

Oracle® Solaris 10 9/10 Installation Guide: Solaris Flash Archives (Creation and Installation)

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related software documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Copyright © 2010, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. UNIX est une marque déposée concédée sous licence par X/Open Company, Ltd.

Contents

Preface	11
1 Solaris Flash (Overview)	15
Solaris Flash Introduction	15
What's New in the Oracle Solaris 10 9/10 Release	15
What's New in the Solaris 10 10/09 Release	17
Installing Clone Systems With an Initial Installation	18
Updating Clone Systems With a Solaris Flash Differential Archive	19
2 Solaris Flash (Planning)	23
Planning Your Solaris Flash Installation	23
Designing an Initial Installation of the Master System	24
Planning the Creation of a Solaris Flash Archive	28
Planning the Installation of Solaris Flash Archives	33
3 Creating Solaris Flash Archives (Tasks)	35
Task Map: Creating Solaris Flash Archives	35
Installing the Master System	36
▼ To Install the Master System for an Initial Installation	37
Creating Customization Scripts	37
▼ To Create a Precreation Script	37
Using a Precreation Script to Create a User-Defined Archive Section	39
▼ To Create a Predeployment Script	39
▼ To Create a Postdeployment Script	40
▼ To Create a Reboot Script	41
Creating a Solaris Flash Archive	41
▼ To Create a Solaris Flash Archive for an Initial Installation	42

Creating a Solaris Flash Archive (Examples)	43
▼ To Create a Solaris Flash Differential Archive With an Updated Master Image	48
▼ To Create a Solaris Flash Differential Archive by Using Solaris Live Upgrade	51
4 Installing and Administering Solaris Flash Archives (Tasks)	55
Installing a Solaris Flash Archive With the Solaris Installation Program	55
▼ Installing a Solaris Flash Archive	56
References to Procedures for Installing Solaris Flash Archives	57
Administering Solaris Flash Archives	58
Splitting a Solaris Flash Archive	58
Merging Solaris Flash Archives	59
Extracting Information From an Archive	61
5 Creating and Using a Disaster Recovery Image	63
Recovery Image Procedures	63
▼ Creating and Saving a FLAR Image	63
▼ Recovering the System Image From a FLAR Image	65
Additional Resources	67
6 Solaris Flash (Reference)	69
Solaris Flash Archive Section Descriptions	69
Solaris Flash Keywords	71
General Keywords	71
Keywords for the Archive Identification Section	71
User-Defined Section Keywords	75
Solaris Flash <code>flar</code> Command	75
<code>flar</code> Command	76
Glossary	81
Index	93

Figures

FIGURE 1-1	Solaris Flash Initial Installation	19
FIGURE 1-2	Solaris Flash Update	21

Tables

TABLE 1-1	Auto Registration Impact	16
TABLE 2-1	Limitations When Creating and Installing a Solaris Flash Archive	23
TABLE 2-2	Flash Archive Sections	32
TABLE 3-1	Task Map: Creating a Solaris Flash Archive to Install With an Initial Installation	35
TABLE 3-2	Task Map: Creating a Solaris Flash Archive to Update a Clone System	36
TABLE 6-1	Flash Archive Sections	69
TABLE 6-2	Values for <code>section_begin</code> and <code>section_end</code> Keywords	71
TABLE 6-3	Archive Identification Section Keywords: General Keywords	72
TABLE 6-4	Archive Identification Section Keywords: Contents of Archive Files Section ...	72
TABLE 6-5	Archive Identification Section Keywords: User Describes the Archive	73
TABLE 6-6	Archive Identification Section Keywords: Software Describes the Archive	75
TABLE 6-7	Command-Line Options for the <code>flar</code> Command	77

Examples

EXAMPLE 3-1	Excerpts From a Precreation Script	38
EXAMPLE 3-2	Precreation Script	39
EXAMPLE 3-3	Predeployment Script	40
EXAMPLE 3-4	Postdeployment Script	41
EXAMPLE 3-5	Creating a reboot Script	41
EXAMPLE 3-6	Creating an Exact Duplicate Archive	43
EXAMPLE 3-7	Creating an Archive That Contains Large Files	44
EXAMPLE 3-8	Creating an Archive From an Alternate root (/) File System	45
EXAMPLE 3-9	Creating an Archive and Adding Keywords to Describe the Archive	45
EXAMPLE 3-10	Creating an Archive and Excluding and Including Files and Directories	46
EXAMPLE 3-11	Creating an Archive Excluding and Including Files and Directories by Using Lists	46
EXAMPLE 3-12	Creating an Archive Excluding Files and Directories by Using a List and Restoring a Directory	47
EXAMPLE 3-13	Creating an Archive Excluding and Including Files and Directories by Using a List With the -z Option	47
EXAMPLE 3-14	Creating a Differential Archive With the New Master Image on the Master System	50
EXAMPLE 3-15	Creating a Differential Archive With the Images Stored on an Inactive Boot Environment	50
EXAMPLE 3-16	Creating a Differential Archive by Using Solaris Live Upgrade	52
EXAMPLE 4-1	Splitting an Archive	59
EXAMPLE 4-2	Merging a Solaris Flash Archive	60
EXAMPLE 4-3	Merging a Solaris Flash Archive and Adding a User-Defined Section	60
EXAMPLE 4-4	Listing Files in an Archive Section	61

Preface

This book provides planning information and instructions for creating Solaris Flash archives and using Solaris Flash archives to install the Solaris Operating System (OS) on multiple systems.

This book does not include instructions about how to set up system hardware or other peripherals. This book is only for installing UFS file systems. Solaris Flash archives cannot be used for Solaris ZFS installations.

Note – This Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the [Solaris OS: Hardware Compatibility Lists](http://www.sun.com/bigadmin/hcl) (<http://www.sun.com/bigadmin/hcl>). This document cites any implementation differences between the platform types.

In this document these x86 related terms mean the following:

- “x86” refers to the larger family of 64-bit and 32-bit x86 compatible products.
- “x64” relates specifically to 64-bit x86 compatible CPUs.
- “32-bit x86” points out specific 32-bit information about x86 based systems.

For supported systems, see the *Solaris OS: Hardware Compatibility Lists*.

Who Should Use This Book

This book is intended for system administrators who are responsible for installing the Solaris OS. These procedures are advanced Solaris installation information for enterprise system administrators who manage multiple Solaris machines in a networked environment.

Related Books

Table P-1 lists documentation for system administrators.

TABLE P-1 Are You a System Administrator Who is Installing Solaris?

Description	Information
Do you need system requirements or high-level planning information? Or want a high-level overview of Solaris ZFS installations, GRUB based booting, Solaris Zones partitioning technology, or creating RAID-1 volumes?	<i>Oracle Solaris 10 9/10 Installation Guide: Planning for Installation and Upgrade</i>
Do you need to install a single system from DVD or CD media? The Solaris installation program steps you through an installation.	<i>Oracle Solaris 10 9/10 Installation Guide: Basic Installations</i>
Do you need to upgrade or patch your system with almost no downtime? Save system downtime when upgrading by using Solaris Live Upgrade.	<i>Oracle Solaris 10 9/10 Installation Guide: Solaris Live Upgrade and Upgrade Planning</i>
Do you need to install a secure installation over the network or Internet? Use WAN boot to install a remote client. Or, do you need to install over the network from a network installation image? The Solaris installation program steps you through an installation.	<i>Oracle Solaris 10 9/10 Installation Guide: Network-Based Installations</i>
Do you need to install Solaris on multiple machines? Use JumpStart to automate your installation.	<i>Oracle Solaris 10 9/10 Installation Guide: Custom JumpStart and Advanced Installations</i>
Do you need to back up your system?	Chapter 23, “Backing Up and Restoring UFS File Systems (Overview),” in <i>System Administration Guide: Devices and File Systems</i>
Do you need troubleshooting information, a list of known problems, or a list of patches for this release?	<i>Oracle Solaris Release Notes</i>
Do you need to verify that your system works on Solaris?	SPARC: <i>Solaris Sun Hardware Platform Guide</i>
Do you need to check on which packages have been added, removed, or changed in this release?	<i>Oracle Solaris Package List</i>
Do you need to verify that your system and devices work with Solaris SPARC and x86 based systems and other third-party vendors.	<i>Solaris Hardware Compatibility List for x86 Platforms</i>
Do you want to learn more about installing a ZFS root pool?	Chapter 6, “ZFS Root File System Installation (Planning),” in <i>Oracle Solaris 10 9/10 Installation Guide: Planning for Installation and Upgrade</i>

Documentation, Support, and Training

See the following web sites for additional resources:

- [Documentation \(http://docs.sun.com\)](http://docs.sun.com)
- [Support \(http://www.oracle.com/us/support/systems/index.html\)](http://www.oracle.com/us/support/systems/index.html)
- [Training \(http://education.oracle.com\)](http://education.oracle.com) – Click the Sun link in the left navigation bar.

Oracle Welcomes Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of its documentation. If you find any errors or have any other suggestions for improvement, go to <http://docs.sun.com> and click Feedback. Indicate the title and part number of the documentation along with the chapter, section, and page number, if available. Please let us know if you want a reply.

[Oracle Technology Network \(http://www.oracle.com/technetwork/index.html\)](http://www.oracle.com/technetwork/index.html) offers a range of resources related to Oracle software:

- Discuss technical problems and solutions on the [Discussion Forums \(http://forums.oracle.com\)](http://forums.oracle.com).
- Get hands-on step-by-step tutorials with [Oracle By Example \(http://www.oracle.com/technology/obe/start/index.html\)](http://www.oracle.com/technology/obe/start/index.html).
- Download [Sample Code \(http://www.oracle.com/technology/sample_code/index.html\)](http://www.oracle.com/technology/sample_code/index.html).

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P-2 Typographic Conventions

Typeface	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name% you have mail.</code>
AaBbCc123	What you type, contrasted with onscreen computer output	<code>machine_name% su</code> Password:
<i>aabbcc123</i>	Placeholder: replace with a real name or value	The command to remove a file is <code>rm filename</code> .

TABLE P-2 Typographic Conventions (Continued)

Typeface	Meaning	Example
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . <i>A cache</i> is a copy that is stored locally. Do <i>not</i> save the file. Note: Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P-3 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	\$
Bash shell, Korn shell, and Bourne shell for superuser	#
C shell	machine_name%
C shell for superuser	machine_name#

Solaris Flash (Overview)

This book provides instructions for creating Solaris Flash archives and using Solaris Flash archives to install the Solaris OS on multiple systems.

For limitations when creating or installing a Solaris Flash archive, see [Table 2-1](#).

Note – If you want an overview of all Solaris installation methods, see “[Choosing a Solaris Installation Method](#)” in *Oracle Solaris 10 9/10 Installation Guide: Planning for Installation and Upgrade*.

Solaris Flash Introduction

The Solaris Flash installation feature enables you to use a single reference installation of the Solaris OS on a system, which is called the master system. Then, you can replicate that installation on a number of systems, which are called clone systems. You can replicate clone systems with a Solaris Flash initial installation that overwrites all files on the system or with a Solaris Flash update that only includes the differences between two system images. A differential update changes only the files that are specified and is restricted to systems that contain software consistent with the old master image.

What's New in the Oracle Solaris 10 9/10 Release

Oracle Solaris Auto Registration

Oracle Solaris Auto Registration is new in the Oracle Solaris 10 9/10 release. When you install or upgrade your system, configuration data about your system is, on rebooting, automatically communicated through the existing service tag technology to the Oracle Product Registration System. This service tag data about your system is used, for example, to help Oracle enhance customer support and services. You can learn about service tags at <http://wikis.sun.com/display/ServiceTag/Sun+Service+Tag+FAQ>.

You can use this same configuration data to create and manage your own inventory of your systems. By registering with your support credentials using one of the registration options below, you have a straightforward way to inventory your systems, by recording and tracking the service tags for the systems and for the software products installed on the systems. For instructions about tracking your registered products, see <http://wikis.sun.com/display/SunInventory/Sun+Inventory>.

You may elect to have your configuration data sent to the Oracle Product Registration System anonymously. An anonymous registration means that the configuration data sent to Oracle has no link to the name of a customer. You, also, have the option to disable Auto Registration.

For an overview of Auto Registration, see “Oracle Solaris Auto Registration” in *Oracle Solaris 10 9/10 Installation Guide: Planning for Installation and Upgrade*.

Does Auto Registration Impact Solaris Flash Archives?

If you create a Solaris Flash archive based on a master system that was installed with a release prior to the Oracle Solaris 10 9/10 release, that archive does **not** include Auto Registration. Auto Registration has no impact on your work with that archive.

If you create a Solaris Flash archive based on a master system that was installed with the Oracle Solaris 10 9/10 release or a later release, that archive does include Auto Registration unless it was specifically disabled on the master system. For details, see the next section.

How Does Auto Registration Impact Solaris Flash Archives?

For any archive based on the Oracle Solaris 10 9/10 release or a later release, Auto Registration is enabled by default unless it was specifically disabled on the master system. When you install the Solaris Flash archive or upgrade a clone system with the differential flash archive, configuration data about that installed or upgraded system is, on rebooting, automatically communicated through the existing service tag technology to the Oracle Product Registration System.

Auto Registration uses support credentials and proxy information that you provide before or during an installation or upgrade. The means of providing that credential and proxy information depends on which installation method is used, as shown in the following table.

TABLE 1-1 Auto Registration Impact

Installation Method	Auto Registration Impact
Interactive installation	During the installation of a Solaris Flash archive, you are prompted in the installer screens to provide your support credentials and, if needed, proxy information. After the installation, the system is registered on reboot. If you do not provide support credentials, an anonymous registration occurs on reboot.

TABLE 1-1 Auto Registration Impact (Continued)

Installation Method	Auto Registration Impact
Solaris JumpStart	You can provide your support credentials and proxy information by using the <code>auto_reg</code> keyword in the <code>sysidcfg</code> file prior to the installation of an archive or prior to an upgrade with a differential flash archive. If you do not use this keyword, you are prompted to provide this information during the installation of the archive or during the upgrade. After the installation or upgrade, the system is registered on reboot. If you do not provide that information, an anonymous registration occurs on reboot.
Live Upgrade	The Solaris Flash archive uses the same Auto Registration settings, including support credentials and proxy information, that were specified on the master system. As long as Auto Registration was not disabled on the master system, the archive system is, after the upgrade, automatically registered on reboot.
Network Installations, including WAN Boot installations	You can provide your support credentials and proxy information by using the <code>auto_reg</code> keyword in the <code>sysidcfg</code> file prior to the network installation of a Solaris Flash archive. If you do not use this keyword, you are prompted during the network installation to provide this information. The archive is registered when the system reboots after the installation. If you do not provide that information, an anonymous registration occurs on reboot.

For further information, including instructions about how to disable Auto Registration, see “Oracle Solaris Auto Registration” in *Oracle Solaris 10 9/10 Installation Guide: Planning for Installation and Upgrade*.

Disaster Recovery Image

Starting with the Oracle Solaris 10 9/10 release, this document now includes instructions about how to create a Flash Archive recovery image that can be used to restore a system to “factory fresh” condition. See Chapter 5, “Creating and Using a Disaster Recovery Image.” This chapter provides the simplest instructions to create a Flash Archive image that can be loaded onto the target system to recover from a failed disk drive.

What's New in the Solaris 10 10/09 Release

Starting with the Solaris 10 10/09 release, you can set up a JumpStart profile to identify a flash archive of a ZFS root pool.

A Flash archive can be created on a system that is running a UFS root file system or a ZFS root file system. A Flash archive of a ZFS root pool contains the entire pool hierarchy, except for the swap and dump volumes, and any excluded datasets. The swap and dump volumes are created when the Flash archive is installed.

You can use the Flash archive installation method as follows:

- Generate a Flash archive that can be used to install and boot a system with a ZFS root file system.
- Perform a JumpStart installation of a system by using a ZFS Flash archive.

Note – Creating a ZFS Flash archive backs up an entire root pool, not individual boot environments. Individual datasets within the pool can be excluded by using the `flarc create` and `flar` command's `-D` option.

