solaris

UPGRADING and PATCHING with SOLARIS™ LIVE UPGRADE

> **Solaris™ 10** How To Guides

Jeff McMeekin, Solaris Marketing
Version 1.1 | *Last updated: 07/16/07*

Sun microsystems

About This Solaris How To Guide

This How To Guide demonstrates the use of Solaris Live Upgrade, a powerful tool for managing change, risk, and system availability on Solaris systems. This guide will cover:

- An overview of Solaris Live Upgrade for OS upgrades, patching, and as a mechanism for developers to install the latest OpenSolaris releases
- Planning issues associated with using Solaris Live Upgrade
- An example using Solaris Live Upgrade to upgrade from one Solaris release to a later one
- An example using Solaris Live Upgrade to patch a system

Contents

Solaris Live Upgrade How To Guide

Solaris Live Upgrade: Overview

Solaris Live Upgrade is a useful tool and strategy for minimizing downtime and risk when upgrading or patching systems. With the release of Solaris 10 8/07, patching Solaris Containers is now supported via Solaris Live Upgrade.

Operational Model

The easiest way to understand the advantages of Solaris Live Upgrade is to summarize how it works:

- Create a copy of the system environment (e.g., a copy of the root (/) file system
- Apply changes (whether OS upgrade, patches, or other changes) to that copy—instead of to the current system environment. The software that applies the changes to the copy runs as a background task on your production server
- Boot to the new environment and assess the impact of the changes. If you are satisfied with the new environment, you are done
- If you encountered problems with the new environment, reboot to the original environment

All tasks except the reboot can be accomplished on an operational production system; the impact on any running process is minimal. The combination of maximizing system availability when applying changes and minimizing risk by offering the ability to reboot to a known working state (your original environment) has made Solaris Live Upgrade a tool of choice for the datacenter since the Solaris 8 OS.

Upgrading

Upgrading a system with a later version of Solaris instead of doing a fresh install is a popular procedure for the simple reason that upgrading preserves all the effort initially spent in configuring the system. Sun offers two ways to upgrade:

- The standard upgrade process requires taking the system offline to do an upgrade and is a one-way operation; there is no easy way to revert to the original environment
- The other approach, using Solaris Live Upgrade, enables you to upgrade while greatly reducing the maintenance window associated with that activity and gives you the option to revert to the pre-upgrade environment

Patching

Solaris Live Upgrade is not limited to only OS upgrades; you can use it to manage downtime and risk when patching a system.

Because you are applying the patches to the inactive boot environment:

- Patching has only minimal impact on the currently running environment. The production applications continue to run as the patches get applied to the inactive environment
- You avoid taking the system down to single-user mode, which is the standard practice for applying the Recommended Patch Clusters
- You can boot the new patched environment, test the applications, and if you are not satisfied, reboot to the original environment
- With Solaris 10 8/07, patching Solaris Containers with Solaris Live Upgrade is supported. This can have a dramatic impact on decreasing downtime during patching

See the “Web Resources” section of this document for a pointer to the new Sun Blueprint on using Solaris Live Upgrade for patching mirrored disk systems.

Planning Considerations for Solaris Live Upgrade

Planning for the use of Solaris Live Upgrade is critical. Many datacenters cannot use Solaris Live Upgrade simply because they did not plan for its use. There are two main considerations when planning ahead: disk space and the N+2 rule.

Disk Space for the Second Boot Environment

Solaris Live Upgrade involves having two simultaneous boot environments, so enough disk space for both is a prerequisite. Either you need an extra disk, or one disk large enough to house both environments.

The safest approach is to order an extra disk. Since virtually all production systems mirror their disks, this means ordering 3 disks. One disk is for the root (/) file system, a second is for the mirror of the root file system and the third is for Solaris Live Upgrade. Another advantage of the third disk is that it can be used as a hot spare should there be a disk failure. Some customers don't want the trade-off between Solaris Live Upgrade and a hot spare, so they put a fourth disk in the system as the hot spare. The cost of hardware service calls and downtime is such that it may be less expensive to provision a system with a spare disk at order time than it is to replace disks in the field.

