Properties" (the user will be prompted to check the file out).

**Making local changes to metadata files:** If the user wants to make a change to a metadata file without checking it out (e.g., make a test change to the class path), they can make the file writeable (either in the underlying file system or via the file's properties in Eclipse). Of course, caution must be taken to make sure that these experimental changes aren't lost when a check-out/check-in happens.

## Known Problems and Limitations

- If a user has been configured to use Web Sablime, but has never actually used Web Sablime, the Import or Share Project operation may fail. As a workaround, the user should access Web Sablime at least once before using the Eclipse plugin.
- The root directory of a generic (i.e., ".") cannot be imported (as either a project or a directory).
- Linked resources are not supported.
- Folders are not really managed by the plug-in. Every local folder in a project shared with the Sablime Team Provider is displayed with the "checked in" icon, but that does not necessarily mean there is corresponding directory in Sablime.
- When temporary branching is turned off (in Sablime), every check out is a reserved check out. Currently the plug-in shows a file's status with the checkmark and [`<mr>`] after the check out, but does not display "(reserved)" until a refresh status action (at any level: project, folder, or file) is performed.
- Linux\* and Solaris\* only: The default font used in the dialogs is larger than the Windows\* versions and some text is clipped because of it. There should be some way to override this font but none of the font settings in the Eclipse preferences seem to affect it.
- Linux and Solaris: Occasionally, after entering the Sablime URL and login information, the product and generic list does not appear. Press the back button and then "next" again.
- Solaris only: Using the file properties dialog to change a read-only file to writable does not work. This appears to be a problem with Eclipse on Solaris. If you attempt to modify the file, and answer "yes" when it asks if you wish to check the file out, it works correctly.
- Solaris only: Occasionally, you may not be able to modify an checked-out file if you still have a window open from attempting to modify the file

while it was read-only. Close the other window, and open the checked-out file again.

- 
- \* Eclipse is a trademark of Eclipse Foundation, Inc.
  - \* Linux is a registered trademark of Linus Torvalds.
  - \* Solaris is a trademark or registered trademark of Sun Microsystems, Inc. in the U.S. and other countries.
  - \* Windows is a trademark or registered trademark of Microsoft Corporation in the U.S. and other countries.

Sablime is a registered trademark of Lucent Technologies. Contents copyright © 2005 Lucent Technologies. Permission to photocopy in support of a licensed installation of Sablime is hereby granted.