For detailed instructions and limitations, see “[Installing a ZFS Root File System \(Oracle Solaris Flash Archive Installation\)](#)” in *Oracle Solaris ZFS Administration Guide*.

Installing Clone Systems With an Initial Installation

You can install a master system with a Solaris Flash archive for an initial installation by using any installation method: Solaris installation program, custom JumpStart, Solaris Live Upgrade, or WAN boot. All files are overwritten. The Solaris Flash installation is a five-part process.

1. Install the master system. You select a system and use any of the Solaris installation methods to install the Solaris OS and any other software.
2. (Optional) Prepare customization scripts to reconfigure or customize the clone system before or after installation.
3. Create the Solaris Flash archive. The Solaris Flash archive contains a copy of all of the files on the master system, unless you excluded some nonessential files.
4. Install the Solaris Flash archive on clone systems. The master system and the clone system must have the same kernel architecture. For further information, see “[Installing a Sun4U Flash Archive on a Sun4V Machine](#)” on page 25.

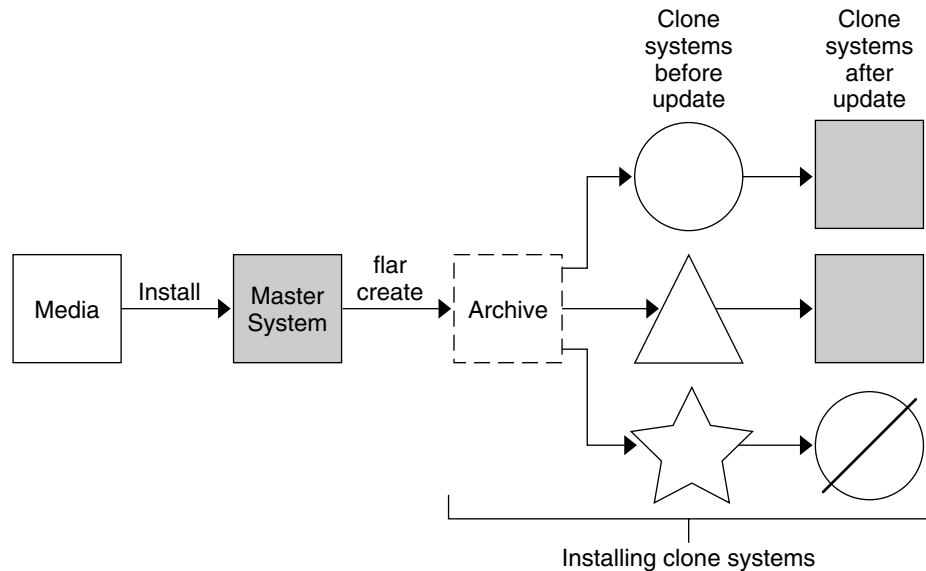
When you install the Solaris Flash archive on a system, all of the files in the archive are copied to that system. The newly installed system now has the same installation configuration as the original master system, thus the system is called a clone system. Some customization is possible:

- Scripts can be used for customization.
 - You can install extra packages with a Solaris Flash archive by using the custom JumpStart installation method. The packages must be from outside the software group being installed or a third-party package.
5. (Optional) Save a copy of the master image. If you plan to create a differential archive, the master image must be available and identical to the image installed on the clone systems.

For step-by-step instructions, see “[Installing the Master System](#)” on page 36.

Figure 1–1 shows an installation of clone systems with an initial installation. All files are overwritten.

FIGURE 1–1 Solaris Flash Initial Installation



- A system running any operating environment
- △ A system with no operating environment
- ☆ A system with a different architecture
- ⊗ Update fails

Updating Clone Systems With a Solaris Flash Differential Archive

If you have a clone system and want to update that system, you can create a differential archive that contains only the differences between two images, the unchanged master image and an updated master image. When you update a clone system with a differential archive, only the files that are in the differential archive are changed. You can choose to install a Solaris Flash differential archive with the custom JumpStart installation method or Solaris Live Upgrade. An update is a five-part process.

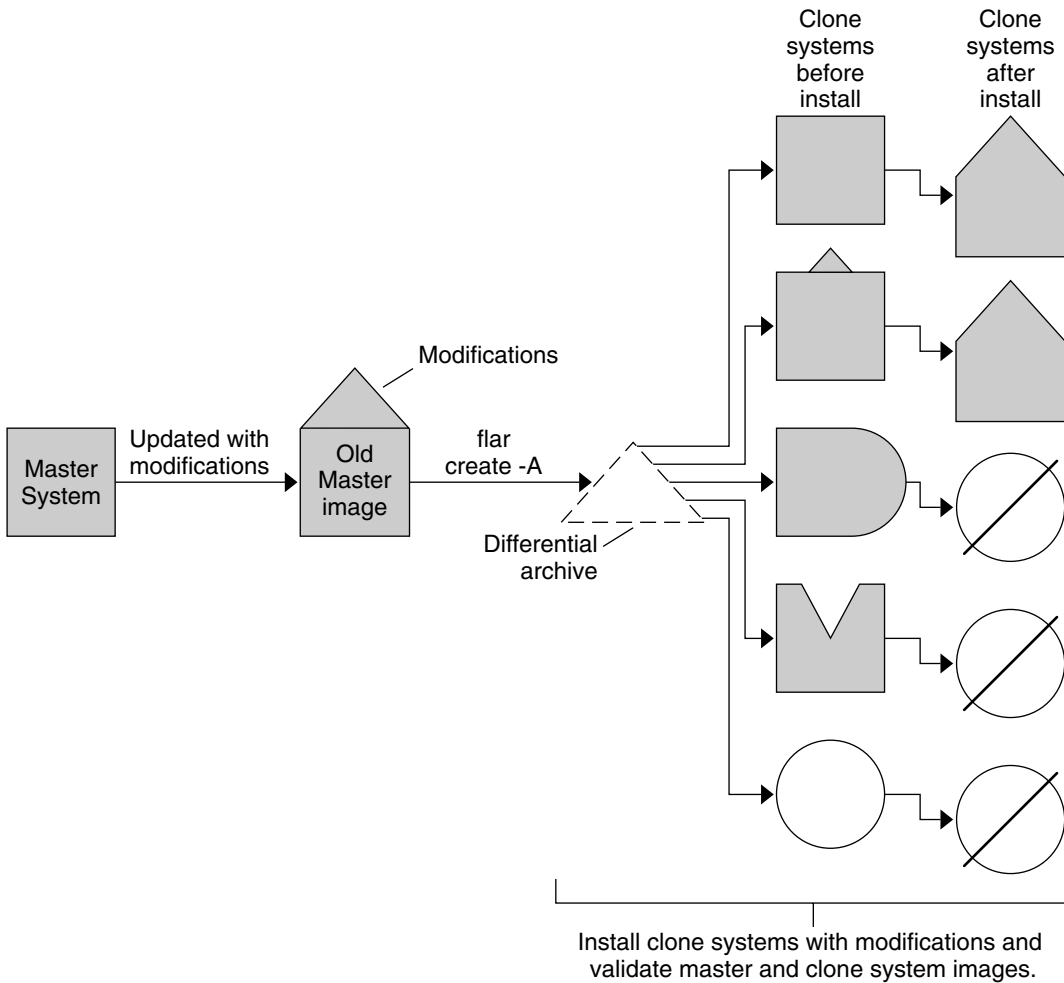
1. Prepare the master system with changes. Before changes are made, the master system should be running a duplicate of the original archive.

Note – If the master system is not running a duplicate of the original archive, the differences between the two system images might result in a large differential archive. Consequently, installing the differential archive could be time consuming. Use an initial installation with a full archive in this case.

2. (Optional) Prepare customization scripts to reconfigure or customize the clone system before or after installation.
3. Mount the directory of a copy of the saved-unchanged master image. This second image is to be used to compare the two system images. Access the image by the following methods.
 - Mounted from a Solaris Live Upgrade boot environment
 - Mounted from a clone system over NFS
 - Restored from backup by using the `ufsrestore` command
4. Create the differential archive with the `-A` option of the `flarcreate` command.
5. Install the differential archive on clone systems with custom JumpStart. Or, you can use Solaris Live Upgrade to install the differential archive on an inactive boot environment.

Figure 1–2 shows the creation and installation of a differential archive. A master image is updated with some modifications. These modifications could be as simple as the addition, reconfiguration, or deletion of a few files, or as complex as propagating patches. The updated master image is compared to the unchanged master image. The differences between the two images become the differential archive. The archive can be used to update other clone systems that are currently using the unchanged master image. If the clone system has already been modified or is not running the unchanged master image, the update fails. If you have many changes to make on the clone systems, you can do an initial installation at any time.

FIGURE 1-2 Solaris Flash Update



- Exact duplicate of master
- ▤ Duplicate of master but contains small changes
- ◐ Duplicate of master but contains extra files
- ◑ Duplicate of master but missing some files
- Created from different master or installed separately
- ⊗ Update fails

Solaris Flash (Planning)

This chapter provides information necessary for planning a Solaris Flash installation in your environment.

Planning Your Solaris Flash Installation

Before you create and install a Solaris Flash archive, you must make some decisions about how you want to install the Solaris OS on your systems. The first time that you install a system, you install with a full archive that is an initial installation. After a system has been installed with an archive, the system can be updated with a differential archive. The differential archive installs only the differences between two archives.

Note – Starting with the Solaris 10 10/09 release, you can set up a JumpStart profile to identify a flash archive of a ZFS root pool. See [“What's New in the Solaris 10 10/09 Release”](#) on page 17.

Review the following limitations before creating and installing a Solaris Flash archive.

TABLE 2-1 Limitations When Creating and Installing a Solaris Flash Archive

Limitation	Description
When installing the Solaris OS with a Solaris Flash archive, the archive and the installation media must contain identical operating system versions.	For example, if the archive is a Solaris 10 operating system and you are using DVD media, then you must use Solaris 10 DVD media to install the archive. If the operating systems versions do not match, the installation on the target system fails.

TABLE 2-1 Limitations When Creating and Installing a Solaris Flash Archive *(Continued)*

Limitation	Description
A Solaris Flash archive cannot be properly created when a non-global zone is installed.	The Solaris Flash feature is not compatible with the Solaris Zones partitioning technology. If you create a Solaris Flash archive, the resulting archive is not installed properly when the archive is deployed under these conditions: <ul style="list-style-type: none"> ■ The archive is created in a non-global zone ■ The archive is created in a global zone that has non-global zones installed
Starting with the Solaris 10 10/08 release , if you are installing a Solaris Flash archive from a release before the Solaris 10 10/08 release, the archive fails to install.	If you need to install a Solaris Flash archive from a previous release, boot from the previous release and install the archive.

Designing an Initial Installation of the Master System

The first task in the Solaris Flash installation process is to install a system, the master system, with the configuration that you want each of the clone systems to have. You can use any of the Solaris installation methods to install an archive on the master system. The installation can be a subset or a complete installation of the Solaris OS. After you complete the installation, you can add or remove software or modify any configuration files. Some limitations to installing the master system are the following:

- The master system and the clone systems must have the same kernel architectures. For example, you can only use an archive that was created from a master system that has a Sun4U architecture to install clones with a Sun4U architecture. For sample instructions, see [“Installing a Sun4U Flash Archive on a Sun4V Machine” on page 25](#).
- You must install the master system with the exact configuration that you want on each of the clone systems. The decisions that you make when you design the installation of the master system depend on the following:
 - The software that you want to install on the clone systems
 - Peripheral devices that are connected to the master system and the clone systems
 - The architecture of the master system and the clone systems

Note – If you already have installed clone systems and want to update these systems with a new configuration, see [“Planning to Create the Solaris Flash Differential Archive for an Update” on page 30](#).

▼ Installing a Sun4U Flash Archive on a Sun4V Machine

Before You Begin Note the following limitations to this procedure:

- These instructions are for simple installations only, not for the following:
 - Installations with zones.
 - Installations with attached storage.
 - Installations with fibre attached or with SAN in use.
- These instructions are for installing through the primary interface only. See CR 6772769.
- These instructions may not work with volume-managed root (encapsulated).
- These instructions are for a UFS root only. Solaris Flash installation of a ZFS root system uses a different installation mechanism.

- 1 **Create a Solaris Flash archive on a Sun4U machine, so that the archive can be installed on a Sun4V machine. Use one of the following two options to add Sun4V as a supported architecture for the archive.**

Note – You must start with a Sun4U machine that has been installed with the Entire Plus OEM Software Group, so that all the driver packages are in the image, even if these packages are not in use. For further information about this requirement, see [“SPARC: Supporting Peripheral Devices Not Found on the Master System” on page 27](#).

Note that, in order for a Flash archive to be installed on different system types, the Entire Plus OEM distribution needs to be installed on the master system.

- **Add the following information to the `/var/sadm/system/admin/.platform` file. Then, create the Solaris Flash archive.**

```
PLATFORM_GROUP=sun4v
```

Note – You can verify that the Sun4V platform group is supported by using the following command:

```
# flar -i <path_to_hybrid>.flar | grep content_architectures
```

This command should display the following results:

```
content_architectures=sun4u,sun4v
```

- **Create the Solaris Flash archive, using the `-U` option to add Sun4V as a supported architecture for the archive. See the following example:**

```
# flarcreate -n S10U5hybrid -U "content_architectures=sun4u,sun4v" \
-c -x /data /data/S10U5hybrid.flar
```

The above sample command provides `/data` for the `-c` option, to indicate the location for the archive. Your value for this `-c` option should reflect your file setup.

Note – You can verify that the Sun4V platform group is supported by using the `flar` command again as shown in the previous step.

2 Install the Solaris Flash archive on the Sun4V machine. You can use Solaris JumpStart and a net image to deploy the Solaris Flash archive.

Note – At this point, the Sun4V machine may not boot. Do not try to patch the machine at this stage. If the machine is allowed to reboot after using JumpStart, you will probably see a message such as:

```
Boot device: /pci@780/pci@0/pci@9/scsi@0/disk@0,0:a File and args:
Boot load failed.
The file just loaded does not appear to be executable.
```

3 Upgrade the Sun4V machine using either a network image or a DVD image.

For example, you could use a Solaris 10 Update 6 JumpStart image. Then, you could boot the Sun4V image from that network image, selecting the upgrade option.

In this example, the upgrade completes with the following issues:

- Where both `.u` and `.v` versions of a package were available, both versions will be installed. See CR 6846077.
- The `/var/sadm/system/admin/.platform` file contains incorrect information. See CR 6523030.
- Any third party `.v` packages are not part of the Solaris image. So, third party packages will probably not be upgraded.

4 Boot the Sun4V machine. You can now apply patches to the machine as needed.

Customizing the Solaris Installation on the Master System

After you install the Solaris OS on the master system by using any of the Solaris installation methods, you can add or delete software and modify system configuration information as necessary. To customize the master system's software, you can do the following:

- Delete software. You can remove software that you determine is not necessary to install on the clone systems. To see a list of software that is installed on the master system, use the Product Registry. For detailed instructions, refer to [System Administration Guide: Basic Administration](#).

- Add software. You can install software that is included in the Solaris release. You can also add software that is not delivered as part of the Solaris OS. All of the software that you install on the master system is included in the Solaris Flash archive and is installed on the clone systems.
- Modify configuration files. You can alter configuration files on the master system. For example, you can modify the `/etc/inet/inetd.conf` file to restrict the daemons that the system runs. All of the modifications that you make are saved as part of the Solaris Flash archive and are installed on the clone systems.
- Further customization can be done when creating the archive. For example, you can exclude large data files that you might not want in the archive. For an overview, see [“Customizing an Archive’s Files and Directories” on page 30](#).

Creating Archives for SPARC and x86 Systems

If you want to install Solaris software by using a Solaris Flash archive on both SPARC and x86 systems, you must create a separate Solaris Flash archive for each platform. Use the Solaris Flash archive that was created from the SPARC master system to install SPARC systems. Use the Solaris Flash archive that was created from the x86 master system to install x86 systems.

