

# Solaris™ Operating Environment Deployment Mechanisms in Replicated and High Availability Environments

Jay Daliparthi and James Falkner

Solaris Software  
Sun Microsystems, Inc.

## Abstract

This article will briefly introduce some of the latest technologies that will help in rapid Solaris™ Operating Environment (OE) installation or software upgrades in high availability environments. Web Start Flash provides a mechanism to create a Flash archive, a system snapshot of the Solaris 8 OE and all other installed software from a master system. This can then be replicated onto multiple machines (clones). These Flash archives can be deployed by media via CD and Tape or over the network via HTTP or NFS using custom Solaris JumpStart™ software or the Solaris Web Start software. Solaris Live Upgrade 2.0 software enables Solaris systems to continue to run uninterrupted while the system administrator installs a Flash archive or upgrades to a new Solaris 8 OE. Solaris Live Upgrade software allows creating, managing, upgrading, comparing, and activating multiple boot environments. Live Upgrade also supports fallback in case of upgrade failure. Live Upgrade reduces system downtime and risk associated with an operating system upgrade or patch maintenance. These tools provide system administrators, the ability to have consistent, controlled, and re-creatable Solaris OE installations and upgrades.

## Web Start Flash Technology Overview

The Web Start Flash technology provides a mechanism to create, a snapshot image of a complete system including the Solaris Operating Environment, the applications, and system configuration into a new Flash archive format. This reference system or master system, on which the archive is created will be referred to as a master system. Using the Flash archive, administrators can replicate reference server configurations onto multiple (clone, or client) machines.

Solaris OE installation using Web Start Flash is a three-step process. First you need to install a master machine with the configuration that you would like to have on your clone machines. For example, if you want to install ten cache servers, then you need to first install a master system and configure it to work as a cache server. To install a master system, you can use any of the Solaris installation methods, such as a Solaris JumpStart™ software installation or interactive installation from a CD to install the Solaris OE. After installing the Solaris OE, any software can be added or deleted and configurations can be modified as necessary. Secondly, you create a flash archive using the *flarcreate* (1M) command. It is important to note

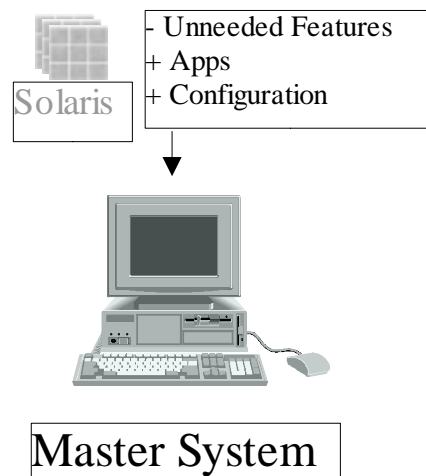
that the flash archive contains copy of all software and the files on the master system unless certain files are specifically excluded. For example, if a flash archive is created from a web server, then that archive contains all the software installed on the master web server. John Howard (2001) in his article on "Introduction to Solaris Web Start Flash" explains, that "Flash is a powerful tool for deploying complex server configurations. Complex system architecture only needs to be implemented and configured once and then a Flash archive created; all subsequent system installations from that Flash archive will be identical. For systems such as database servers, the Flash archive should be created before the system goes into the production".

These Flash archives can then be saved to a disk or tape or copied to any file system or media such as an NFS or HTTP server, Tape, CD, or DVD. The Flash archives can be installed over the network using custom Solaris JumpStart or the Solaris™ Web Start software for interactive installation.

The three steps used to create, archive, and deploy a Flash archive are illustrated below. In the first step, a machine is loaded with the desired software stack. This typically includes the Solaris OE, and any 3<sup>rd</sup>-party applications necessary to complete the software load.

This first step is the most time-consuming step, as it requires the administrator to install one or more software packages, and then configure the system to perform as it would in the field. However, this step is the most critical - once the software load is defined and archived, it is considered to be stable and complete. It is important to spend resources to achieve that goal.