The other advantage of a three- or four-disk environment is that you aren't dependent on disk mirroring for safety. Some customers will use Solaris Live Upgrade as part of a back-up scenario whenever any changes are being contemplated. An operator error (which would of course be duplicated on the mirror) can still be worked around fairly quickly if a copy was created before any system changes were made.

A system with three disks (active boot environment, mirror of the active environment, and inactive boot environment) is generally considered the best practice if you want to use Solaris Live Upgrade, but it is possible to have only two disks (active and mirror) if one of them is large enough to hold both the active and inactive boot environments. If they aren't large enough, it's still possible by temporarily breaking the mirror to enable the second disk to be used by Solaris Live Upgrade. The wisdom of the latter approach depends on how critical the application is that's running on the system. Can the organization tolerate the risk of the active boot environment disk failing during the creation of the copy? Some organizations can tolerate this risk, others cannot.

The N+2 Rule

If you intend to use Solaris Live Upgrade for upgrading from one Solaris release to another, then keep in mind the N+2 rule: Sun supports and tests an upgrade up to two releases ahead of the release you are running; e.g., if you are running any Solaris 8 release you can upgrade to any Solaris 9 or Solaris 10 release.

Example: Upgrading to a New Release

For example, say you want to upgrade a Solaris 10 3/05 system to the Solaris 10 1/06 release. Solaris Live Upgrade has many capabilities but for the simplest situation, upgrading a system to a new Solaris release, it involves three simple commands:

- lucreate to create the copy
- luupgrade to upgrade the OS on the copy
- luactivate to choose the environment to boot

1. Begin by logging into the root account.
   You must apply a few key patches before using Solaris Live Upgrade. See Infodoc #72099 from SunSolve (sun.com/sunsolve) which describes the Solaris Live Upgrade SPARC or X86 patches required for each Solaris release.
2. Create a directory /var/tmp/lupatches and download the patches to that directory.
3. Patch the system:

   ```
   # cd /var/tmp/lupatches
   ```

   Then add each patch individually:

   ```
   # patchadd <patch_id>
   ```

   where <patch_id> is the patch number. The order in which the patches should be applied is specified in Infodoc 72099.
4. Next, if you are running on x86 hardware, you must reboot the system:

   ```
   # init 6
   ```

   After the system reboots, log back into the root account.

Before upgrading, you must install the Solaris Live Upgrade packages from the release to which you are upgrading. New capabilities are added to the upgrade tools, therefore installing the new packages from the target release is important. In this example, you will upgrade from Solaris 10 3/05 to Solaris 10 1/06, so you must get the Solaris Live Upgrade packages from the Solaris 10 1/06 DVD.

5. Install the latest Solaris Live Upgrade packages using the liveupgrade20 script. The script runs silently and installs the latest Solaris Live Upgrade packages. Run the following command without the -noconsole and -nodisplay options and you will see the GUI install tool.
   On SPARC systems:

   ```
   # cd /cdrom/cdrom0/s0/Solaris_10/Tools/Installers
   # ./liveupgrade20 -noconsole -nodisplay
   ```

   On x86/x64 systems:

   ```
   # cd /cdrom/cdrom0/Solaris_10/Tools/Installers
   # ./liveupgrade20 -noconsole -nodisplay
   ```

When you migrate from one major Solaris release to another major release (for example, Solaris 8 to Solaris 10) there may be additional package requirements. The best strategy is to review Chapter 7, "Solaris Live Upgrade (Planning)" in the "Solaris 10 Installation Guide: Solaris Live Upgrade and Upgrade Planning" document available from docs.sun.com.

6. Run the lucreate command to create the copy of the active boot environment.

You are now ready to make a copy of the root (/) file system. Assume you have the situation as pictured in Figure 1; you have two disks and you have created a partition on the second disk to be the same size as the root (/) partition on the first disk.