SPARC: Supporting Peripheral Devices Not Found on the Master System

Choosing the drivers to install on the master system has the following dependencies.

- The type of peripheral devices attached to both the master system and the clone system.
- The type of software group installed.

The Entire Plus OEM Software Group installs all drivers regardless of the hardware that is present on the system. Other software groups provide limited support. If you install another software group and the clone systems have different peripheral devices than the master system, you need to install the appropriate drivers on the master system before you create the archive.

Note – In order for a Flash archive to be installed on different system types, the Entire Plus OEM distribution needs to be installed on the master system.

How to Get the Support for Peripherals That You Need

You can install support for peripherals on clone systems that are different from the master system in by installing the Entire Plus OEM Software Group or installing selected packages.

Type of Installation	Description
Install the Entire Plus OEM Software Group	<p>The Entire Plus OEM Software Group is the largest Software Group available. This group contains every package that is found in the Solaris OS. The Entire Plus OEM Software Group installs all drivers regardless of the hardware that is present on the system. A Solaris Flash archive that is created with the Entire Plus OEM Software Group works on any clone system that has peripheral devices supported by the installed release of the Solaris OS.</p> <p>Note – In order for a Flash archive to be installed on different system types, the Entire Plus OEM distribution needs to be installed on the master system.</p> <p>Installing master systems with the Entire Plus OEM Software Group guarantees compatibility with other peripheral configurations. However, the Entire Plus OEM Software Group requires at least 2.9 Gbytes of disk space. The clone systems might not have the space that is required to install the Entire Plus OEM Software Group.</p>
Install other software groups	<p>If you install the master system with the following software groups, you are limiting the support for peripherals. The master system supports only the peripheral devices that are attached to the master system at the time of installation.</p> <ul style="list-style-type: none"> ■ Reduced Networking Software Group ■ Core Software Group ■ End User Software Group ■ Developer Software Group ■ Entire Software Group <p>Installing these software groups could result in your clone systems failing to have all the drivers needed. For example, if you install the Entire Software Group on a master system that has a GX CG6 frame buffer, only the GX CG6 frame buffer driver is installed. This situation is not a problem if all the clone systems that you want to install have the GX CG6 frame buffer or no frame buffer.</p>
Install selected packages	<p>When you install the master system, you can install only the packages that you need for the master system and the clone systems. By selecting specific packages, you can install only support for the peripherals that you know exist on the master system or clone systems.</p>

Planning the Creation of a Solaris Flash Archive

You can create an archive from the master system for an initial installation. Or, if you have already installed an archive on clone systems, you can create a differential archive from two system images. The differential archive installs only the differences between the two images.

Planning to Create the Solaris Flash Archive for an Initial Installation

After you install the master system, the next task in the Solaris Flash installation process is to create a Solaris Flash archive. Files on the master system are copied to a Solaris Flash archive along with various pieces of identification information. You can create a Solaris Flash archive while the master system is running in multiuser mode or single-user mode. You can also create a Solaris Flash archive after you boot from one of the following:

- Solaris Operating System DVD
- Solaris Software - 1 CD
- An image of the Solaris Software CDs and the Solaris Languages CDs

Note – Starting with the Oracle Solaris 10 9/10 release, only a DVD is provided. Solaris Software CDs are no longer provided.



Caution – A Solaris Flash archive cannot be properly created when a non-global zone is installed. The Solaris Flash feature is not compatible with the Solaris Zones partitioning technology. If you create a Solaris Flash archive, the resulting archive is not installed properly when the archive is deployed under these conditions:

- The archive is created in a non-global zone
 - The archive is created in a global zone that has non-global zones installed
-

Creating Solaris Flash Archives With RAID-1 Volumes

You can create a Solaris Flash archive when you have Solaris Volume Manager RAID-1 volumes configured. The Solaris Flash creation software removes all RAID-1 volume information from the archive to keep the integrity of the clone system. With custom JumpStart you can rebuild the RAID-1 volumes by using a JumpStart profile. With Solaris Live Upgrade, you create a boot environment with RAID-1 volumes configured and install the archive. The Solaris installation program cannot be used to install RAID-1 volumes with a Solaris Flash archive.

- For examples of RAID-1 volumes in JumpStart profiles, see “[Profile Examples](#)” in *Oracle Solaris 10 9/10 Installation Guide: Custom JumpStart and Advanced Installations*.
- For examples of Solaris Live Upgrade boot environments configured with RAID-1 volumes, see “[Creating a New Boot Environment](#)” in *Oracle Solaris 10 9/10 Installation Guide: Solaris Live Upgrade and Upgrade Planning*.

Note – Veritas VxVM stores configuration information in areas not available to Solaris Flash. If Veritas VxVM file systems have been configured, you should not create a Solaris Flash archive. Also, Solaris install, including JumpStart and Solaris Live Upgrade do not support rebuilding VxVM volumes at installation time. Therefore, if you are planning to deploy Veritas VxVM software using a Solaris Flash archive, the archive must be created prior to configuring the VxVM file systems. The clone systems must be then configured individually after the archive has been applied and the system rebooted.

Creating an Archive That Contains Large Files

The default copy method that is used when you create a Solaris Flash archive is the `cpio` utility. Individual file sizes cannot be over 4 Gbytes. If you have large individual files, you can create an archive with the `pax` copy method. The `flarcreate` command with the `-L pax` option uses the `pax` utility to create an archive without limitations on individual file sizes. Individual file sizes can be greater than 4 Gbytes.

Planning to Create the Solaris Flash Differential Archive for an Update

If you have a clone system that is already installed with an archive and want to update it, you can create a differential archive that contains only the differences between two images, the unchanged master image and an updated master image. The differences between these two images is the differential archive.

- One image is running on the master system that was the original software installed on the clone system. This image might need be installed on the master system if it was saved in a directory for future use.
- Another image is to be accessed and used for comparison. This image contains the new additions or deletions that will be installed on the clone systems.

After you update a clone system with a differential archive, only the files that are in the differential archive are changed on the clone system. Scripts can be used to customize the archive before or after installation, which is especially helpful for reconfiguration.

You can install a Solaris Flash differential archive with the custom JumpStart installation method. Or, you can use Solaris Live Upgrade to install a differential archive on an inactive boot environment.

An unchanged master image should be saved after the initial installation so this image can be accessed by any of the following methods.

- A Solaris Live Upgrade boot environment, mounted on some directory that uses the `lumount` command. For a description of a Solaris Live Upgrade boot environment, see [Chapter 2, “Solaris Live Upgrade \(Overview\),” in *Oracle Solaris 10 9/10 Installation Guide: Solaris Live Upgrade and Upgrade Planning*](#).
- A clone system that is mounted over Network File System (NFS) with root permissions.
- A system backup that can be restored with the `ufsdump` command.

For step-by-step instructions, see [“To Create a Solaris Flash Differential Archive With an Updated Master Image” on page 48](#).

Customizing an Archive's Files and Directories

When you create a Solaris Flash archive, some files and directories that are to be copied from the master system can be excluded. If you have excluded a directory, you can also restore

specified files or subdirectories under that directory. For example, you could create an archive that excludes all files and directories in `/a/aa/bb/c`. The content of the `bb` subdirectory could be included. The only content would then be in the `bb` subdirectory.



Caution – Use the `flarc` create file-exclusion options with caution. If you exclude some directories, others that you were unaware of might be left in the archive, such as system configuration files. The system would then be inconsistent and the installation would not work. Excluding directories and files is best used with data that can easily be removed without disrupting the system, such as large data files.

The following table lists the `flarc` create command options that can exclude files and directories and restore files and subdirectories.

How Specified?	Options That Exclude	Options That Include
Specify the name of the directory or file	<code>-x exclude_dir/filename</code>	<code>-y include_dir/filename</code>
Use a file that contains a list	<code>-X list_filename</code>	<code>-f list_filename</code>
	<code>-z list_filename</code>	<code>-z list_filename</code>

For descriptions of these options, see [Table 6–7](#).

For examples of customizing an archive, see “[Creating a Solaris Flash Archive and Customizing Files \(Examples\)](#)” on page 46.

Customizing an Archive With Scripts

After the software is installed on the master system, special scripts can be run during creation, installation, postinstallation and first reboot. These scripts enable you to do the following:

- Configure applications on clone systems. You can use a custom JumpStart script for some uncomplicated configurations. For more complicated configurations, special configuration-file processing might be necessary on the master system or before or after installation on the clone system.
- Protect local customizations on clone systems. Local preinstallation and postinstallation scripts reside on the clone. These scripts protect local customizations from being overwritten by the Solaris Flash software.
- Identify nonclonable, host-dependent data that enables you to make the archive host independent. Host independence is enabled by modifying such data or excluding it from the archive. An example of host-dependent data is a log file.
- Validate software integrity in the archive during creation.
- Validate the installation on the clone system.

Guidelines for Creating a Custom Script

When creating scripts other than the reboot script, following these guidelines to assure the script does not corrupt the OS or otherwise disrupt the system. These guidelines enable the use of Solaris Live Upgrade, which creates a new boot environment for installation of the OS. The new boot environment can be installed with an archive while the current system is running.

Note – These guidelines are not for reboot scripts that are allowed to run daemons or make other types of modification to the root (/) file system.

- Scripts must not affect the currently running system. The currently running OS might not be the one running when the Solaris Flash archive is installed.
- Scripts must not start or stop any daemon processes.
- Scripts must not depend on the output of commands such as `ps`, `truss`, or `uname`, which are dependent on the OS. These commands report information about the currently running system.
- Scripts must not send any signals or otherwise affect any currently running processes.
- Scripts can use standard UNIX commands that facilitate shell scripting such as `expr`, `cp`, and `ls`.

For an overview of Solaris Live Upgrade, see [Chapter 2, “Solaris Live Upgrade \(Overview\)”](#) in *Oracle Solaris 10 9/10 Installation Guide: Solaris Live Upgrade and Upgrade Planning*.

Solaris Flash Archive Sections

Solaris Flash archives contain the following sections. Some sections can be used by you to identify and customize the archive and view status information on the installation. For a further description of each section, see [Chapter 6, “Solaris Flash \(Reference\)”](#).

TABLE 2-2 Flash Archive Sections

Section Name	Informational Only	Description
Archive cookie	X	The first section contains a cookie that identifies the file as a Solaris Flash archive.
Archive identification		The second section contains keywords with values that provide identification information about the archive. Some identification information is supplied by the archive software. Other specific identification information can be added by you with options to the <code>flarcreate</code> command.
User-defined		This section follows the archive identification section. You can define and insert these sections to customize the archive. The Solaris Flash archive does not process any sections that you insert. For example, a section could contain a description of the archive or perhaps a script to check the integrity of an application.

TABLE 2-2 Flash Archive Sections (Continued)

Section Name	Informational Only	Description
Manifest	X	This section is produced for a Solaris Flash differential archive and is used for validating a clone system. The manifest section lists the files on a system to be retained, added to, or deleted from the clone system. This section is informational only, lists the files in an internal format, and cannot be used for scripting.
Predeployment, Postdeployment, Reboot	X	This section contains internal information that the flash software uses before and after installing an OS image. Any scripts that you have provided are included in this section.
Summary		This section contains messages about the archive creation. The section also records the activities of predeployment and postdeployment scripts. You can view the success of the installation in this section by writing a script to send output to this section.
Archive files	X	The archive files section contains the files that have been gathered from the master system.

When to Create the Archive for an Initial Installation

Create the archive when the system is in as static a state as possible. Create the archive after software is installed on the master system and before software is configured.

Where to Store the Solaris Flash Archive

After you create the Solaris Flash archive, you can save the archive on the hard disk of the master system or on a tape. After you save the archive, you can copy this archive to any file system or media that you choose.

- Network File System (NFS) server
- HTTP or HTTPS server
- FTP server
- Tape
- CD, DVD
- Diskette
- Local drive of clone system that you want to install

Compressing the Archive

When you create the Solaris Flash archive, you can specify that the archive be saved as a compressed file by using the `compress (1)` utility. An archive that is compressed requires less disk storage space and creates less congestion when you install the archive over a network.

Planning the Installation of Solaris Flash Archives

The final task in the Solaris Flash installation process is to install Solaris Flash archives on clone systems. You can use any of the Solaris installation methods to install Solaris Flash archives on clone systems.

Note – Starting with the Oracle Solaris 10 9/10 release, Auto Registration is enabled by default. The impact of Auto Registration on your work with Solaris Flash archives varies depending on which installation method is used. See [“What’s New in the Oracle Solaris 10 9/10 Release”](#) on page 15.

If you are using a pre-Oracle Solaris 10 9/10 archive, there is no Auto Registration impact.

Installation Program	Archives Storable on This Media	For Step-by-Step Instructions
Solaris installation program	<ul style="list-style-type: none"> ■ NFS server ■ HTTP server ■ FTP server ■ Local tape ■ Local device, including DVD or CD ■ Local file 	“Installing a Solaris Flash Archive With the Solaris Installation Program” on page 55
Custom JumpStart installation program	<ul style="list-style-type: none"> ■ NFS server ■ HTTP or HTTPS server ■ FTP server ■ Local tape ■ Local device, including DVD or CD ■ Local file 	“To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation” in <i>Oracle Solaris 10 9/10 Installation Guide: Custom JumpStart and Advanced Installations</i>
Solaris Live Upgrade	<ul style="list-style-type: none"> ■ NFS server ■ HTTP server ■ FTP server ■ Local tape ■ Local device, including DVD or CD ■ Local file 	“Installing Solaris Flash Archives on a Boot Environment” in <i>Oracle Solaris 10 9/10 Installation Guide: Solaris Live Upgrade and Upgrade Planning</i>

Creating Solaris Flash Archives (Tasks)

This chapter provides the procedures for creating a Solaris Flash archive. These procedures include installing a master system and then creating a Solaris Flash archive from that master system. You can also create a differential archive if you have previously installed an archive on a clone system. When the differential archive is created, two images are compared: the unchanged master image and an updated image. The differential archive installs only the differences between the two images. Also, procedures to create scripts are provided to reconfigure or otherwise customize the archive.

For limitations when creating or installing a Solaris Flash archive, see [Table 2-1](#).

- [“Task Map: Creating Solaris Flash Archives” on page 35](#)
- [“Installing the Master System” on page 36](#)
- [“Creating Customization Scripts” on page 37](#)
- [“Creating a Solaris Flash Archive” on page 41](#)

Task Map: Creating Solaris Flash Archives

TABLE 3-1 Task Map: Creating a Solaris Flash Archive to Install With an Initial Installation

Task	Description	For Instructions
Install your chosen software configuration on the master system	Determine the configuration that meets your needs and use any of the Solaris installation methods to install the master system.	“To Install the Master System for an Initial Installation” on page 37
(Optional) Create customization scripts	Determine if you need to create scripts to do the following: <ul style="list-style-type: none"> ■ Customize or reconfigure the archive ■ Protect local changes on clone systems 	“Creating Customization Scripts” on page 37

TABLE 3-1 Task Map: Creating a Solaris Flash Archive to Install With an Initial Installation (Continued)

Task	Description	For Instructions
Create the Solaris Flash archive	Use the <code>flarc</code> command to create an archive.	“To Create a Solaris Flash Archive for an Initial Installation” on page 42
(Optional) Save a copy of the archive	Keep a copy of the archive for future comparison to update a clone system with a differential archive.	“To Create a Solaris Flash Archive for an Initial Installation” on page 42

TABLE 3-2 Task Map: Creating a Solaris Flash Archive to Update a Clone System

Task	Description	For Instructions
Prepare master image	Make changes to the unchanged master image such as adding or deleting packages or installing patches.	“Installing the Master System” on page 36
(Optional) Create customization scripts	Determine if you need to create scripts to do the following: <ul style="list-style-type: none"> ■ Customize or reconfigure the archive ■ Protect local changes on clone systems 	“Creating Customization Scripts” on page 37
Create the Solaris Flash differential archive	<ol style="list-style-type: none"> 1. Mount the unchanged master image. 2. Use the <code>flarc</code> command to compare the two images and create the differential archive. 	“To Create a Solaris Flash Differential Archive With an Updated Master Image” on page 48

Installing the Master System

You install the master system with the software configuration that you want other systems to have. You can install clone systems with an initial installation that overwrites all files on the system or with an update that only includes the differences between two images. For an initial installation, use any of the Solaris installation methods to install the Solaris OS on the master system.