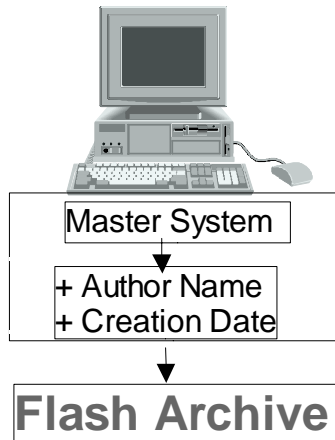
Once a stable and complete system is configured, it need not be configured again - the modifications made in this step are archived as part of the Flash archive, and remain frozen until extracted at a later date. When the archive is later extracted to multiple machines, the resource savings become apparent. For example, a horizontally-scaled web server is a typical application of Web Start Flash. In this scenario, a machine is installed with the Solaris Operating Environment. Next, various entries in `/etc/inetd.conf` are removed to harden the machine against any security breaches. Then a web server such as Apache is installed, and configured to provide optimum performance and security. Finally, the machine is archived using `flarcreate`. This archive can then be deployed across tens or even hundreds of machines to provide a quick, virtually bulletproof farm of web servers. Each machine need not be individually configured - the required configuration is taken care of "at the factory" when the master machine is configured.



### *Step 1. Configuring the Master System*

Once the machine has been configured to satisfaction, a snapshot is taken. This second step assumes the master system is

complete and functional. In this step, the content of the master system is archived into a Flash archive.

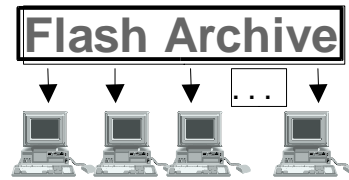


### Step 2. Creating a Flash Archive

The second step is done using the *flarcreate* (1M) command. The archival process includes not only the software on the system, but any identifying information (such as the size, date of creation, or author's name) that can be used to later identify the archive. This process can be time-consuming depending on the size of the system to be archived. However, it is completely hands-off. The output of this step is the Flash archive. The archive can then be stored on a network server, or on local media such as a CD or magnetic tape. Typical sizes for an archive containing the Solaris Operating Environment range from 100mb up to 1gb. Compression (**flarcreate -c**) can be used to minimize the resulting footprint.

The final step is deploying the archive. Several methods of deployment can be used. Traditional Solaris Installation technologies can be used, such as Solaris Interactive Installation or Custom JumpStart software, using the new **archive\_location** keyword. In addition, Live Upgrade (discussed later

in this paper) can be used to deploy Flash archives.



### Step 3. Deploying a Flash Archive

Since the majority of configuration has already been done, the performance of this step is primarily limited by the available bandwidth between the location of the stored archive(s) and clone machines. The overhead associated with a traditional package-based install is eliminated using Flash archives. It is important to note that not everything on the clone machines is identical - for example, each clone machine will have a unique IP address. These machine-specific configurations are minimal and are accomplished using the same techniques found in traditional installations.

There are other important things to note in creating Flash archives. The master and the clone machines preferably should have the same kernel architecture. For example, Flash archives created from a master machine that has a sun4u architecture can only be used to install other servers with a sun4u architecture. However, postinstall JumpStart scripts can be written for any necessary modifications to configuration files such as **/etc/nsswitch.conf**. These modifications will allow for any differences in kernel architecture or device differences between the master machine and the installation client (Howard, 2001). Separate Flash archives need to be created for both SPARC™

and IA machines if you need to deploy on both the platforms.

Installation using Web Start Flash is faster compared to traditional package-based installation, where each package is installed individually. As Howard (2001) observes, “A traditional JumpStart software installation using packages may become time consuming as *pkgadd* (1M) needs to update the installed package database as well as relocate files”.