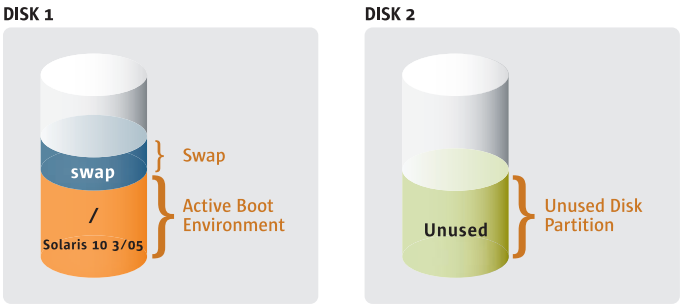

Figure 1—Before running the lucreate command

The object of this step is to create a copy of the current root (/) file system so that you have the situation pictured in Figure 2.

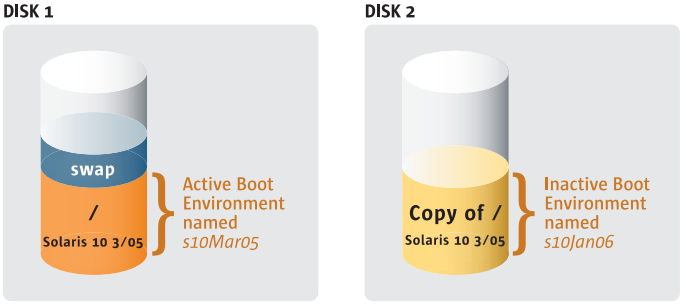

Figure 2—After running the lucreate command

You will need to name both the current (active) boot environment and the copy (the inactive boot environment). The partition you will use for the copy must not appear in use in /etc/vfstab—in other words it should not have a file system on it. In this case, the active environment has been named "s10Mar05". Because the goal of this exercise is to upgrade to Solaris 10 1/06, the new inactive boot environment has been named "s10Jan06". You will also need to specify that you want to make a copy of the root (/) file system and where that new partition is to be located (c0t1d0s0 in this example). Also note that the file system is of type UFS. Note how the last three pieces of information are concatenated for the '-m' argument.

Create a new (inactive) boot environment:

```
# lucreate -c s10Mar05  -n s10Jan06 -m /:c0t1d0s0:ufs
```

This command will generate output similar to the following. The time to complete varies depending on the speed of the system and its disks.

```
Discovering physical storage devices.
Discovering logical storage devices.
Cross referencing storage devices with boot environment configurations.
Determining types of file systems supported.
Validating file system requests.
The device name <c0t1d0s0> expands to device path </dev/dsk/c0t1d0s0>.
Preparing logical storage devices.
Preparing physical storage devices.
Configuring physical storage devices.
Configuring logical storage devices.
Analyzing system configuration.
No name for current boot environment.
Current boot environment is named <s10Mar05>.
Creating initial configuration for primary boot environment <s10Mar05>.
The device </dev/dsk/c0t0d0s0> is not a root device for any boot
environment.
PBE configuration successful: PBE name <s10Mar05> PBE Boot Device
</dev/dsk/c0t0d0s0>.
Comparing source boot environment <s10Mar05> file systems with the file
system(s) you specified for the new boot environment. Determining which
file systems should be in the new boot environment.
Updating boot environment description database on all BEs.
Searching /dev for possible boot environment filesystem devices.

Updating system configuration files.
The device </dev/dsk/c0t1d0s0> is not a root device for any boot
environment.
Creating configuration for boot environment <s10Jan06>.
Source boot environment is <s10Mar05>.
Creating boot environment <s10Jan06>.
Creating file systems on boot environment <s10Jan06>.
Creating <ufs> file system for </> on </dev/dsk/c0t1d0s0>.
Mounting file systems for boot environment <s10Jan06>.
Calculating required sizes of file systems for boot environment <s10Jan06>.
Populating file systems on boot environment <s10Jan06>.
Checking selection integrity.
Integrity check OK.
Populating contents of mount point </>.
Copying.
Creating shared file system mount points.
Creating compare databases for boot environment <s10Jan06>.
Creating compare database for file system </>.
Updating compare databases on boot environment <s10Jan06>.
Making boot environment <s10Jan06> bootable.
Population of boot environment <s10Jan06> successful.
Creation of boot environment <s10Jan06> successful.
```

You might find it useful, particularly if you encounter problems running the Solaris Live Upgrade commands, to use the lustatus utility to see the state of the boot environment.

After the new boot environment is created, you can then begin the upgrade procedure.