If you have previously installed an archive on a clone system, you can update that system with changes by using a differential archive. The changes are made to the original image such as installing patches, or adding and removing packages. The differential archive overwrites only the files specified in the archive. For the procedure that is for updating the original master image and creating a differential archive, see [“To Create a Solaris Flash Differential Archive With an Updated Master Image” on page 48](#).

▼ To Install the Master System for an Initial Installation

- 1 Identify the system configuration that you want to install.
- 2 With the use of the Solaris installation programs, install the Solaris OS on the master system. For a discussion of the different installation program, refer to [“Choosing a Solaris Installation Method”](#) in *Oracle Solaris 10 9/10 Installation Guide: Planning for Installation and Upgrade*.
- 3 Customize your Solaris installation in any of the following ways:
 - Delete software.
 - Add software.
 - Modify configuration files.
 - Add support for peripheral devices on the clone system.

You can create custom scripts or use the `flarc create` command to create the archive.

- To create custom scripts, see [“Creating Customization Scripts”](#) on page 37.
- To create the archive, see [“Creating a Solaris Flash Archive”](#) on page 41.

Creating Customization Scripts

Scripts can customize the archive. Use these scripts for the following purposes:

- A precreation script validates the archive at creation time and prepares the archive for later customization, especially differential archives. This script also can create a user-defined section in the archive.
- A predeployment script validates the archive during installation and prepares the archive for later customization.
- A postdeployment script reconfigures a new system image on a clone system.
- A reboot script processes a final reconfiguration after the system is rebooted.

For guidelines about creating scripts, see [“Guidelines for Creating a Custom Script”](#) on page 32.

▼ To Create a Precreation Script

This script runs during archive creation. The script has various uses.

- Validates the contents and the integrity of the software. The script fails the archive creation if the integrity is broken.
- Prepares products for later customization on clone system.
- Registers other installation scripts dynamically during archive creation.

- Adds a message to the flash-creation summary file. The message must be short and record only that scripts were started and finished and the results. You can view the results in the summary section.
- 1 **Create the precreation script. Follow the guidelines that are described in “Guidelines for Creating a Custom Script” on page 32.**
 - 2 **Store the script in the `/etc/flash/precreation` directory.**

Example 3-1 Excerpts From a Precreation Script

The following examples are excerpts from a precreation script.

- To log the start time in the summary section, use the following example:


```
echo "MyApp precreation script started">> $FLASHDIR/summary
```
- To check the software integrity, use the `flcheck` command. This command cannot be used at the command line. The syntax for this command is as follows:

```
flcheck software component files and directories ... | -
```

For example, to validate the files and directories, use the following example:

```
flcheck software component files and directories  
If Not in selection - refuse creation
```

```
echo "Myapp Integrity Damage">>$FLASHDIR/summary
```

Or, to keep new files and directories that are unexpected and not fail the archive creation, use the following example:

```
flcheck software component files and directories  
If Not in selection include by force  
flinclude software component
```

- To register deployment scripts and data, use the following example:
 - Copy the script to the following directory:


```
cp predeployment script /etc/flash/predeployment
```
 - Or, to register the script dynamically during archive creation, copy the script to the following directory.

```
cp predeployment script $FLASHDIR/predeployment
```

- To see application-specific data in a user-defined section, use the following example:

```
cp custom section $FLASHDIR/custom_sections/MyApp
```

- To log the success of the installation in the summary section, use the following example:

```
echo "product one flash preparation started." >>$FLASH_DIR/summary  
...  
echo "product one flash preparation finished successfully">>$FLASH_DIR/summary
```

Example 3-2 Precreation Script

```
#!/bin/sh
echo "Test precreation script started" >> $FLASH_DIR/summary
cat /opt/TestApp/critical_file_list | flcheck -
if [ $? != 0 ]; then
    echo "Test precreation script failure" >> $FLASH_DIR/summary
    exit 1
fi
echo "Test precreation script started" >> $FLASH_DIR/summary
/opt/TestApplication/license_cloning
    $FLASH_DIR/predeployment/.TestApplicationLicenceTransfer \
    $FLASH_DIR/custom_sections/TestApplicationLicenceCounter
echo "Test precreation script finished" >> $FLASH_DIR/summary
exit 0
```

Using a Precreation Script to Create a User-Defined Archive Section

A precreation script can create a user-defined section in the archive to provide specific application information. This section is intended for archive maintenance. The script must be put in the `$FLASH_DIR/sections` directory. The Solaris Flash archive does not process a user-defined section. For example, a section could contain a description of the archive or perhaps a script to check the integrity of an application.

A user-defined section requires the following format.

- Must be line oriented
- Must terminate with newline (ASCII 0x0a) characters
- Can have unlimited length of individual lines
- Must encode binary data by using base64 or a similar algorithm

▼ To Create a Predeployment Script

This script is run before the installation of the archive. If the script is meant to validate the archive, it is kept in the archive. If the script is meant to preserve local configurations on the clone system, it is kept on the clone system. This script also can analyze and collect local data necessary for later customization. For example, client-specific information can be saved before being overwritten by files about to be extracted. This information can then be used in the final stage after extraction.

- 1 **Create the predeployment script. Follow the guidelines that are described in [“Guidelines for Creating a Custom Script”](#) on page 32.**
- 2 **Store the script in one of the following directories.**
 - For archive validation, store in the `/etc/flash/predeployment` directory.

- If you are referencing with a precreation script, store in the `$FLASH_DIR/preinstall` directory.
- If you are preserving configurations on a clone system, provide the path to the script that is stored on the clone system with the `local_customization` keyword in the JumpStart profile.

Example 3-3 Predeployment Script

```
#!/bin/sh
$FLASH_DIR/TestApplication/check_hardware
if [ $? != 0 ]; then
    echo Unsupported hardware
    exit 1
fi
$FLASH_DIR/TestApplication/check_licence_key
if [ $? != 0 ]; then
    echo No license for this host
    exit 1
fi
$FLASH_DIR/TestApplication/deploy_license_key \
    $FLASH_DIR/TestApplication/.TestApplicationLicenceTransfer
$FLASH_DIR/TestApplication/save_data_files $FLASH_DIR/flash

exit 0
```

▼ To Create a Postdeployment Script

This script is kept in the archive or stored in a local directory on the clone system and runs after installation. The script reconfigures a new system image on a clone system. If the script is stored in the archive, the changes affect all the clone systems. If the script is stored in a local directory on the clone system, the changes affect only the clone system. For example, client-specific information that is saved by a predeployment script can be applied to the clone environment, completing the installation.

Postdeployment scripts can also be used to clean up files after the archive is installed. For example, log files such as those files in `/var/adm` can be cleaned out.

Note – Not all log files need a script for cleanup. Log files in `/var/tmp` can be excluded when the archive is created.

- 1 **Create the postdeployment script. Follow the guidelines that are described in “Guidelines for Creating a Custom Script” on page 32.**
- 2 **Store the script in one of the following directories.**
 - To affect all clone systems, store the script in the `/etc/flash/postdeployment` directory.

- To affect only a local clone system, provide the path to the script that is stored on the clone system with the `local_customization` keyword in the JumpStart profile.

Example 3-4 Postdeployment Script

```
#!/bin/sh
$FLASH_DIR/TestApplication/clone_reconfiguration
$FLASH_DIR/TestApplication/restore_data $FLASH_DIR/flash
```

▼ To Create a Reboot Script

This script is kept in the archive and runs after the system is rebooted. The script does any final configuration after system reconfiguration.

After you install the Solaris Flash archive on a clone system, some host-specific files are deleted and are re-created for the clone machine. The installation program uses the `sys-unconfig(1M)` command and the `sysidtool(1M)` programs to delete and re-create host-specific network configuration files. The files that are re-created include, for example, `/etc/hosts`, `/etc/defaultrouter`, and `/etc/defaultdomain`. You can use the reboot script for any final reconfiguration.

- 1 **Create the reboot script.**
- 2 **Store the script in the `/etc/flash/reboot` directory.**

Example 3-5 Creating a reboot Script

```
#!/bin/sh
$FLASH_DIR/TestApplication/finalize_license
```

Creating a Solaris Flash Archive

You can create an archive with an initial installation that overwrites all the files on the clone system, or you can create a differential archive that only overwrites the changes that are specified. For an overview of a differential archive, see [“Planning to Create the Solaris Flash Differential Archive for an Update”](#) on page 30.



Caution – A Solaris Flash archive cannot be properly created when a non-global zone is installed. The Solaris Flash feature is not compatible with the Solaris Zones partitioning technology. If you create a Solaris Flash archive, the resulting archive is not installed properly when the archive is deployed under these conditions:

- The archive is created in a non-global zone
 - The archive is created in a global zone that has non-global zones installed
-

▼ To Create a Solaris Flash Archive for an Initial Installation

After you install the master system, create a Solaris Flash archive to use to install other systems.

1 Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see “[Configuring RBAC \(Task Map\)](#)” in *System Administration Guide: Security Services*.

2 Boot the master system and run it in as inactive a state as possible.

When possible, run the system in single-user mode. If that is not possible, shut down any applications that you want to archive and any applications that require extensive operating system resources.

You can create a Solaris Flash archive while the master system is running in multiuser mode, single-user mode, or while booted from one of the following:

- Solaris Operating System DVD.

Note – **Starting with the Oracle Solaris 10 9/10 release**, only a DVD is provided. Solaris Software CDs are no longer provided.

- Solaris Software - 1 CD.
- An image of the Solaris Software. If you are using CD media, the image could include the Solaris Languages CDs if needed.

3 To create the archive, use the `flarcreate` command.

```
# flarcreate -n name options path/filename
```

name The name that you give the archive. The *name* you specify is the value of the `content_name` keyword.

options For a description of options, see “[flar Command](#)” on page 76.

path The path to the directory in which you want to save the archive file. If you do not specify a path, `flarcreate` saves the archive file in the current directory.

filename The name of the archive file.

- If the archive creation is successful, the `flarcreate` command returns an exit code of 0.
- If the archive creation fails, the `flarcreate` command returns a nonzero exit code.

- 4 **Make a copy of the archive and save it. The copy can be used in the future to update a clone system with a differential archive.**

Creating a Solaris Flash Archive (Examples)

File systems can be copied exactly or can be customized by excluding some directories or files. You can achieve the same results by using different options. Use the options that best suit your environment.

The file systems in the following examples have been greatly simplified for clarification. Rather than use file system names such as `/var`, `/usr`, or `/opt`, the master system file structure for these examples is the following:

```
/aaa/bbb/ccd/ddd
/aaa/bbb/fff
/aaa/eee
/999
```



Caution – Use the `flarcreate` file-exclusion options with caution. If you exclude some directories, others that you were unaware of might be left in the archive, such as system configuration files. The system would then be inconsistent and the installation would not work. Excluding directories and files is best used with data that can easily be removed without disrupting the system, such as large data files.

Creating a Solaris Flash Archive (Various Examples)

EXAMPLE 3-6 Creating an Exact Duplicate Archive

In this example, the archive is named `archive1`. This archive is copied exactly from the master system and then compressed. The archive is an exact duplicate of the master system and is stored in `archive1.flar`.

```
# flarcreate -n archive1 -c archive1.flar
```

To check the file structure of the archive, type the following.

EXAMPLE 3-6 Creating an Exact Duplicate Archive (Continued)

```

# flar info -l archive1.flar lost+found
export
export/home
export/home/lost+found
var
var/sadm
var/sadm/install
var/sadm/install/admin
var/sadm/install/admin/default
var/sadm/install/logs
var/sadm/install/contents
var/sadm/install/.lockfile
var/sadm/install/.pkg.lock
var/sadm/pkg
var/sadm/pkg/SUNWocfd
var/sadm/pkg/SUNWocfd/install
var/sadm/pkg/SUNWocfd/install/copyright
var/sadm/pkg/SUNWocfd/save
var/sadm/pkg/SUNWocfd/save/pspool
var/sadm/pkg/SUNWocfd/save/pspool/SUNWocfd
.....
.....
usr/bin/sparcv7
usr/bin/sparcv7/savecore
usr/bin/sparcv7/gcore
....
....
usr/lib/diff3prog
usr/lib/madv.so.1
usr/lib/mpss.so.1
usr/lib/cpu
usr/lib/cpu/sparcv8plus
....
....
devices/pseudo/udp6@0:udp6
devices/pseudo/udp@0:udp
devices/pseudo/tcp@0:tcp
devices/pseudo/iwscn@0:iwscn
devices/pseudo/wc@0:wscns
devices/pseudo/tcp6@0:tcp6
devices/pseudo/sctp6@0:sctp6
var/fm/fmd/ckpt
var/fm/fmd/rsrc
kernel/drv/st.conf
kernel/drv/st.conf
kernel/drv/st.conf
kernel/drv/st.conf
#

```

EXAMPLE 3-7 Creating an Archive That Contains Large Files

In this example, some individual files are greater than 4 Gbytes. The default archiving utility, `cpio`, cannot handle these large files. The `-L pax` copy method is used to create an archive that

EXAMPLE 3-7 Creating an Archive That Contains Large Files (Continued)

contains large individual files. The archive is named `archive1`. This archive is copied exactly from the master system and then compressed. The archive is an exact duplicate of the master system and is stored in `archive1.flar`.

```
# flarcreate -L pax -n archive1 -c archive1.flar
```

To check the file structure of the archive, type the following.

```
# flar info -l archive1.flar
aaa
aaa/bbb
aaa/bbb/ccc
aaa/bbb/ccc/ddd
aaa/bbb/fff
aaa/eee
aaa/eee
ggg
```

EXAMPLE 3-8 Creating an Archive From an Alternate root (/) File System

In this example, the archive is named `archive4`. This archive is copied exactly from the master system and then compressed. The archive is an exact duplicate of the master system and is stored in `archive4.flar`. The `-R` option is used to create the archive from another directory tree.

```
# flarcreate -n archive4 -c -R /x/yy/zz archive4.flar
```

EXAMPLE 3-9 Creating an Archive and Adding Keywords to Describe the Archive

In this example, the archive is named `archive3`. This archive is copied exactly from the master system and then compressed. Options add descriptions to the archive-identification section, which can help you to identify the archive later. For information about keywords, their values, and formats, see [“Solaris Flash Keywords” on page 71](#).

```
# flarcreate -n archive3 -i 20000131221409 -m pumbaa \
-e "Solaris 8 Print Server" -a "Mighty Matt" -U "Internal Finance" \
-T server archive3.flar
```

After the archive is created, you can access the archive identification section that contains the detailed description. An example of an archive identification section follows.

```
section_begin=identification
files_archived_method=cpio
files_compressed_method=compress
files_archived_size=259323342
files_unarchived_size=591238111
creation_date=20000131221409
creation_master=pumbaa
```

EXAMPLE 3-9 Creating an Archive and Adding Keywords to Describe the Archive (Continued)

```

content_name=Finance Print Server
content_type=server
content_description=Solaris 8 Print Server
content_author=Mighty Matt
content_architectures=sun4u
creation_node=pumbaa
creation_hardware_class=sun4u
creation_platform=SUNW,Sun-Fire
creation_processor=sparc
creation_release=5.9
creation_os_name=SunOS
creation_os_version=s81_49
x-department=Internal Finance

```

Creating a Solaris Flash Archive and Customizing Files (Examples)**EXAMPLE 3-10** Creating an Archive and Excluding and Including Files and Directories

In this example, the archive is named `archive2`. This archive is copied from the master system but is not an exact copy. The content under the `/aaa` directory is excluded, but the content in `/aaa/bbb/ccd` remains.

```
# flarcreate -n archive2 -x /aaa -y /aaa/bbb/ccd archive2.flar
```

To check the file structure of the archive, type the following. The excluded directories that include copied files appear, but only the files that were restored contain data.

```
# flar info -l aaa
aaa
aaa/bbb/ccd
aaa/bbb/ccd/ddd
aaa/bbb
ggg
```

EXAMPLE 3-11 Creating an Archive Excluding and Including Files and Directories by Using Lists

In this example, the archive is named `archive5`. This archive is copied from the master system but is not an exact copy.

