

Sablime® v6.1 Update 2

Upgrade Guide
June 2006

What this Guide Covers

This guide provides information and instructions for downloading Sablime and upgrading a Sablime instance to v6.1u2.

There are two sections: one for users upgrading from Sablime v6.1 or v6.1u1 to v6.1u2, and the second section for all other upgraders.

First-time installers of Sablime should follow the instructions in the [Sablime v6.1 Installation Guide](#).

Upgrading from Sablime v6.1 or v6.1u1 to Sablime v6.1u2.

The upgrade from v6.1 (base) or from v6.1u1 (v6.1, Update 1) to v6.1u2 (v6.1, Update 2) is a simple matter of downloading the update and running the update script.

There is also an optional database conversion script that needs to be executed in order to take advantage of the "User Defined Fields available to more commands" and "Unedput able to leave file checked out" enhancements.

Download the file (sab61u2_<architecture>_update.tar.Z) from [Sablime's v6.1u2 download page](#). If your browser renames the file (changing the first "." to a "_"), rename it back to its original name before continuing.

Also download the update script "update_sab61_to_u2". Place these two files into the same directory on the Sablime UNIX* /

Linux* host. (This script will update either base v6.1 or v6.1u1 to v6.1u2).

Note: The installation instructions are intended for use with the Korn shell (ksh). The upgrade should be done while logged into the Sablime host machine as the user ID that owns the Sablime binaries and databases. Please log in as this user instead of using "su". This user (usually "sablime") should NOT be the "root" user.

Note for Linux users: The shell variables SHELL and EXECShell must both be set to the location of ksh (normally /bin/ksh) and exported. The variable PWD must also be exported (pdksh does not export PWD by default).

- 1) Verify that the update file ("sab61u2_<architecture>_update.tar.Z") and the update script ("update_sab61_to_u2") are present in the current directory.

```
$ ls sab61u2*.Z update_sab61_to_u2
sab61u2_<architecture>_update.tar.Z
update_sab61_to_u2
```

- 2) Run the Sablime setup script for a generic in the v6.1 instance you wish to update.

```
$ . sablime <your_generic>

(...Normal Sablime setup output....)
```

- 3) Make sure a copy of **perl** (version 5.004_01 or later) is in your \$PATH.

```
$ perl -v

This is perl, version 5.004_01
...etc...
```

- 4) Run the update script

```
$ ksh ./update_sab61_to_u2
```

(The script will display the location of your current Sablime bin and its current version. If the version is v6.1 or v61u1, the script will ask if you're ready to proceed. If so, answer "y" or "yes".)

Note: The update script might report that it has "insufficient permissions" to install either "wsab_install" or "config.input.tpl."

This means that the script noticed a difference between your copy of these and the most current one, but since the file or directory is owned by root, the script wasn't able to update it.

These are used in the basic installation of Web Sablime, and because you already *have* a working Web Sablime installed, you don't really need to update them. But in case you ever want to re-install, you should ask someone with Root privileges to copy the files as suggested by the script.

After it finishes updating the executables, the script will offer to run the database conversion for you. The conversion will upgrade the databases for all the products in the instance where you initialized Sablime (in step 2, above).

If you choose to skip the conversion (or if you want to run the conversion manually on databases not covered by the initial conversion) you can run it yourself by initializing Sablime to a generic in the instance, and then typing "**\$sabLCB/dbconvert**".

Note: The Database conversion for this update supports the extension of User Defined Fields to more commands and the ability of edput to leave a file checked out. Because the conversion is small and simple, the usual audits and backups are not needed, nor is it necessary to stop and restart the databases.

The script should finish in less than a minute, and will say "Update complete" after a successful update.

- 5) Remove the download file and the update script (if desired).

```
$ rm sab61u2_<architecture>_update.tar.Z update_sab61_to_u2
```

- 6) Optional: Remove the working files and backup files.

The update script makes a directory and stores working files there. Among them is a full "tar" backup of all the files that it replaced or removed. Once you are satisfied that the update is performing properly, you can remove the directory and its contents.

```
$ cd $sabLCB
$ rm -fr v61u2_update
```

It is harmless (other than the disk space usage) to keep the directory around indefinitely.

If you have any problems with the upgrade, you can re-run the script and it will pick up where it left off. If you want to back-out the upgrade you can re-run the script with the --rollback option.