Web Start Flash technology can also be used to store existing server configurations in the archive format as a backup. In case of a failure, software can be reinstalled rapidly using Flash archives. Web Start Flash is also ideal for the replication and deployment of server farms. Web Start Flash technology provides the ability to have consistent, controlled, reliable, and recreatable Solaris OE installations in mission critical operations such as those in Data Centers.

Web Start Flash technology is included in the Solaris 8 4/01 release. Additional information on Web Start Flash is also available from the Solaris 8 Installation Guide, Sun Microsystems; Introduction to Solaris Web Start Flash by John S. Howard in Sys Admin, Oct. 2001 edition; and [http:// www.sun.com/solaris/](http://www.sun.com/solaris/).

## **Solaris Live Upgrade 2.0 Software Overview**

The process of performing an upgrade to the Solaris 8 OE and other system software or other routine patch maintenance can impact availability of systems for normal operations. Systems running mission critical operations, such

as those in Data Centers, can't afford system down time for patch application or software upgrade. Solaris Live Upgrade software will allow upgrading Solaris 2.6, Solaris 7 or Solaris 8 OE or general maintenance on an alternative boot environment without bringing the system down. Solaris Live Upgrade software also provides a mechanism to fall back to the previous state in case of a problem with the software upgrade or patch application.

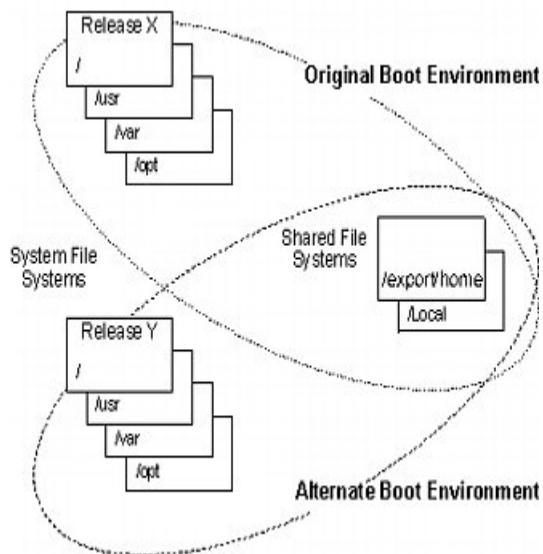
Solaris Live Upgrade software provides a framework to create, manage, manipulate, and activate multiple boot environments (BE) on the same system. Live Upgrade software provides support to advanced file system operations such as splitting, merging, or sharing file systems between BEs. It allows migrating file systems from a storage device to another device. Live Upgrade works with mirrors and other disk layouts. Live Upgrade enables you to maintain multiple installations of software packages including the Solaris OE, which will share user data file systems. These inactive boot environments can be used as a backup for fault tolerance.

Solaris Live Upgrade 2.0 8/01 version has integrated to work with Web Start Flash technology. This functionality provides a mechanism to install the Flash archive on an inactive boot environment, while the active boot environment is fully functional and unaffected by the installation. When satisfied, the administrator can migrate to the other environment by rebooting to that environment.

Solaris Live Upgrade software can be used to manage patch updates. Patches can be applied to an inactive boot

environment and, after they have been tested, be rolled out to the active boot environment. This process helps in reducing the system downtime. In addition, it makes it less difficult to fall back to a previous state with a reboot. It helps in reducing service outages associated with the normal test and evaluation process required when introducing a new version of a patch of an operating environment.

Live Upgrade supports software migration and fallback to a previous active boot environment in case of failure. Live Upgrade provides a mechanism to analyze the reasons for failure. It also synchronizes the differences between the active and inactive boot environments. This feature helps to prevent any losses while the system is in the upgrade or fallback process.



**Figure 1. Active and Inactive (Alternate) Boot Environments** (Source: Solaris Live Upgrade 2.0 Guide).

## Boot Environment Overview

A boot environment is a grouping of file systems and their associated mount points. These BEs can be created on the same disk or on separate disks, however a single root “/” file system is the recommended layout for Solaris OE (Howard, 2000). The active boot environment is the BE that is currently booted and active, and all others are considered inactive or alternate BEs.