The `exclude` file contains the following list:

```
/aaa
```

The `include` file contains the following list:

```
/aaa/bbb/ccd
```

The content under the `/aaa` directory is excluded, but the content in `/aaa/bbb/ccd` remains.

EXAMPLE 3-11 Creating an Archive Excluding and Including Files and Directories by Using Lists
(Continued)

```
# flarcreate -n archive5 -X exclude -f include archive5.flar
```

To check about the file structure of the archive, type the following. The excluded directories that include copied files appear, but only the files that were restored contain data.

```
# flar info -l archive5.flar
aaa
aaa/bbb/ccc
aaa/bbb/ccc/ddd
aaa/bbb
ggg
```

EXAMPLE 3-12 Creating an Archive Excluding Files and Directories by Using a List and Restoring a Directory

You can combine options `-x`, `-y`, `-X` and `-f`. In this example, options `-X` and `-y` are combined. The archive is named `archive5`. This archive is copied from the master system but is not an exact copy.

The `exclude` file contains the following list:

```
/aaa
```

The `-y` option restores the `/aaa/bbb/ccc` directory. The following command produces the archive.

```
# flarcreate -n archive5 -X exclude -y /aaa/bbb/ccc archive5.flar
```

To check about the file structure of the archive, type the following. The excluded directories that include copied files appear, but only the files that were restored contain data.

```
# flar info -l archive5.flar
aaa
aaa/bbb
aaa/bbb/ccc
aaa/bbb/ccc/ddd
ggg
```

EXAMPLE 3-13 Creating an Archive Excluding and Including Files and Directories by Using a List With the `-z` Option

In this example, the archive is named `archive3`. It is copied from the master system but is not an exact copy. The files and directories to be selected are included in `filter1` file. Within the files, the directories are marked with a minus (`-`) or a plus (`+`) to indicate which files to exclude and restore. In this example, the directory `/aaa` is excluded with a minus and the subdirectory `/aaa/bbb/ccc` is restored with a plus. The `filter1` file contains the following list.

EXAMPLE 3-13 Creating an Archive Excluding and Including Files and Directories by Using a List With the `-z` Option *(Continued)*

```
- /aaa  
+ /aaa/bbb/ccc
```

The following command produces the archive.

```
# flarcreate -n archive3 -z filter1 archive3.flar
```

To check the file structure of the archive, type the following command. The excluded directories that include copied files appear, but only the files that were restored contain data.

```
# flar info -l archive3.flar  
aaa  
aaa/bbb  
aaa/bbb/ccc  
aaa/bbb/ccc/ddd  
ggg
```

▼ To Create a Solaris Flash Differential Archive With an Updated Master Image

Before creating a differential archive, you need two images to compare: an unchanged master image and an updated master image. One image is the unchanged master image that has been kept unchanged. This image was stored and needs to be accessed. The second image is the unchanged master image that is updated with minor changes. The root (`/`) file system is the default for the new image, but you can access this image if it has been stored elsewhere. After you have the two images, you can create a differential archive, which contains only the differences between the two images. The differential archive can then be installed on clones that were installed previously with the unchanged master image.

- 1 **Prepare the master system with changes. Before changes are made, the master system should be running a duplicate of the original archive.**

Note – A copy of the unchanged master image must be kept protected from changes and available for mounting later.

- 2 **Update the unchanged master image with any of the following changes.**
 - Delete packages.
 - Add packages or patches.
 - Modify configuration files.
 - Add support for peripheral devices on the clone system.

3 (Optional) Create custom scripts. See [“Creating Customization Scripts” on page 37](#).

4 Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see [“Configuring RBAC \(Task Map\)” in *System Administration Guide: Security Services*](#).

5 Deliver the unchanged master image in a mount point.

- If the unchanged master image is stored on an inactive boot environment, retrieve it by using the `lumount` command.

```
# lumount BE_name mountpoint
```

BE_name Specifies the boot environment name where the unchanged master image is stored

mountpoint Specifies a root (/) file system where the image is stored

In the following example, the inactive boot environment is named `unchanged_master1`. The mount point is the directory `/a` on the master system.

```
# lumount unchanged_master1 /a
```

- If the image is stored on a clone, mount the clone by using NFS.
 - a. On the master system, share the clone's root (/) file system and give the master root permissions on the clone system.

```
# share -F nfs -o rw,root=master_system "/"
```

master_system is the name of the master system.

- b. On the master system, mount the clone.

```
# mount -F nfs clone_system:/ master_dir
```

clone_system Specifies the name of the system to be mounted

master_dir Specifies the directory where the unchanged master image is stored

- If you saved the image with the `ufsdump` command, use the `ufsrestore` command to retrieve a copy. For information about how to use these commands, see [Chapter 27, “UFS Backup and Restore Commands \(Reference\)” in *System Administration Guide: Devices and File Systems*](#).

6 Create the differential archive.

```
# flarcreate -n archive_name -A unchanged_master_image_dir \
  options path/filename
```

archive_name

Specifies the name that you give the archive. The *archive_name* you specify is the value of the `content_name` keyword. The name is listed in the archive-identification section.

-A unchanged_master_image_dir Creates a differential archive by comparing a new system image with the image that is specified by the *unchanged_master_image_dir* argument. By default, the new system image is root (/). You can change the default with the -R option. *unchanged_master_image_dir* is a directory where the unchanged system image is stored or mounted through UFS, NFS, or the `lumount` command.

You can include and exclude some files by using the options for contents selection. For a list of options, see “[flar Command](#)” on page 76.

options For a description of options, see “[flar Command](#)” on page 76.

path Specifies the path to the directory in which you want to save the archive file. If you do not specify a path, `flarcreate` saves the archive file in the current directory.

filename Specifies the name of the archive file.

- If the differential archive creation is successful, the `flarcreate` command returns an exit code of 0.
- If the differential archive creation fails, the `flarcreate` command returns a nonzero exit code.

For procedures about installing an archive, see “[To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation](#)” in *Oracle Solaris 10 9/10 Installation Guide: Custom JumpStart and Advanced Installations*.

Example 3–14 Creating a Differential Archive With the New Master Image on the Master System

In this example, the directory for unchanged master image is named `unchanged_master1`. The new master image that contains changes is the root (/) directory. The new master image is compared to the unchanged master image and the resulting differential archive is then compressed. The differential archive is stored in `diffarchive1.flar` file. The archive contains files that are to be deleted, changed, or added when installed.

```
# flarcreate -n diffarchive1 -A /a/unchanged_master1 -c diffarchive1.flar
```

Example 3–15 Creating a Differential Archive With the Images Stored on an Inactive Boot Environment

In this example, the unchanged master image, `unchanged_master1`, is stored on an inactive boot environment and is accessed by mounting the boot environment. The new master image is the root (/) directory. The new master image is compared to the unchanged master and the

resulting differential archive is then compressed. The archive is stored in `diffarchive4.flar`. The archive contains files that are to be deleted, changed, or added when installed.

```
# lumount unchanged_master1 /a
# flarcreate -n diffarchive4 -A /a -c diffarchive4.flar
```

▼ To Create a Solaris Flash Differential Archive by Using Solaris Live Upgrade

To manage system updates, you can use Solaris Live Upgrade to copy the OS, which creates a new boot environment. This copy can be compared to the master system that has been updated with minor changes. The resulting Solaris Flash differential archive can then be installed on clone systems.

For more information about Solaris Live Upgrade, see [Chapter 2, “Solaris Live Upgrade \(Overview\)”](#) in *Oracle Solaris 10 9/10 Installation Guide: Solaris Live Upgrade and Upgrade Planning*.

1 From the unchanged master system, create a new boot environment by using the `lucreate` command.

This new boot environment is an exact copy of the master system and can be used to create the differential archive.

2 Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see [“Configuring RBAC \(Task Map\)”](#) in *System Administration Guide: Security Services*.

3 Check the status of the two boot environments.

```
# lustatus copy_BE
```

boot environment Name	Is Complete	Active Now	Active OnReboot	Can Delete	Copy Status
master_BE	yes	yes	yes	no	-
copy_BE	yes	no	no	yes	-

4 Update the master image with any of the following changes.

- Delete packages.
- Add packages or patches.
- Modify configuration files.
- Add support for peripheral devices on the clone system.

5 (Optional) Create custom scripts. See [“Creating Customization Scripts”](#) on page 37.

6 Create the differential archive.

a. Mount the newly created boot environment.

```
# lumount BE_name /a
```

b. Create the differential archive by comparing the master system to the boot environment.

```
# flarcreate -n archive_name -A new_BE_dir\ options path/filename
```

archive_name Specifies the name that you give the archive.

-A new_BE_dir Creates a differential archive by comparing a new system image with the image that is specified by the *new BE_dir* argument.

options For a list of options, see “[flar Command](#)” on page 76.

path Specifies the path to the directory in which you want to save the archive file. If you do not specify a path, `flarcreate` saves the archive file in the current directory.

filename Specifies the name of the archive file.

c. Unmount the new boot environment.

```
# lumount copy_BE
```

The `flarcreate` command returns an exit code.

- If the creation is successful, an exit code of 0 is returned.
- If a failure occurs, a nonzero exit code is returned.

7 Install the Solaris Flash differential archive by using a JumpStart profile.

The clone systems that are to be installed must be a duplicate of the original master system or the installation fails.

The following example profile installs a differential archive, `test.diff`, on the device `c1t1d0s0`.

```
JumpStart profile
-----
install_type flash_update
archive_location http server /rw/test.diff
root_device c1t1d0s0
```

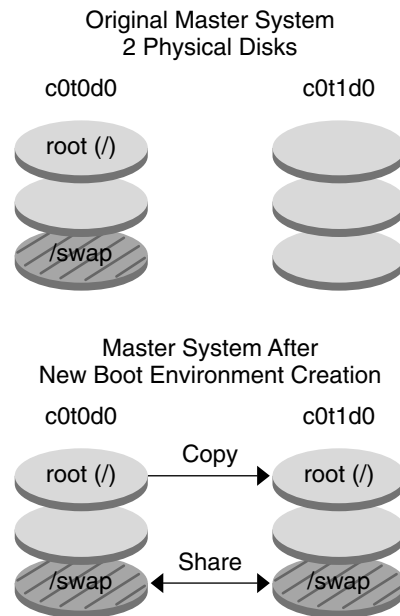
Example 3–16 Creating a Differential Archive by Using Solaris Live Upgrade

`master_BE` is the name of the current boot environment. `copy_BE` is the name of the new boot environment. The file systems `root (/)` and `/usr` are placed on `s0` and `s3`. The `lustatus` command reports that the new boot environment `copy` is complete. The `SUNWman` package is added to the master system. After the master system is updated by adding the `SUNWman` package, the `flarcreate` command creates a differential archive by comparing the changed master and the unchanged new boot environment.

```
# lucreate -c master_BE -m /:/dev/dsk/c0t1d0s0:ufs \
-m /usr:/dev/dsk/c0t1d0s3:ufs -n copy_BE
# lustatus
# pkgadd SUNWman
# lumount copy_BE /a
# flarcreate -n test.diff -c -A /a /net/server/export/test.diff
# luumount copy_BE
```

Install the differential archive on clone systems. For procedures about installing an archive, see [“To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation”](#) in *Oracle Solaris 10 9/10 Installation Guide: Custom JumpStart and Advanced Installations*.

The following image shows the creation of the new boot environment by using the `lucreate` command.



```
Command: # lucreate
          -m /:/dev/dsk/c0t1d0s0:ufs \
          -n second_disk
```


Installing and Administering Solaris Flash Archives (Tasks)

This chapter provides step-by-step procedures for installing a Solaris Flash archive by using the Solaris installation program. This chapter also provides references to procedures for installing Solaris Flash archives when using other installation programs. Also, step-by-step procedures for administering an archive are provided.

For limitations when creating or installing a Solaris Flash archive, see [Table 2-1](#).



Caution – When installing the Solaris OS with a Solaris Flash archive, the archive and the installation media must contain identical operating system versions. For example, if the archive is a Solaris 10 operating system and you are using DVD media, then you must use Solaris 10 DVD media to install the archive. If the operating systems versions do not match, the installation on the target system fails.

- If you want to use the Solaris installation program, see “[Installing a Solaris Flash Archive With the Solaris Installation Program](#)” on page 55.
- If you want to use the custom JumpStart installation method or Solaris Live Upgrade, see “[References to Procedures for Installing Solaris Flash Archives](#)” on page 57.
- To split or merge an archive, see “[Administering Solaris Flash Archives](#)” on page 58.

Installing a Solaris Flash Archive With the Solaris Installation Program

To use the Solaris installation program to install a Solaris Flash archive, use the following procedure.

▼ Installing a Solaris Flash Archive

- 1 **Begin the Solaris installation program and proceed through the panels until you reach the Specify Media panel. Continue with Solaris Flash installation.**

For the step-by-step procedures, see either of the following procedures.

- SPARC: “Performing an Installation or Upgrade With the Solaris Installation Program for UFS File Systems” in *Oracle Solaris 10 9/10 Installation Guide: Basic Installations*
- x86: “Performing an Installation or Upgrade With the Solaris Installation Program for UFS File Systems” in *Oracle Solaris 10 9/10 Installation Guide: Basic Installations*

- 2 **Specify the media you are using to install.**

- a. **Type the information that you are prompted to enter.**

Media Selected	Prompt
DVD or CD	Insert the disc where the Solaris Flash archive is located.
Network File System	Specify the path to the network file system where the Solaris Flash archive is located. You can also specify the archive file name.
HTTP	Specify the URL and proxy information that is needed to access the Solaris Flash archive.
FTP	Specify the FTP server and the path to the Solaris Flash archive. Specify the user and password information that allows you access to the FTP server. Specify any proxy information that is needed to access the FTP server.
Local tape	Specify the local tape device and the position on the tape where the Solaris Flash archive is located.

If you selected to install an archive from a DVD, CD, or from an NFS server, the Select Flash Archives panel is displayed.

- b. **For archives that are stored on a disc or an NFS server, on the Select Flash Archives panel, select one or more Solaris Flash archives to install.**
- c. **On the Flash Archives Summary panel, confirm the selected archives and click Next.**
- d. **On the Additional Flash Archives panel, you can install an additional Solaris Flash archive by specifying the media where the other archive is located. If you do not want to install additional archives, select None.**

3 Click Next to continue the installation.

Follow the steps from one of the following procedures to complete the installation.

- SPARC: “Performing an Installation or Upgrade With the Solaris Installation Program for UFS File Systems” in *Oracle Solaris 10 9/10 Installation Guide: Basic Installations*
- x86: “Performing an Installation or Upgrade With the Solaris Installation Program for UFS File Systems” in *Oracle Solaris 10 9/10 Installation Guide: Basic Installations*

References to Procedures for Installing Solaris Flash Archives

You can use any of the Solaris installation methods to install Solaris Flash archives for an initial installation. You must use custom JumpStart or Solaris Live Upgrade to install a Solaris Flash differential archive.

Note – Starting with the Oracle Solaris 10 9/10 release, Auto Registration is enabled by default. The impact of Auto Registration on your work with Solaris Flash archives varies depending on which installation method is used. See “What’s New in the Oracle Solaris 10 9/10 Release” on page 15.

If you are using a pre-Oracle Solaris 10 9/10 archive, there is no Auto Registration impact.