In this example, you will upgrade from the Solaris 10 1/06 DVD. After completing this step, you will have the situation depicted in Figure 3.

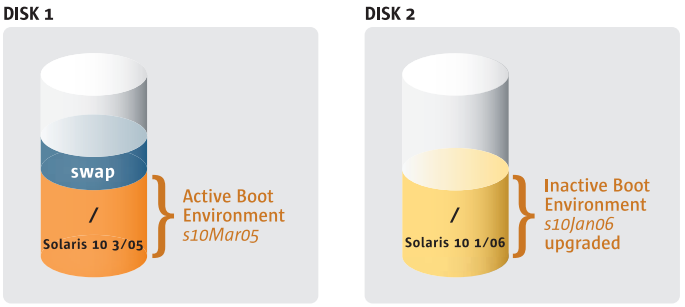

Figure 3—After running the luupgrade command

7. To upgrade to a new Solaris release, you will use the luupgrade command with the -u option. The -s option identifies the path to the media. Due to slightly different file system organization on X86 installation DVDs than on SPARC DVDs, the location -s option varies.

   For Solaris running on SPARC hardware, a typical command line would be:

```
# luupgrade  -u  -n s10Jan06   -s /cdrom/cdrom0/s0
```

   For Solaris running on x86 platforms:

```
# luupgrade  -u  -n s10Jan06   -s /cdrom/cdrom0
```

   This command will generate output similar to the following:

```
Validating the contents of the media </cdrom/cdrom0/s0>.
The media is a standard Solaris media.
The media contains an operating system upgrade image.
The media contains <Solaris> version <10>.
Constructing upgrade profile to use.
Locating the operating system upgrade program.
Checking for existence of previously scheduled Solaris Live Upgrade
requests.
Creating upgrade profile for BE <s10Jan06>.
Determining packages to install or upgrade for BE <s10Jan06>.
Performing the operating system upgrade of the BE <s10Jan06>.
CAUTION: Interrupting this process may leave the boot environment unstable
or unbootable.
```

```
Upgrading Solaris: 6% completed
Upgrading Solaris: 100% completed
Installation of the packages from this media is complete.
Updating package information on boot environment <s10Jan06>.
Package information successfully updated on boot environment <s10Jan06>.
Adding operating system patches to the BE <s10Jan06>.
The operating system patch installation is complete.
INFORMATION: The file </var/sadm/system/logs/upgrade_log> on boot
environment <s10Jan06> contains a log of the upgrade operation.
INFORMATION: The file </var/sadm/system/data/upgrade_cleanup> on boot
environment <s10Jan06> contains a log of cleanup operations required.
INFORMATION: Review the files listed above. Remember that all of the files
are located on boot environment <s10Jan06>. Before you activate boot
environment <s10Jan06>, determine if any additional system maintenance is
required or if additional media of the software distribution must be
installed.
The Solaris upgrade of the boot environment <s10Jan06> is complete.
```

At this point, you have created a new (inactive) boot environment and upgraded it to the Solaris 10 1/06 release. In the next two steps, you will first tell Solaris to use the new environment the next time you boot, and then you will reboot the system so that it comes up running the new environment (as shown in Figure 4).

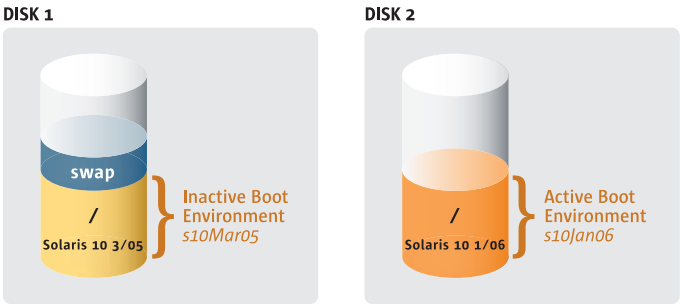

Figure 4—After running the luactivate command and rebooting

8. Indicate which boot environment is to be used the next time you reboot.

```
# luactivate s10Jan06
```

where s10Jan06 is the boot environment you want to make active.

On a running system, some files might change during or after the lucreate/luupgrade process. You might want those changes reflected in the new boot environment in order to synchronize the two systems. To address this, there is a synchronization mechanism that the luactivate command will automatically call. See the synclist(4) man page for more information.