## Solaris Live Upgrade Software Process Overview

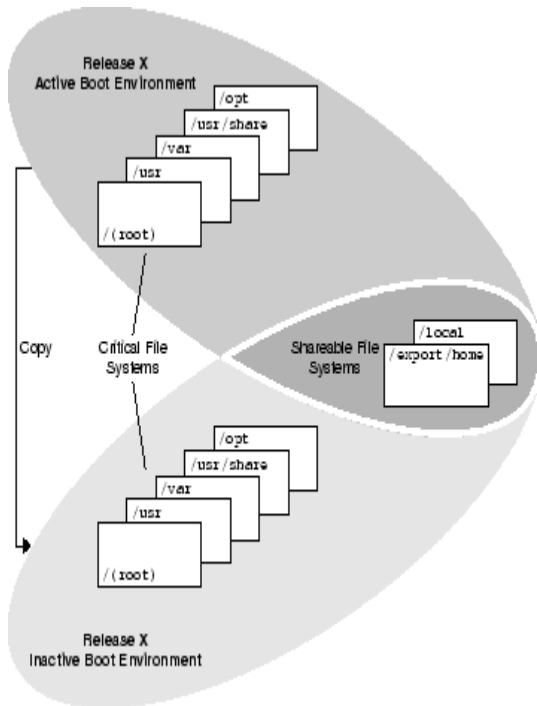
Solaris Live Upgrade software provides functionality for software upgrade, system maintenance, and installation of a Flash archive. Live Upgrade has the following operational steps:

1. Creating a BE
2. Upgrading an inactive BE (or)
3. Installing Web Start Flash archive on inactive BE
4. Activating an inactive BE
5. Falling back to original BE

## Creating a Boot Environment Overview

Live Upgrade provides mechanisms to distinguish between critical file systems and shareable file systems. Critical file systems are those required by Solaris OE and have separate device entries in the **vfstab** of the active and inactive BEs. For example, **root (/)** **/usr**, **/var** or **/opt**. These file systems are always copied from the active to the inactive BE. Shareable file systems are user defined, for example, **/export** that contain the same mount point and device entry in the **vfstab** in both the active and inactive BEs (Figure 2). After

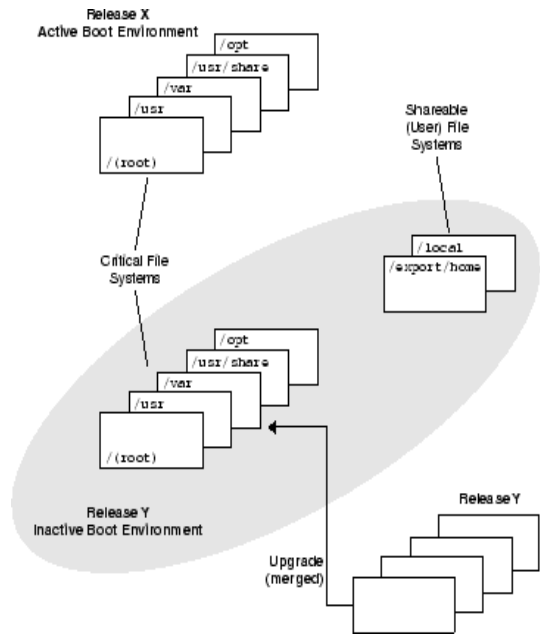
defining an alternative BE using the **lucreate (1M)** command, critical file systems will be copied over (Source: Solaris 8 Installation Guide, Sun Microsystems). Please consult the Solaris 8 Installation Guide for detailed instructions.