Type of Installation	Reference
An initial installation to install a Solaris Flash archive	<ul style="list-style-type: none"> ■ Solaris installation program – See the previous procedure “Installing a Solaris Flash Archive With the Solaris Installation Program” on page 55 ■ Solaris Live Upgrade – See “Installing Solaris Flash Archives on a Boot Environment” in <i>Oracle Solaris 10 9/10 Installation Guide: Solaris Live Upgrade and Upgrade Planning</i>. ■ Custom JumpStart installation program – See “Creating a Profile” in <i>Oracle Solaris 10 9/10 Installation Guide: Custom JumpStart and Advanced Installations</i> and “To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation” in <i>Oracle Solaris 10 9/10 Installation Guide: Custom JumpStart and Advanced Installations</i>. ■ WAN boot installation method – See Chapter 10, “WAN Boot (Overview),” in <i>Oracle Solaris 10 9/10 Installation Guide: Network-Based Installations</i>.

Type of Installation	Reference
An update with a Solaris Flash differential archive	<ul style="list-style-type: none"> ■ Custom JumpStart installation program – See “Creating a Profile” in <i>Oracle Solaris 10 9/10 Installation Guide: Custom JumpStart and Advanced Installations</i> and “To Prepare to Install a Solaris Flash Archive With a Custom JumpStart Installation” in <i>Oracle Solaris 10 9/10 Installation Guide: Custom JumpStart and Advanced Installations</i>. ■ Solaris Live Upgrade – See “To Install a Solaris Flash Archive With a Profile” in <i>Oracle Solaris 10 9/10 Installation Guide: Solaris Live Upgrade and Upgrade Planning</i>.

Administering Solaris Flash Archives

The `flar` command enables you to administer archives. You can split an archive into sections. Those sections can be modified, added to, or deleted, and then merged to create an archive. You can also obtain information about the archive.



Caution – Do not modify the Archive Files section or you compromise the integrity of the archive.

Splitting a Solaris Flash Archive

You can split an archive into sections, which enables you to modify some sections, add new sections, or delete sections. After you have modified the sections, you need to merge the sections to create a new archive. For example, you might want to add a user-defined section or modify the archive identification section. Do not modify the Archive Files section or you compromise the integrity of the archive.

The `flar split` command splits a Solaris Flash archive into sections. The `flar` command copies each section into a separate file in the current directory or specified directory. The files are named after the sections, for example, the archive cookie is saved in a file that is named `cookie`. You can specify that the `flar split` command save only one section. The syntax of the command is as follows:

```
flar split [-d dir] [-u section] [-f archive] [-S section] [-t [-p posn] [-b blocksize]] filename
```

`-d dir`

Retrieves the sections to copy from `dir`, rather than from the current directory.

<code>-u section</code>	<ul style="list-style-type: none"> ■ If you use this option, <code>flar</code> copies the cookie, identification, archive, and <i>section</i> sections. You can specify a single section name or a space-separated list of section names. ■ If you do <i>not</i> use this option, <code>flar</code> copies the cookie, identification, and archive sections only.
<code>-f archive</code>	Extracts the archive section into a directory that is named <code>archive</code> , rather than placing it in a file with the name <code>archive</code> .
<code>-S section</code>	Only copies the section that is named <i>section</i> from the archive. This section is user defined.

EXAMPLE 4-1 Splitting an Archive

In the following example, `archive1.flar` is split into three files:

- `cookie` – The first line of the archive, which identifies the version of the archive format. Do not change this identifier.
- `identification` – A copy of the archive identification section with all keyword-value pairs.
- `archive` – The archive itself. This file can be compressed.

```
# flar split archive1.flar
```

After the archive is split, you can modify the archive identification section or add a user-defined section. The sections can then be merged to re-create the archive.

Merging Solaris Flash Archives

After you have split an archive into sections, you can combine the sections to create a new archive.

The `flar combine` command creates a Solaris Flash archive from individual sections. The following table describes how the `flar` command handles combining sections.

Conditions	Description
Minimum number of files	Each section is assumed to be in a separate file, the names of which are the section names. These three files must be present: <ul style="list-style-type: none"> ■ Archive cookie (<code>cookie</code>) ■ Archive identification (<code>identification</code>) ■ Archive files (<code>archive</code>)

Conditions	Description
Archive copy method	<p>If archive is a directory, the contents are archived before including the directory in the combined archive by using the <code>cpio</code> copy utility.</p> <ul style="list-style-type: none"> ▪ <code>cpio</code> is the default copy method. Individual file sizes cannot be greater than 4 GB. ▪ <code>pax</code> is the copy method to handle large individual files. The <code>flarcreate</code> command with the <code>-L pax</code> option uses the <code>pax</code> utility to create an archive without limitations on individual file sizes. Individual file sizes can be greater than 4 GB.
Compressing an archive	If the archive identification section specifies to compress the archive, <code>flar</code> compresses the contents of the newly combined archive.
Validation	No validation is performed on any of the sections. In particular, no fields in the archive identification section are validated or updated.

The following command syntax is for `flar combine` command.

```
flar combine [-d dir] [-u section] [-t [-p posn] [-b blocksize]] filename
```

`-d dir` Retrieves the sections to combine from *dir*, rather than from the current directory.

`-u section`

- If you use this option, `flar` copies the cookie, identification, archive, and *section* sections. You can specify a single section name or a space-separated list of section names.
- If you do *not* use this option, `flar` copies the cookie, identification, and archive sections only.

EXAMPLE 4-2 Merging a Solaris Flash Archive

In this example, an archive cookie section, an archive identification section, and an archive files section are combined to become a complete archive. The archive is named `newarchive.flar`.

```
# flar combine newarchive.flar
```

EXAMPLE 4-3 Merging a Solaris Flash Archive and Adding a User-Defined Section

In this example, an archive cookie section, an archive identification section, an archive files section, and a user-defined section are combined to become a complete archive. The archive is named `newarchive.flar`. The user-defined section content is in the file that is named `user-defined`, which is in the current directory.

```
# flar combine -u user_defined newarchive.flar
```

Extracting Information From an Archive

Use the `flar info` command to obtain information about archives you have already created. The syntax of the command is as follows:

```
flar info [-l] [-k keyword] [-t [-p posn] [-b blocksize]] filename
```

`-k keyword` Returns only the value of the keyword *keyword*.

`-l` Lists all the files in the archive section.

EXAMPLE 4-4 Listing Files in an Archive Section

In this example, the command checks the file structure of the archive named `archive3.flar`.

```
# flar info -l archive3.flar
aaa
aaa/bbb
aaa/bbb/ccc
aaa/bbb/ccc/ddd
aaa/eee
```


Creating and Using a Disaster Recovery Image

You can use the following procedures to create a Flash Archive recovery image that can be used to restore a system to “factory fresh” condition.

- “Creating and Saving a FLAR Image” on page 63
- “Recovering the System Image From a FLAR Image” on page 65

Recovery Image Procedures

The following procedures provide the simplest instructions for creating a Flash Archive image that can be loaded onto the target system in order to recover from a failed disk drive.

▼ Creating and Saving a FLAR Image

Before You Begin To perform these procedures, you need access to the following:

- Either the initial boot media, such as the installation CD or DVD, or a netinstall service
- Off-system storage for the FLAR image

1 Record the partition table of the disk drive that the image is for.

This step assumes that the replacement disk drive will be the same size and partitioned identically to the original drive.

Use one of the following two methods to obtain information about the partition table on the disk drive.

- **As superuser (root), use the `format` command to print the partition table for the drive that the FLAR image will be taken from.**

```
# format
```

The `format` command provides the names of the partitions.

a. Select the boot drive from the list.

The first disk in the list is usually the boot drive.

b. Use the `partition` sub-command to access the Partition Menu.

c. Use the `quit` sub-command to exit from the Partition Menu.

d. Use the `quit` sub-command to exit from the Format application.

For further information, see the `format (1M)` man page.

■ **As superuser (`root`), use the `prtvtoc` command to generate the partition information.**

```
# prtvtoc /dev/dsk/c0t0d0s0
```

The `prtvtoc` command provides the size of the partitions by the number of cylinders for each partition.

Save the information to a safe location. You will use this information during the restoration of the system image during recovery.

2 Ensure there is adequate space for the FLAR image where it will be created.

The FLAR archive will require up to 15 GB of space without compression.

```
# df -h /tmp
```

Note – If you do not have sufficient space in `/tmp`, try a different filesystem, such as `/export`, instead. In which case, use the alternate filesystem, such as `/export`, instead of `/tmp` in the following steps.

3 Shutdown and reboot the system into single-user mode.

For instructions on shutting down a system, see “[How to Shut Down a Stand-Alone System](#)” in *System Administration Guide: Basic Administration*.

■ **For SPARC systems, boot as follows:**

```
>OK boot -s
```

For further information, see “[How to Boot a System to Run Level S \(Single-User Level\)](#)” in *System Administration Guide: Basic Administration*.

■ **For x86 systems, boot using instructions at “[How to Boot a System to Run Level S \(Single-User Level\)](#)” in *System Administration Guide: Basic Administration*.**

4 Create the FLAR archive.

Execute the `flarcreate` command as shown in the following example.

In this example, the FLAR image will be stored to a directory under /tmp named FLAR_recovery. The FLAR image will be named newsystem_recovery.flar.

```
# mkdir /FLAR_recovery
# flarcreate -n my_recovery_image -x /FLAR_recovery \
/FLAR_recovery/newsystem_recovery.flar
```

In this example:

- -n my_recovery_image implants a name into the FLAR image. The name should be something unique and meaningful to better identify it as the FLAR image for the system.
- -x /FLAR_recovery causes the /FLAR_recovery directory and its contents to be excluded from the FLAR image, since it will not be needed in the recovery image.

Note – By default, the flarcreate command ignores items that are located in “swap” partitions.

- /FLAR_recovery/newsystem_recovery.flar is the path and filename of the FLAR image. The filename should be something unique and meaningful to better identify it as the FLAR image for the system.

5 Save the FLAR image to a secure off-system location.

The FLAR image must be saved to a local storage device that is not the boot device, or to a remote location across NFS. The storage device, or remote location, must be accessible to the system at recovery time.

Copy the new FLAR to a safe location, as in this example.

```
# cp /FLAR_recovery/newsystem_recovery.flar \
/net/my-safe-machine/FLAR_image
```

▼ Recovering the System Image From a FLAR Image

The recovery process begins as a normal installation using whichever install method you choose. Instead of installing from the boot method, the installer is used to install from the FLAR image.

1 Begin the boot process using one of the following options.

- Use the initial boot media (installation CD/DVD).
- Use the netinstall service.

```
ok> boot net
```

2 Supply the network, date and time, and password information for the system.

- 3 When the Specify Media screen appears, select Network File System.**
- 4 In the Specify Network File System Path screen, provide the path to the off-system location of the FLAR image.**

For example, enter the following path.

```
/net/my-safe-machine/FLAR_image/newsystem_recovery.flar
```

The Flash Archive Summary screen is displayed.

- **If the file information is correct, click Next.**
- **If the file information is incorrect, click Deselect All Archives, and you will be able to repeat this step with the correct information.**

- 5 Specify the FLAR image location.**

From our example, the location would be the following.

```
my-safe-machine:/FLAR_image/newsystem_recovery.flar
```

- 6 At the Disk Selection screen, select the disk where the FLAR image is to be installed.**
- 7 Choose not to preserve existing data.**
- 8 At the File System and Disk Layout screen, select Customize to edit the disk slices to input the values of the disk partition table from the original disk.**

The partition table corresponds to each slice on the disk. Partition 0 from the partition table maps to Slice 0 (s0) on the hard drive.

- The slice sizes can be viewed in Cylinders to better match the output from the partition table. Select Cyl in the Partition form to view the form by cylinders.
- Do not change the size of Slice 2. It must span the entire disk regardless of the space being allocated.
- To get the Start and Size values for the Partition form, use the partition information that you recorded earlier when you ran the `prtvtoc` command. To get the value for Start in the Partition form, divide the First Sector value by the Sectors/Cylinder value, both found in the `prtvtoc` command output. The Size value in the Partition form is found by dividing the Sector Count by the Sectors/Cylinder value, information also provided by the `prtvtoc` command output.
- If the replacement disk has more storage space than the original disk, then it can be partitioned to use the space available. However, at least as much space for each partition must be allocated as was allocated on the original disk.

After the system reboots, the recovery is now complete.

9 (Optional) Rebuilding the Device Trees

The recovery instructions above assume that none of the hardware components have been added, removed or moved between the time that the recovery image was created and the time that a recovery is performed. If, however, a system has been recovered after hardware has been changed, then it is possible that the device trees (`/dev` and `/devices`) need to be updated. This update can be done using either a reconfiguration reboot of the system or by using the `devfsadm` command.

To rebuild the device trees, as a root-level user, use the `devfsadm` command as follows.

```
# devfsadm -C
```

Additional Resources

- *Using Flash Archive in the Solaris Operating System for Disaster Recovery*
- “Installing a ZFS Root File System (Oracle Solaris Flash Archive Installation)” in *Oracle Solaris ZFS Administration Guide*
- *Oracle Solaris 10 9/10 Installation Guide: Network-Based Installations*
- *Oracle Solaris 10 9/10 Installation Guide: Custom JumpStart and Advanced Installations*

Solaris Flash (Reference)

This chapter provides a description of Solaris Flash sections, keywords, and keyword values. Also, the chapter describes the `flar` command options.

For limitations when creating or installing a Solaris Flash archive, see [Table 2-1](#).

- “Solaris Flash Archive Section Descriptions” on page 69
- “Solaris Flash Keywords” on page 71
- “Solaris Flash `flar` Command” on page 75

Solaris Flash Archive Section Descriptions

Each Solaris Flash archive is grouped into sections. Some sections are generated by the Solaris Flash software and need no input from you. Some sections require input or optionally allow you to add information. The following table describes each section.

TABLE 6-1 Flash Archive Sections

Section Name	Description	Required by Archive?	Requires Input From User?
Archive cookie	The first section contains a cookie that identifies the file as a Solaris Flash archive. The deployment code uses the cookie for identification and validation purposes. The cookie must be present for an archive to be valid.	Yes	No

TABLE 6-1 Flash Archive Sections (Continued)

Section Name	Description	Required by Archive?	Requires Input From User?
Archive identification	<p>The second section contains keywords with values that provide identification information about the archive. The software generates some information such as the following:</p> <ul style="list-style-type: none"> ■ The archive ID number ■ The method of archival such as <code>cpio</code> ■ The creation date by default <p>You are required to specify a name for the Solaris Flash archive. Other information that you can specify about the archive includes the following:</p> <ul style="list-style-type: none"> ■ The author of the archive ■ The date that the archive was created ■ The name of the master system that you used to create the archive <p>For a list of keywords that describe the archive, see “Keywords for the Archive Identification Section” on page 71.</p>	Yes	Content is generated by both user and the software
Manifest	<p>A section of a Solaris Flash archive that is used to validate a clone system. The manifest section lists the files on a system to be retained, added to, or deleted from the clone system. The installation fails if the files do not match the expected file set. This section is informational only. The section lists the files in an internal format and cannot be used for scripting.</p> <p>You can exclude this section by creating the differential archive with the <code>flarccreate -M</code> option. Because no validation of the archive occurs, excluding this section is not recommended.</p>	No	No
Predeployment, Postdeployment, Reboot	<p>This section contains internal information that the flash software uses before and after installing an OS image. Any customization scripts that you have provided are stored in this section.</p>	Yes	No
Summary	<p>This section contains messages about the archive creation and records the activities of predeployment scripts.</p>	Yes	Content is generated by both user and the software
User-defined	<p>This section follows the archive-identification section. The archive can contain zero or more user-defined sections. These sections are not processed by the archive extraction code. These sections are retrieved separately and can be used for content descriptions.</p>	No	Yes
Archive files	<p>The archive files section contains the files that have been gathered from the master system in binary data. This section begins with <code>section_begin=archive</code>, but it does not have an ending section boundary.</p>	Yes	No

Solaris Flash Keywords

Solaris Flash keywords are like custom JumpStart keywords. They define elements of the installation. Each keyword is a command that controls one aspect of how the Solaris Flash software installs the software on a clone system.