The luactivate can also be used to revert to the previous boot environment. In most cases, this is very straightforward, but due to the adoption on x86 platforms of GRUB (the new bootloader for Solaris 10 1/06 and subsequent Solaris releases)

there are some issues when reverting from a GRUB system to a pre-GRUB release. These are all explained in the Solaris Live Upgrade documentation referenced below.

The output of the luactivate command will look like the following:

```
**********************************************************************

The target boot environment has been activated. It will be used when you
reboot. NOTE: You MUST NOT USE the reboot, halt, or uadmin commands. You
MUST USE either the init or the shutdown command when you reboot. If you
do not use either init or shutdown, the system will not boot using the
target BE.

**********************************************************************

In case of a failure while booting to the target BE, the following process
needs to be followed to fallback to the currently working boot environment:

1. Enter the PROM monitor (ok prompt).

2. Change the boot device back to the original boot environment by typing:

     setenv boot-device /pci@1f,0/pci@1,1/ide@3/disk@0,0:a

3. Boot to the original boot environment by typing:

     boot

**********************************************************************

Activation of boot environment <s10Jan06> successful.
```

Note the warning in the above about the proper command to use to reboot. You must use “init” or “shutdown.”

9. Reboot the system

```
# init 6
```

You have now succeeded in upgrading to the Solaris 10 1/06 release. Note that the reboot is the only downtime experienced using this upgrade method—a far different situation than if you had done a standard install or standard upgrade.

One final note—make sure that you dispose of a boot environment properly when you are finished with it. The proper way is to use the ludelete command. If you destroy the boot environment through some other mechanism (e.g. newfs, or rm -rf) you risk having a system that will not boot.

Example: Patching a System Using Solaris Live Upgrade

One of the most common uses of Solaris Live Upgrade is to minimize downtime for patching. Upgrading to a newer Solaris release is a relatively infrequent activity in comparison to patching a system. Depending on the length of the maintenance window and the need to minimize downtime, patching an alternate boot environment may be an advantageous operations choice. The steps are:

- Create a new boot environment
- Patch the new boot environment
- Boot from the new boot environment
- Assess if the result of the changes is acceptable

As in the case of upgrading, Solaris Live Upgrade has the advantage of applying the patches in the background with little impact on what is running on the system.

To begin, follow the steps 1-6 in the first example and optionally step 7 (if you wish to upgrade before patching). Now you are ready to apply the Recommended Patch Cluster to the system.

7a. Obtain the patches:
Access the Recommended Patch Cluster for Solaris 10 from sun.com/sunsolve.

This step typically involves downloading the file 10_Recommended.zip (or 10_x86_Recommended.zip). For this example, assume you have downloaded it to /var/tmp. Then use the unzip command to uncompress the downloaded file to create the directory 10_Recommended containing all the patches. The file 10_Recommended/CLUSTER_README will tell you to take the system to single user mode, and then use the install_cluster script. In our case, since you are patching the inactive boot environment, you do not have to take the system to single user mode. You will also employ a slightly different procedure to apply the patches; you do not use the install_cluster script.

7b. Make sure you are in the patch directory:

```
# cd /var/tmp/10_Recommended
```

7c. Patch the Inactive Boot Environment:

```
# luupgrade -n s10Jan06 -s /var/tmp/10_Recommended -t `cat patch_order`
```

In the above you identify which boot environment to patch, (s10Jan06"), where the patches are located (the -s option and path argument), and patches to apply (the -t option followed by the patch numbers). Note the argument to the -t option is placed in backquotes meaning the expression will be evaluated by the shell before sending the results (the list of patches) to the luupgrade command.

To finish, follow steps 8 and 9 in the first example.

The above is a very simple view of how to use Solaris Live Upgrade for patching. A more realistic treatment—using Solaris Live Upgrade to patch in an enterprise mirrored-disk environment—is covered in a Sun Blueprints publication referenced below in the Web Resources section.