The remainder of this Guide is for users upgrading from releases prior to v6.1.

Upgrading from Sablime v5.0, v5.1, v5.2 or v6.0

Scheduling the Upgrade

Users will not have access to the Sablime instance during the upgrade, so you should plan accordingly, and notify the users in advance.

Although upgrade failures are very rare, you should allocate additional time to restore the instance from backups if necessary.

How much time?

The download and installation steps take only a short while, depending on your network and machine capabilities. Downloading can, of course, be done ahead of time.

Running the recommended audits will take longer, and the duration will be highly dependent on the size of your database. Assuming you run the audits regularly, you should be able to predict the amount of time for them. (If you do not run audits regularly, you should run them before scheduling the upgrade, since you may have a substantial number of issues to resolve).

The (optional) backups will be simply a matter of copying the databases to another location. If you have available disk space, this can be rather quick. Probably about the same time as the audits. If you have to backup to tape or such, the duration will be greater.

Database conversion takes usually a few seconds to a few minutes per product. The time depends on the size of the database and the available CPU.

Package Description

Sablime v6.1u2 is packaged as a compressed UNIX/Linux "tar" file. "Tar" is a utility that packages multiple files into one. This file is then compressed using the "compress" utility.

There is a different package for each architecture. We use "architecture" to designate the hardware type and operating system of the Sablime host. "Sparc5", for example, is the architecture designation for a Sun Microsystems machine running the "SPARC" processor with Solaris* 2.4 or above as the operating system.

A license file is required in order to run most Sablime commands. Your existing license should continue to work with Sablime v6.1u1.

Download

Download the file (sab61u2_<architecture>_full.tar.Z) from [Sablime's v6.1u2 download page](#). If your browser renames the file (changing the first "." to a "_"), rename it back to its original name before continuing.

The "patchlevel" and "whatoutput" are each included within the "tar" file, so you do not need to download them separately. They are on the web site for reference purposes.

If you did not download it there directly, move or copy the file(s) to your UNIX/Linux Sablime host machine.

Upgrade Instructions

The following upgrade instructions and the Sablime software itself are intended for use with the Korn shell (ksh).

Note for Linux users: The shell variables SHELL and EXECHELL must both be set to the location of ksh (normally /bin/ksh) and exported. The variable PWD must also be exported (pdksh does not export PWD by default).

1. Log in

Log into the Sablime host as the user ID that owns the Sablime binaries and databases. Please log in as this user rather than using "su".

2. Stop the databases - Optional

For each product in your instance, issue the "dbstop" command:

```
$ export sabPROD=<product_name>
$ dbstop
```

(dbstop requires that you edit a file that includes an explanation of why the db is stopped and for how long. Since "how long" depends on when it was started, it is really better to include the estimated completion time instead. Something like

this:

```
Sablime is being upgraded to v6.1u2. The expected completion
time is <time>. Contact <admin> for further information.
```

When you exit the editor (assuming the resulting file isn't empty), the product will be stopped and no database-changing commands will be allowed to execute.

If your instance contains multiple products, run the above for each of them.

If you choose not to do this step, the **dbconvert** program, below, will stop the databases during its execution, and restart them when it completes. If you want to do some manual verifications of the conversion before allowing users to make changes, you'll want to stop and start the databases yourself, rather than having **dbconvert** do it.

3. Verify the databases (Run the Audits) - Optional

You might want to verify that your databases are "clean" before conversion. Although highly unlikely, it is possible that a database inconsistency could cause a conversion failure.

For each product:

```
$ export sabPROD=<product_name>
$ dbcross > crossout.$sabPROD
$ dbdelta > deltaout.$sabPROD
$ dbxcross > xcross.$sabPROD
```

Review the three output files and repair reported problems.

4. Backup the databases - Optional

Although conversion problems are rare, they can occur due to system failures or other events. We recommend that you create a full backup of your Source, Active, Inactive, and Global databases.

The following example is for an instance where all four databases (sdb, adb, idb, gdb) are in the same place, and they are being copied to a new directory named db_backups. Adjust the process according to your own setup and backup capabilities.

```
$ cd <db_location>
$ mkdir db_backups
$ find sdb adb idb gdb -type f -print | cpio -pdm db_backups
```

Note: The Database conversion process does not change anything in the Source Database. However, we still recommend that you back up that database so that you'll have a Source database that matches the backed-up pre-conversion *Active* database.