Use the following guidelines to format keywords and values:

- Keywords and values are separated by a single equal sign with only one pair per line
- Keywords are case insensitive
- Individual lines can be any length

General Keywords

Each Solaris Flash archive section is defined by the `section_begin` and `section_end` keywords. For example, the archive files section includes a `section_begin` keyword, though with a different value. User-defined archive sections are delimited by `section_begin` and `section_end` keywords, with values appropriate to each section. The values for the `section_begin` and `section_end` keywords are described in the following table.

TABLE 6-2 Values for `section_begin` and `section_end` Keywords

Archive Section	Value for <code>section_begin</code> and <code>section_end</code> keywords
Archive cookie	<code>cookie</code> – This section is not delimited by the <code>section_begin</code> and <code>section_end</code> keywords.
Archive identification	<code>identification</code>
User-defined sections	<code>section_name</code> – An example of a <code>section_name</code> keyword is <code>X-user_section_1</code> .
Archive files	<code>archive</code>

Keywords for the Archive Identification Section

The following tables describe the keywords for use in the archive identification section and the values you can define for them.

Every section uses the keywords in [Table 6-3](#) to delimit each section.

TABLE 6-3 Archive Identification Section Keywords: General Keywords

Keywords	Value Definitions	Value	Required
section_begin section_end	These keywords are used to delimit sections in the archive and are not limited exclusively to the archive identification section. For a description of these keywords, see “General Keywords” on page 71.	Text	Yes

The following keywords, used in the archive-identification section, describe the contents of the archive files section.

TABLE 6-4 Archive Identification Section Keywords: Contents of Archive Files Section

Keywords	Value Definitions	Value	Required
archive_id (optional)	This keyword uniquely describes the contents of the archive. This value is used by the installation software only to validate the contents of the archive during archive installation. If the keyword is not present, no integrity check is performed. For example, the archive_id keyword might be FLASH-ARChive-2.0.	Text	No
files_archived_method	This keyword describes the archive method that is used in the files section. <ul style="list-style-type: none"> ■ If this keyword is not present, the files section is assumed to be in cpio format with ASCII headers. This format is the cpio -c option. ■ If this keyword is present, it has one of the following values: <ul style="list-style-type: none"> ■ cpio – The archive format in the files section is cpio with ASCII headers. ■ pax – The archive format in the files section is pax with extended tar interchange format. The pax utility enables archiving and extracting files that are greater than 4 GB. <p>If the files_compressed_method is present, the compression method is applied to the archive file that is created by the archive method.</p>	Text	No
files_archived_size	This keyword value is the size of the archived files section in bytes.	Numeric	No

TABLE 6-4 Archive Identification Section Keywords: Contents of Archive Files Section (Continued)

Keywords	Value Definitions	Value	Required
<code>files_compress_method</code>	<p>This keyword describes the compression algorithm that is used on the files section.</p> <ul style="list-style-type: none"> ■ If the keyword is present, it can have one of the following values. <ul style="list-style-type: none"> ■ <code>none</code> – The archive file section is not compressed. ■ <code>compress</code> – The file section is compressed by using the <code>compress</code> command. ■ If this keyword is not present, the archive files section is assumed to be uncompressed. <p>The compression method that is indicated by this keyword is applied to the archive file created by the archive method indicated by the <code>files_archived_method</code> keyword.</p>	Text	No
<code>files_unarchived_size</code>	This keyword defines the cumulative size in bytes of the extracted archive. The value is used for file-system size verification.	Numeric	No

The following keywords provide descriptive information about the entire archive. These keywords are generally used to assist you in archive selection and to aid in archive management. These keywords are all optional and are used to help you to distinguish between individual archives. You use options for the `flarcreate` command to include these keywords. For details, see [Example 3-9](#).

TABLE 6-5 Archive Identification Section Keywords: User Describes the Archive

Keywords	Value Definitions	Value	Required
<code>creation_date</code>	<p>This keyword value is a textual timestamp that represents the time that you created the archive.</p> <ul style="list-style-type: none"> ■ You can use the <code>flarcreate</code> command with the <code>-i</code> option to create the date. ■ If you do not specify a creation date with the <code>flarcreate</code> command, the default date is set in Greenwich mean time (GMT). ■ The value must be in ISO-8601 complete basic calendar format without the time designator (ISO-8601, §5.4.1(a)). The format is <code>CCYYMMDDhhmmss</code>. For example, <code>20000131221409</code> represents January 31, 2000, 10:14:09 p.m. 	Text	No
<code>creation_master</code>	This keyword value is the name of the master system you used to create the archive. You can use the <code>flarcreate -m</code> option to create this value. If you do not specify a value, the value is taken from the <code>uname -n</code> command.	Text	No

TABLE 6-5 Archive Identification Section Keywords: User Describes the Archive (Continued)

Keywords	Value Definitions	Value	Required
content_name	This keyword identifies the archive. The value is generated from the <code>flarc create -n</code> option. Follow these guidelines when you create this value: <ul style="list-style-type: none"> ▪ The descriptive name can be no longer than 256 characters. ▪ The description should contain the function and purpose of the archive. 	Text	Yes
content_type	This keyword value specifies a category for the archive. You use the <code>flarc create -T</code> option to generate the value.	Text	No
content_description	The keyword value describes the contents of the archive. The value of this keyword has no length limit. You use the <code>flarc create -E</code> option to create this value.	Text	No
content_author	This keyword value identifies the creator of the archive. You use the <code>flarc create -a</code> option to create this value. Suggested values include the full name of the creator and the creator's email address.	Text	No
content_architectures	This keyword value is a comma-separated list of the kernel architectures that the archive supports. <ul style="list-style-type: none"> ▪ If the keyword is present, the installation software validates the kernel architecture of the clone system against the list of architectures that the archive supports. The installation fails if the archive does not support the kernel architecture of the clone system. ▪ If the keyword is not present, the installation software does not validate the architecture of the clone system. 	Text list	No

The following keywords also describe the entire archive. By default, the values are filled in by `uname` when the flash archive is created. If you create a flash archive in which the root directory is not `/`, the archive software inserts the string `UNKNOWN` for the keywords. The exceptions are the `creation_node`, `creation_release`, and `creation_os_name` keywords.

- For `creation_node`, the software uses the contents of the `nodename` file.
- For `creation_release` and `creation_os_name`, the software attempts to use the contents of root directory `/var/sadm/system/admin/INST_RELEASE`. If the software is unsuccessful in reading this file, it assigns the value `UNKNOWN`.

Regardless of their sources, you cannot override the values of these keywords.

TABLE 6-6 Archive Identification Section Keywords: Software Describes the Archive

Keyword	Value
<code>creation_node</code>	The return from <code>uname -n</code>
<code>creation_hardware_class</code>	The return from <code>uname -m</code>
<code>creation_platform</code>	The return from <code>uname -i</code>
<code>creation_processor</code>	The return from <code>uname -p</code>
<code>creation_release</code>	The return from <code>uname -r</code>
<code>creation_os_name</code>	The return from <code>uname -s</code>
<code>creation_os_version</code>	The return from <code>uname -v</code>

User-Defined Section Keywords

In addition to the keywords that are defined by the Solaris Flash archive, you can define other keywords. The Solaris Flash archive ignores user-defined keywords, but you can provide scripts or programs that process the archive identification section and use user-defined keywords. Use the following format when creating user-defined keywords:

- Begin the keyword name with an `X`.
- Create the keyword with any characters other than linefeeds, equal signs, and null characters.
- Suggested naming conventions for user-defined keywords include the underscore-delimited descriptive method used for the predefined keywords. Another convention is a federated convention similar to the naming of Java packages.

For example, `X-department` is a valid name for a user-defined keyword.

For an example of using options to include user-defined keywords in the archive identification section, see [Example 3-9](#).

Solaris Flash `flar` Command

Use the Solaris Flash `flar` command to create a Solaris Flash archive and administer the archive.

flar Command

You can use the `flar` command with the following options:

- `flar create` creates an archive
- `flar combine` merges two archives
- `flar split` breaks an archive into sections
- `flar info` checks the structure of an archive

Use the `flar create` command to create a Solaris Flash archive from a master system. You can use this command when the master system is running in multiuser mode or single-user mode. You can also use `flar create` when the master system is booted from the following media.

- Solaris Operating System DVD

Note – Starting with the Oracle Solaris 10 9/10 release, only a DVD is provided. Solaris Software CDs are no longer provided.

- Solaris Software - 1 CD
- A Solaris network installation image of the DVD or CDs.

The master system should be in as stable a state as possible when you create a Solaris Flash archive.

Note – You can create a Solaris Flash archive by using either of these command options:

- As two words: `flar` with the `create` subcommand
 - As one word: `flar create`
-

The syntax of the command is as follows:

```
flarcreate -n archive_name [-R root] [-A unchanged_master_image_dir]
[-H][-I][-M][[-S]-c][[-t [-p posn] [-b blocksize]][-i date][-u section ...][-m
master][-f [list_filename] -] [-F][-a author][-e descr][-E descr_file][-L pax] [-T
type][-U key=val ...][-x exclude_dir/filename] [-y include_dir/filename] [-z
list_filename] [-X list_filename]path/filename
```

```
flar combine [-d dir] [-u section...] [-t [-p posn] path/filename
```

```
flar split [-d dir] [-u section...] [-f] [-S section] [-t [-p posn] path/filename
```

```
flar info [-l] [-k keyword] [-t [-p posn] path/filename
```

In the previous command lines, *path* is the directory in which you want the archive file to be saved. *filename* is the name of the archive file. If you do not specify a path, `flar create` saves the archive file in the current directory.

TABLE 6-7 Command-Line Options for the `flar` Command

Option	Description
Required Options	
-n <i>archive_name</i>	The value of this flag is the name of the archive. The <i>archive_name</i> you specify is the value of the <code>content_name</code> keyword.
Option for Compression	
-c	Compresses the archive by using <code>compress (1)</code> .
Options for Directories and Sizes	
-R <i>root</i>	Creates the archive from the file system tree that begins at the file system specified by <i>root</i> . If you do not specify this option, <code>flar create</code> creates an archive from a file system that begins at the root (<code>/</code>) file system.
-S	Omits sizing information in the archive.
-H	Does not generate the hash identifier.
Options for Creating a Differential Archive	
-A <i>unchanged_master_image_dir</i>	Creates a differential archive by comparing a new system image with the image that is specified by the <i>unchanged_master_image_dir</i> argument. By default, the new system image is root (<code>/</code>). You can change the default with the <code>-R</code> option. <i>unchanged_master_image_dir</i> is a directory where the unchanged master system image is stored or mounted through UFS, NFS, or <code>lumount</code> . You can modify the effects of file selection for a differential archive by using the options for contents selection described in the next section of the table.
-M	Excludes the manifest file. When you use this option, no validation occurs on the differential archive. When creating a differential archive, <code>flar create</code> creates a long list of the files in the system that are unchanged, are changed, and are to be deleted from the archive. This list is stored in the manifest section of the archive. When the differential archive is deployed, the software uses this list to perform a file-by-file check, ensuring the integrity of the clone system. Use of this option avoids such a check and saves the space that is used by the manifest section in a differential archive. However, you must consider the savings in time and disk space against the loss of an integrity check upon installation. Because no validation occurs, avoid using this option.
Options for Contents Selection	
Caution – Use the <code>flar create</code> file-exclusion options with caution. If you exclude some directories, others that you were unaware of might be left in the archive, such as system configuration files. The system would then be inconsistent and the installation would not work. Excluding directories and files is best used with data that can easily be removed without disrupting the system, such as large data files.	

TABLE 6-7 Command-Line Options for the `flar` Command (Continued)

Option	Description
<code>-y include_dir/filename</code>	<p>Adds to the archive those files and directories that are specified on the command line. This option is used when you have excluded a directory, but want to restore individual subdirectories or files.</p> <p><i>include_dir/filename</i> is the name of the subdirectory or file to be included.</p>
<code>-f list_filename</code>	<p>Adds files and directories from a list to the archive.</p> <p><i>list_filename</i> is the full path to a file that contains a list. The contents of the file are added to the file list unless <code>-F</code> is specified.</p> <ul style="list-style-type: none"> ■ The <i>list_filename</i> file must contain one file per line. ■ If you specify a file system with <code>-R root</code>, the path to each file must be relative to the alternate root directory or an absolute path. ■ If <i>filename</i> is “-”, <code>flar create</code> reads standard input as the list of files. When you use the value “-”, the archive size is not calculated.
<code>-F</code>	<p>Uses only the files in <code>-f list_filename</code> to create the archive. This option makes the <code>-f list_filename</code> the absolute list, rather than a list that is appended to the normal file list.</p>
<code>-x exclude_dir/filename</code>	<p>Excludes files and directories from the archive. These files and directories are specified at the command line. You can use multiple instances of this option to exclude more than one file or directory.</p> <p><i>exclude_dir/filename</i> is the name of the directory or file to be excluded.</p>
<code>-X list_filename</code>	<p>Excludes a list of files or directories from the archive.</p> <p><i>list_filename</i> is the full path to a file that contains the list.</p> <ul style="list-style-type: none"> ■ The <i>list_filename</i> file must contain one file per line. ■ If you specify a file system with <code>-R root</code>, the path to each file must be relative to the alternate root directory or an absolute path. ■ If <i>list_filename</i> is “-”, <code>flar create</code> reads standard input as the list of files. When you use the value “-”, the archive size is not calculated.
<code>-z list_filename</code>	<p>Excludes or includes a list of files or directories from the archive. Each file or directory in the list is noted with a plus “+” or minus “-”. A plus indicates an included file or directory and the minus indicates an excluded file or directory.</p> <p><i>list_filename</i> is the full path to a file that contains the list.</p> <ul style="list-style-type: none"> ■ The <i>list_filename</i> file must contain one file per line. ■ If you specify a file system with <code>-R root</code>, the path to each file must be relative to the alternate root directory or an absolute path.

TABLE 6-7 Command-Line Options for the `flar` Command (Continued)

Option	Description
-I	Override the integrity check. To prevent you from excluding important system files from an archive, <code>flar create</code> runs an integrity check. This check examines all files that are registered in a system package database and stops archive creation if any of them are excluded. Use of this option overrides this integrity check. Therefore, avoid the use of the -I option.
Options for Splitting and Merging Archives	
-d <i>dir</i>	Retrieves the sections to copy from <i>dir</i> , rather than from the current directory.
-u <i>section</i>	<ul style="list-style-type: none"> ■ If you use this option, <code>flar</code> copies the cookie, identification, archive, and <i>section</i> sections. You can specify a single section name or a space-separated list of section names. ■ If you do <i>not</i> use this option, <code>flar</code> copies the cookie, identification, and archive sections only.
-f <i>archive</i>	Extracts the archive section into a directory that is named <i>archive</i> , rather than placing it in a file with the name <i>archive</i> . Used for splitting an archive.
-S <i>section</i>	Only copies the section that is named <i>section</i> from the archive. This section is user defined. Used for splitting an archive.
Option Used To Copy Files (Archive)	
-L <i>pax</i>	The <code>cpio</code> utility is the default copy method. If you have large individual files, the -L <i>pax</i> option uses the <code>pax</code> utility to create an archive without limitations on individual file sizes. Individual file sizes can be greater than 4 GB.
Options Used With User-Defined Sections	
-u <i>section</i>	Includes <i>section</i> as a user-defined section. To include more than one user-defined section, <i>section</i> must be a space-separated list of section names.
-d <i>dir</i>	Retrieves the section file that is specified with -u from <i>dir</i> .
Options Used With Tape Archives	
-t	Creates an archive on a tape device. The <i>filename</i> argument is the name of the tape device.
-p <i>posn</i>	Use only with the -t option. Specifies the position on the tape device for <code>flar create</code> to store the archive. If you do not use this option, <code>flar create</code> places the archive at the current position of the tape.
-b <i>blocksize</i>	Specifies the block size <code>flar create</code> uses when creating the archive. If you do not specify a block size, <code>flar create</code> uses the default block size of 64 KB.
Options for Archive Identification	
These keywords and values appear in the archive identification section of the archive.	