The Solaris Live Upgrade tools are engineered to be backward-compatible to Solaris 8. This offers some operational advantages. The use of Live Upgrade for patching systems with Containers was not supported on Solaris 10 releases prior to 8/07. Suppose you have a Solaris 10 11/06 system running Containers, you can use the Solaris 10 8/07 Live Upgrade tools to patch your system without having to upgrade to Solaris 10 8/07. As always though, you must patch the target using the information in Infodoc 72099 before attempting to use the 8/07 tools on an 11/06 system.

Patching Considerations

Behind the scenes, Solaris Live Upgrade is simply using the patchadd command. You could accomplish the same patching of the inactive boot environment by mounting it and using patchadd with the -R option. If you want to understand the behavior of patching using Solaris Live Upgrade, familiarize yourself with the patchadd command.

The Solaris 10 patchadd command is smart enough to order patches correctly, but Solaris 9 and earlier releases require patches to be in dependency order. Sun uses a command line similar to the above as part of the standard testing, so it makes sense to use a similar approach when you use Solaris Live Upgrade to patch—no matter what Solaris release you are patching.

Third-party patches may not support being applied through Solaris Live Upgrade. All Sun patches conform to the requirement that pre- and post-install scripts never modify the running system when the target is an inactive boot environment. Furthermore, testing the application of Recommended Patches via Solaris Live Upgrade is part of Sun's standard test procedures. However, Sun cannot guarantee that all third-party patches are equally well-behaved. Depending on your organization's reliance on non-Sun patches, you may need to verify that a third-party patch does not contain a script that attempts to modify the current environment when you intend only to patch an inactive boot environment.

With the release of Solaris 10 1/06, Sun includes a new patching tool, Sun Connection, that analyzes your system and then applies the appropriate patches. The September 2006 release 1.0.8 of this tool set provides various options for working with Solaris Live Upgrade. The Solaris 10 8/07 smpatch(1M) man page has details on the options.

Summary

Solaris Live Upgrade is an enterprise-class tool, and these examples have described only a small subset of its possibilities. Solaris Live Upgrade offers many different ways to control the creation of a new boot environment; for example, there are other options besides making a complete copy of the current environment. Solaris Live Upgrade integrates with Solaris Volume Manager and, with some extra scripting, with Veritas Volume Manager 4.1 or later (contact Symantec for details). Solaris Live Upgrade can be used to add packages as well as patches, and of course remove packages and patches. It can also be used to load a Solaris Flash Archive image, a high-performance approach to quickly building a system. For more details see the “Additional Resources” section below.

In closing, Solaris Live Upgrade offers an excellent value proposition for enterprises—the ability to make a copy of a system, make changes to the copy or the original, and then decide which will be the new boot environment. The only downtime for such an operation is the actual reboot. All the other operations can take place in the background of a running production system. The most important factor in using Solaris Live Upgrade is to plan ahead by configuring your systems properly.

For More information

We have made brief reference to the lucreate(1M), lustatus(1M), luupgrade(1M), and luactivate(1M) commands in this document. You might also review other commands (see, for example, the See Also section of the lucreate(1M) man page). The man pages reference lu(1m), a simple tool for managing Solaris Live Upgrades. However, this GUI-like tool is no longer current with the capabilities of the rest of the Solaris Live Upgrade command-line tools, so we do not recommend its use. The lu command will likely be dropped in a future Solaris release.

Web Resources	
The man pages for Solaris 10 can be found in the Solaris 10 Reference Manual Collection	docs.sun.com/app/docs/coll/40.10
Solaris 10 Installation Guide: Solaris Live Upgrade and Upgrade Planning	docs.sun.com/app/docs/coll/1236.1
An excellent Sun Blueprints document on patching using Solaris Live Upgrade, “Patching Mirrored Systems with the Solaris Live Upgrade Software”	sun.com/blueprints/0607/820-2188.html
A document specific to using Solaris Live Upgrade with Solaris Express Developer Editions	developers.sun.com/sxde/upgrade_guide.jsp
Search the Sun forums. There are two forums where Solaris Live Upgrade questions tend to appear: 1) Click on Solaris tab, then select "Installation" 2) Click on Administration tab then select "Talk to the Sysop"	forum.java.sun.com/

sun.com/solaris

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA Phone 1-650-960-1300 or 1-800-555-9SUN Web sun.com