**Figure 2. Creating an Inactive Boot Environment** (Source: Solaris Live Upgrade 2.0 Guide)

### Upgrading A Boot Environment

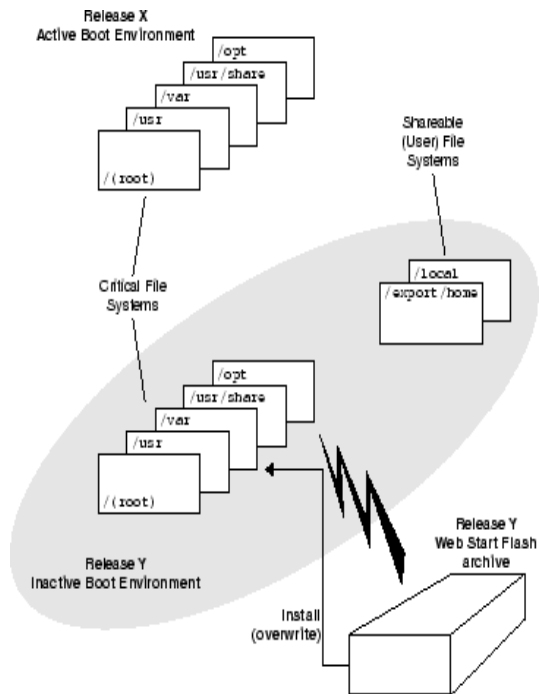
Once an inactive BE has been created it remains unchanged and is identical to the active BE. Solaris Live Upgrade software provides a mechanism to perform a software upgrade or patch application on the inactive boot environment with the **luupgrade (1M)** command. The upgrade process will not affect any files in the active boot environment. Figure 3 shows an upgrade of an inactive boot environment.



**Figure 3. Upgrading an Inactive Boot Environment** (Source: Solaris Live Upgrade 2.0 Guide).

### Installing Web Start Flash Archive on Inactive BE

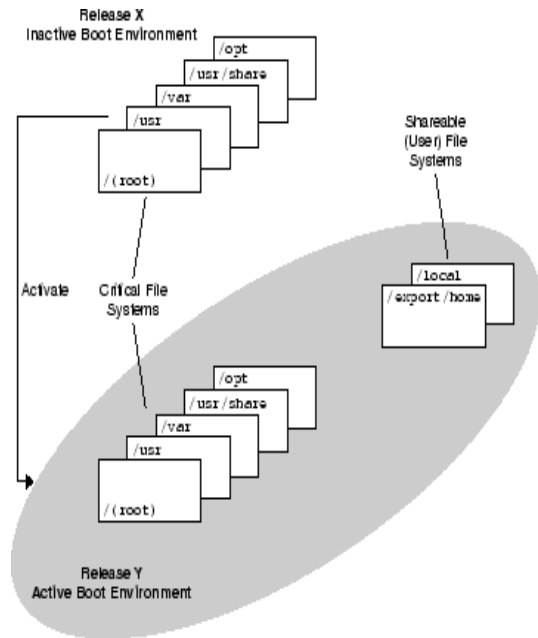
The latest release of Live Upgrade 2.0 8/01 version provides a mechanism to install Web Start Flash archive. For example, a Flash archive containing the Solaris 8 7/01 OE and other software can be installed on an inactive BE while the Solaris 2.6 OE is up and live on the active boot environment. When you install the Web Start Flash archive on a system, all the files in the archive are copied to that system and a new software version is installed without affecting the active BE. But, unlike an upgrade which merges files, installing a Web Start Flash archive overwrites the files as an initial installation would. Figure 4 shows an installation of a Web Start Flash archive on an inactive boot environment. Review Solaris 8 Advanced Installation Guide for detailed information on Web Start Flash and Live Upgrade.