5. Install the v6.1u2 software

5.1 Set aside the current Sablime binaries directory, and create a new directory to be the location of the Sablime binaries.

In these instructions, we'll refer to that location as *SABHOME*.

The examples below assume that you want the new binaries to be where the old ones were.

```
$ mv SABHOME SABHOME.old
$ mkdir SABHOME
```

5.2 Set your "umask" (affects the file permissions of the new files).

```
$ umask 022
```

- 5.3 Move the downloaded file into *SABHOME*.

```
$ mv sab61u2_<architecture>_full.tar.Z SABHOME
```

- 5.4 Uncompress and unpack the tar file.

```
$ cd SABHOME
$ zcat sab61u2_<architecture>_full.tar.Z | tar xvof -
```

Note: This is a UNIX "compress" file, not a PC "zip" file. Do not use PC tools to uncompress it.

- 5.5 Copy your license file into *SABHOME*.

```
$ cp SABHOME.old/.usrld SABHOME
```

- 5.6 Your "sablime" and "xsablime" initialization scripts should be compatible with your new installation. Set aside the ones that come with the new installation, and replace them with your old versions:

```
$ mkdir new_init
$ mv *(x)sablime*(.sh) new_init
$ cd SABHOME.old
$ cp *(x)sablime*(.sh) SABHOME
```

Note: One of the enhancements in v6.1 is better support for filenames with spaces. Your existing xsablime script does NOT properly handle such file names. If this is a concern for you, you'll need to use the new script, and move your local customizations into it. See the [Getting Started Guide](#) and the script comments for instructions on updating xsablime(.sh).

- 5.7 If you are using the Sablime "External MR Communication" feature, there will be files in your *SABHOME.old* named ".BIN" and ".EMR". Copy these to your new *SABHOME*.

```
$ cd SABHOME.old
$ cp .EMR .BIN SABHOME
```

- 5.8 If you have any triggers (in the "hooks" subdirectory) or any other locally crafted software in *SABHOME.old*, copy these items over to *SABHOME*.

6. Convert the Databases

- 6.1. Establish your Sablime environment to any generic in your instance.

```
$ . sablime <generic>
```

- 6.2. Invoke the conversion script

```
$ dbconvert
```

The conversion script will convert all the products of this Sablime instance. It will provide you with status output as it progresses.

7. Verify the Databases - Optional

Re-run the audits. Depending on what version of Sablime you converted *from*, you may find a few issues that were not reported in the pre-conversion audits. This is because the current audit programs do a more thorough job of validating the database.

For each product:

```
$ export sabPROD=<product_name>
$ dbcross > crossout.$sabPROD
$ dbdelta > deltaout.$sabPROD
$ dbxcross > xcross.$sabPROD
```

Review the three output files and repair any reported problems.

8. Restart the Databases

For each product in your instance, issue the "dbstart" command:

```
$ export sabPROD=<product_name>
$ dbstart
```

If your instance contains multiple products, run the above for each of them.

This is only necessary if the databases were stopped before you ran step 6. Otherwise **dbconvert** will have stopped and restarted the databases automatically.

9. Upgrade Web Sablime

9a) Upgrading from v6.0u1, v6.0u2, or v6.0u3

Upgrading from these recent versions of Web Sablime requires execution of just one script

```
$ web_update_to_v61 SABHOME.old
```

This script will update your Web Sablime information based on your previous setup.

Note: The "web_update_to_v61" script moves some files from your *SABHOME.old* directory into *SABHOME*. Among these are files that are owned by root, which is why they cannot just be copied. Note also that *SABHOME.old* and *SABHOME* must be on the same filesystem for this to work.

9b) Upgrading from v5.x or v6.0

If you are upgrading from v6.0 (no update), or from any pre-v6.0 version of Sablime, then you need to follow the instructions in the [Sablime v6.1u2 Web Sablime Installation and Upgrade Guide](#).

-
- * UNIX is a registered trademark of The Open Group.
 - * Linux is a registered trademark of Linus Torvalds.
 - * Sun, Sun Microsystems, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Sablime is a registered trademark of Lucent Technologies. Contents copyright © 2006 Lucent Technologies. Permission to photocopy in support of a licensed installation of Sablime is hereby granted.