**Figure 4. Installing a Web Start Flash Archive** (Source: Solaris Live Upgrade 2.0 Guide).

### Activating BE

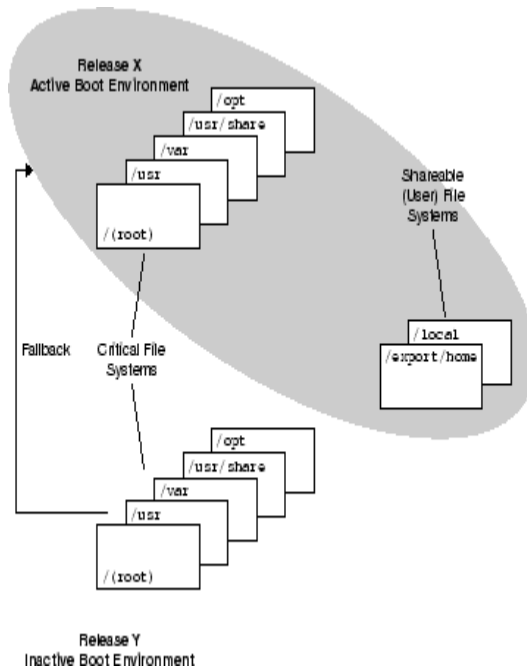
Solaris Live Upgrade software provides a mechanism to activate any inactive BE. Once the upgrade process is complete, the inactive BE can then be activated by the `luactivate` command. Now the target BE will be activated and will become the active BE after reboot. Solaris Live Upgrade performs file synchronization when activating the inactive BE. Figure 4 shows a switch after activation of an inactive boot environment.



**Figure 5. Activating Alternative Boot Environment** (Source: Solaris Live Upgrade 2.0 Guide).

### Fall Back Mechanism

Solaris Live Upgrade software provides a mechanism to fall back if a failure occurs or if you are not satisfied with the upgrade process. It provides functionality to immediately fall back to the original BE with an activation using the `luactivate (1M)` command and then rebooting. The upgraded BE will be saved and Live Upgrade provides functionality to analyze the failure. The process to boot from media is first mount the root file system, then run the `luactivate` on the target fall back BE, and finally reboot (Source: Solaris Live Upgrade 2.0 Guide). Figure 6 shows the switch that is made when you reboot to fall back. For detailed information review Solaris Advanced Installation Guide.



**Figure 6. Fall Back to Original Boot Environment** (Source: Solaris Live Upgrade 2.0 Guide).

## Summary

This paper has briefly introduced Web Start Flash and Live Upgrade technologies. Web Start Flash technology provides a mechanism for rapid deployment of servers and reduces the complexity of software configurations. Live Upgrade software provides the framework to upgrade and maintain systems while the servers are up and live. It helps most mission critical and data center operations as they cannot afford lengthy system down time for software upgrade or patch application. The most important benefit of Solaris Live Upgrade software is the ability to fall back to previous environment in case of upgrade or installation failure. Web Start Flash technology is available with Solaris 8 OE 4/01 and Solaris Live Upgrade 2.0 8/01 software is available for download from: <http://www.sun.com/solaris/liveupgrade/>.

## References

1. Introduction to Solaris™ Web Start Flash by John. S. Howard. Sys Admin. Oct.2001.
2. An Introduction to Live Upgrade. John S. Howard. Sun BluePrints™ Online-July 2000, Sun Microsystems.
3. Solaris 8 Installation Guide, Sun Microsystems.
4. Sun Best Practices and Blueprints at <http://www.sun.com/blueprints/>.
5. JumpStart Technology: Effective use in the Solaris Operating Environment by John. S. Howard and Alex Noordergraaf. ISBN 0-13-062154-4

## Acknowledgments

The authors would like to thank Gary Gere, Matt Williamson, Eric Nielsen, Derek Maxwell, Sandeep Bhalerao, Evert Hoogendoorn, and John S. Howard of Sun Microsystems for reviewing this article.

## Trademarks

2001 Sun Microsystems, Inc. All rights reserved.

Sun, Sun Microsystems, Solaris, Solaris JumpStart, Sun BluePrints, and the Sun Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the United States and othe Countries. Products bearing SPARC trademarks are based upon an architecuture developed by Sun Microsystems, Inc.