TABLE 6-7 Command-Line Options for the `flar` Command (Continued)

Option	Description
<code>-U key=val</code>	Includes user-defined keywords and values in the archive identification section.
<code>-i date</code>	Uses <i>date</i> as the value for the <code>creation_date</code> keyword. If you do not specify a date, <code>flar create</code> uses the current system time and date.
<code>-m master</code>	Uses <i>master</i> as the name of the master system on which you created the archive. <i>master</i> is the value for the <code>creation_master</code> keyword. If you do not specify <i>master</i> , <code>flar create</code> uses the system name that is reported by <code>uname -n</code> .
<code>-e descr</code>	Uses <i>descr</i> for the value of the <code>content_description</code> keyword. You cannot use this option when you use the <code>-E</code> option.
<code>-E descr_file</code>	Retrieves the value for the <code>content_description</code> keyword from the <i>descr_file</i> file. You cannot use this option when you use the <code>-e</code> option.
<code>-a author</code>	Uses <i>author</i> as the author name in the archive identification section. <i>author</i> is the value for the <code>content_author</code> keyword. If you do not specify an author, <code>flar create</code> does not include the <code>content_author</code> keyword in the archive identification section.
<code>-T type</code>	Uses <i>type</i> as the value for the <code>content_type</code> keyword. <i>type</i> is user defined. If you do not specify a <i>type</i> , <code>flar create</code> does not include the <code>content_type</code> keyword.

Glossary

3DES	([Triple DES] Triple-Data Encryption Standard). A symmetric-key encryption method that provides a key length of 168 bits.
AES	(Advanced Encryption Standard) A symmetric 128-bit block data encryption technique. The U.S. government adopted the Rijndael variant of the algorithm as its encryption standard in October 2000. AES replaces DES encryption as the government standard.
archive	<p>A file that contains a collection of files that were copied from a master system. The file also contains identification information about the archive, such as a name and the date that you created the archive. After you install an archive on a system, the system contains the exact configuration of the master system.</p> <p>An archive could be a differential archive, which is a Solaris Flash archive that contains only the differences between two system images, an unchanged master image and an updated master image. The differential archive contains files to be retained, modified, or deleted from the clone system. A differential update changes only the files specified and is restricted to systems that contain software consistent with the unchanged master image.</p>
arrow keys	One of the four directional keys on the numeric keypad.
begin script	A user-defined Bourne shell script, specified within the <code>rules</code> file, that performs tasks before the Solaris software is installed on the system. You can use begin scripts only with custom JumpStart installations.
boot	To load the system software into memory and start it.
boot archive	<p>x86 only: A boot archive is a collection of critical files that is used to boot the Solaris OS. These files are needed during system startup before the root (<code>/</code>) file system is mounted. Two boot archives are maintained on a system:</p> <ul style="list-style-type: none">■ The boot archive that is used to boot the Solaris OS on a system. This boot archive is sometimes called the primary boot archive.■ The boot archive that is used for recovery when the primary boot archive is damaged. This boot archive starts the system without mounting the root (<code>/</code>) file system. On the GRUB menu, this boot archive is called failsafe. The archive's essential purpose is to regenerate the primary boot archive, which is usually used to boot the system.
boot environment	A collection of mandatory file systems (disk slices and mount points) that are critical to the operation of the Solaris OS. These disk slices might be on the same disk or distributed across multiple disks.

The active boot environment is the one that is currently booted. Exactly one active boot environment can be booted. An inactive boot environment is not currently booted, but can be in a state of waiting for activation on the next reboot.

boot loader	x86 only: The boot loader is the first software program that runs after you turn on a system. This program begins the booting process.
boot server	A server system that provides client systems on the same network subnet with the programs and information that they need to start. A boot server is required to install over the network if the install server is on a different subnet than the systems on which Solaris software is to be installed.
bootlog-cgi program	The CGI program that enables a web server to collect and store remote client-booting and installation console messages during a WAN boot installation.
certificate authority	(CA) A trusted third-party organization or company that issues digital certificates that are used to create digital signatures and public-private key pairs. The CA guarantees that the individual who is granted the unique certificate is who she or he claims to be.
certstore file	A file that contains a digital certificate for a specific client system. During an SSL negotiation, the client might be asked to provide the certificate file to the server. The server uses this file to verify the identity of the client.
CGI	(Common Gateway Interface) An interface by which external programs communicate with the HTTP server. Programs that are written to use CGI are called CGI programs or CGI scripts. CGI programs handle forms or parse output the server does not normally handle or parse.
checksum	The result of adding a group of data items that are used for checking the group. The data items can be either numerals or other character strings that are treated as numerals during the checksum calculation. The checksum value verifies that communication between two devices is successful.
client	In the client-server model for communications, the client is a process that remotely accesses resources of a compute server, such as compute power and large memory capacity.
clone system	A system that you install by using a Solaris Flash archive. The clone system has the same installation configuration as the master system.
cluster	A logical collection of packages (software modules). The Solaris software is divided into <i>software groups</i> , which are each composed of clusters and <i>packages</i> .
command line	A string of characters that begins with a command, often followed by arguments, including options, file names, and other expressions, and terminated by the end-of-line character.
concatenation	A RAID-0 volume. If slices are concatenated, the data is written to the first available slice until that slice is full. When that slice is full, the data is written to the next slice, serially. A concatenation provides no data redundancy unless it is contained in a mirror. See also RAID-0 volume.
Core Software Group	A software group that contains the minimum software that is required to boot and run the Solaris OS on a system. Core includes some networking software and the drivers that are required to run the Common Desktop Environment (CDE) desktop. Core does not include the CDE software.

critical file systems	File systems that are required by the Solaris OS. When you use Solaris Live Upgrade, these file systems are separate mount points in the <code>vfstab</code> file of the active and inactive boot environments. Example file systems are <code>root (/)</code> , <code>/usr</code> , <code>/var</code> , and <code>/opt</code> . These file systems are always copied from the source to the inactive boot environment.
custom JumpStart	A type of installation in which the Solaris software is automatically installed on a system that is based on a user-defined profile. You can create customized profiles for different types of users and systems. A custom JumpStart installation is a JumpStart installation you create.
custom probes file	A file, which must be located in the same JumpStart directory as the <code>rules</code> file, that is a Bourne shell script that contains two types of functions: probe and comparison. Probe functions gather the information you want or do the actual work and set a corresponding <code>SI_</code> environment variable you define. Probe functions become probe keywords. Comparison functions call a corresponding probe function, compare the output of the probe function, and return 0 if the keyword matches or 1 if the keyword doesn't match. Comparison functions become rule keywords. See also <i>rules file</i> .
dataset	A generic name for the following ZFS entities: clones, file systems, snapshots, or volumes.
decryption	The process of converting coded data to plain text. See also encryption .
derived profile	A profile that is dynamically created by a <code>begin</code> script during a custom JumpStart installation.
DES	(Data Encryption Standard) A symmetric-key encryption method that was developed in 1975 and standardized by ANSI in 1981 as ANSI X.3.92. DES uses a 56-bit key.
Developer Solaris Software Group	A software group that contains the End User Solaris Software Group plus the libraries, include files, man pages, and programming tools for developing software.
DHCP	(Dynamic Host Configuration Protocol) An application-layer protocol. Enables individual computers, or clients, on a TCP/IP network to extract an IP address and other network configuration information from a designated and centrally maintained DHCP server or servers. This facility reduces the overhead of maintaining and administering a large IP network.
differential archive	A Solaris Flash archive that contains only the differences between two system images, an unchanged master image and an updated master image. The differential archive contains files to be retained, modified, or deleted from the clone system. A differential update changes only the files that are specified and is restricted to systems that contain software consistent with the unchanged master image.
digital certificate	A nontransferable, nonforgeable, digital file issued from a third party that both communicating parties already trust.
disc	An optical disc, as opposed to a magnetic disk, which recognizes the common spelling that is used in the compact disc (CD) market. For example, a CD-ROM or DVD-ROM is an optical disc.
disk	A round platter, or set of platters, of a magnetized medium that is organized into concentric tracks and sectors for storing data such as files. See also <code>disc</code> .
disk configuration file	A file that represents a structure of a disk (for example, bytes/sector, flags, slices). Disk configuration files enable you to use the <code>pfinstall</code> command from a single system to test profiles on different-size disks.
diskless client	A client on a network that relies on a server for all of its disk storage.

document root directory	The root of a hierarchy on a web server machine that contains the files, images, and data you want to present to users who are accessing the web server.
domain	A part of the Internet naming hierarchy. A domain represents a group of systems on a local network that share administrative files.
domain name	The name that is assigned to a group of systems on a local network that share administrative files. The domain name is required for the Network Information Service (NIS) database to work properly. A domain name consists of a sequence of component names that are separated by periods (for example: <code>tundra.mpk.ca.us</code>). As you read a domain name from left to right, the component names identify more general (and usually remote) areas of administrative authority.
encryption	The process of protecting information from unauthorized use by making the information unintelligible. Encryption is based on a code, called a key, which is used to decrypt the information. See also decryption .
End User Solaris Software Group	A software group that contains the Core Software Group plus the recommended software for an end user, including the Common Desktop Environment (CDE) and DeskSet software.
Entire Solaris Software Group	A software group that contains the entire Solaris release.
Entire Solaris Software Group Plus OEM Support	A software group that contains the entire Solaris release plus additional hardware support for OEMs. This software group is recommended when installing Solaris software on SPARC based servers. In order for a Flash archive to be installed on different system types, the Entire Plus OEM distribution needs to be installed on the master system.
/etc directory	A directory that contains critical system configuration files and maintenance commands.
/etc/netboot directory	The directory on a WAN boot server that contains the client configuration information and security data that are required for a WAN boot installation.
/export file system	A file system on an OS server that is shared with other systems on a network. For example, the <code>/export</code> file system can contain the root (<code>/</code>) file system and swap space for diskless clients and the home directories for users on the network. Diskless clients rely on the <code>/export</code> file system on an OS server to boot and run.
failsafe boot archive	x86 only: A boot archive that is used for recovery when the primary boot archive is damaged. This boot archive starts the system without mounting the root (<code>/</code>) file system. This boot archive is called failsafe on the GRUB menu. The archive's essential purpose is to regenerate the primary boot archive, which is usually used to boot the system. See <i>boot archive</i> .
fallback	A reversion to the environment that ran previously. Use fallback when you are activating an environment and the boot environment that is designated for booting fails or shows some undesirable behavior.
fdisk partition	A logical partition of a disk drive that is dedicated to a particular operating system on x86 based systems. To install the Solaris software, you must set up at least one Solaris <code>fdisk</code> partition on an x86 based system. x86 based systems allow up to four different <code>fdisk</code> partitions on a disk. These partitions can be used to hold individual operating systems. Each operating system must be located on a unique <code>fdisk</code> partition. A system can only have one Solaris <code>fdisk</code> partition per disk.
file server	A server that provides the software and file storage for systems on a network.
file system	In the SunOS operating system, a tree-structured network of files and directories that you can access.

finish script	A user-defined Bourne shell script, specified within the <code>rules</code> file, that performs tasks after the Solaris software is installed on the system but before the system reboots. You use finish scripts with custom JumpStart installations.
format	To put data into a structure or divide a disk into sectors for receiving data.
function key	One of the 10 or more keyboard keys that are labeled F1, F2, F3, and so on that are mapped to particular tasks.
global zone	In Solaris Zones, the global zone is both the default zone for the system and the zone used for system-wide administrative control. The global zone is the only zone from which a non-global zone can be configured, installed, managed, or uninstalled. Administration of the system infrastructure, such as physical devices, routing, or dynamic reconfiguration (DR), is only possible in the global zone. Appropriately privileged processes running in the global zone can access objects associated with other zones. See also <i>Solaris Zones</i> and <i>non-global zone</i> .
GRUB	x86 only: GNU GRand Unified Bootloader (GRUB) is an open source boot loader with a simple menu interface. The menu displays a list of operating systems that are installed on a system. GRUB enables you to easily boot these various operating systems, such as the Solaris OS, Linux, or Microsoft Windows.
GRUB edit menu	x86 only: A boot menu that is a submenu of the GRUB main menu. GRUB commands are displayed on this menu. These commands can be edited to change boot behavior.
GRUB main menu	x86 only: A boot menu that lists the operating systems that are installed on a system. From this menu, you can easily boot an operating system without modifying the BIOS or <code>fdisk</code> partition settings.
hard link	A directory entry that references a file on disk. More than one such directory entry can reference the same physical file.
hash	A number that is produced by taking some input and generating a number that is significantly shorter than the input. The same output value is always generated for identical inputs. Hash functions can be used in table search algorithms, in error detection, and in tamper detection. When used for tamper detection, hash functions are chosen such that it is difficult to find two inputs that yield the same hash result. MD5 and SHA-1 are examples of one-way hash functions. For example, a message digest takes a variable-length input such as a disk file and reduces it to a small value.
hashing	The process of changing a string of characters into a value or key that represents the original string.
HMAC	Keyed hashing method for message authentication. HMAC is used with an iterative cryptographic hash function, such as MD5 or SHA-1, in combination with a secret shared key. The cryptographic strength of HMAC depends on the properties of the underlying hash function.
host name	The name by which a system is known to other systems on a network. This name must be unique among all the systems within a particular domain (usually, this means within any single organization). A host name can be any combination of letters, numbers, and minus signs (-), but it cannot begin or end with a minus sign.
HTTP	(Hypertext Transfer Protocol) (n.) The Internet protocol that fetches hypertext objects from remote hosts. This protocol is based on TCP/IP.
HTTPS	A secure version of HTTP, implemented by using the Secure Sockets Layer (SSL).

initial installation	<p>An installation that overwrites the currently running software or initializes a blank disk.</p> <p>An initial installation of the Solaris OS overwrites the system's disk or disks with the new version of the Solaris OS. If your system is not running the Solaris OS, you must perform an initial installation. If your system is running an upgradable version of the Solaris OS, an initial installation overwrites the disk and does not preserve the OS or local modifications.</p>
install server	<p>A server that provides the Solaris DVD or CD images from which other systems on a network can install Solaris (also called a <i>media server</i>). You can create an install server by copying the Solaris DVD or CD images to the server's hard disk.</p>
IPv6	<p>IPv6 is a version (version 6) of Internet Protocol (IP) that is designed to be an evolutionary step from the current version, IPv4 (version 4). Deploying IPv6, by using defined transition mechanisms, does not disrupt current operations. In addition, IPv6 provides a platform for new Internet functionality.</p>
job	<p>A user-defined task to be completed by a computer system.</p>
JumpStart directory	<p>When you use a profile diskette for custom JumpStart installations, the JumpStart directory is the root directory on the diskette that contains all the essential custom JumpStart files. When you use a profile server for custom JumpStart installations, the JumpStart directory is a directory on the server that contains all the essential custom JumpStart files.</p>
JumpStart installation	<p>A type of installation in which the Solaris software is automatically installed on a system by using the factory-installed JumpStart software.</p>
Kerberos	<p>A network authentication protocol that uses strong, secret-key cryptography to enable a client and server to identify themselves to each other over an insecure network connection.</p>
key	<p>The code for encrypting or decrypting data. See also encryption.</p>
keystore file	<p>A file that contains keys shared by a client and server. During a WAN boot installation, the client system uses the keys to verify the integrity of, or decrypt the data and files transmitted from, the server.</p>
LAN	<p>(local area network) A group of computer systems in close proximity that can communicate by way of some connecting hardware and software.</p>
LDAP	<p>(Lightweight Directory Access Protocol) A standard, extensible directory access protocol that is used by LDAP naming service clients and servers to communicate with each other.</p>
locale	<p>A geographic or political region or community that shares the same language, customs, or cultural conventions (English for the U.S. is <code>en_US</code>, and English for the U.K. is <code>en_UK</code>).</p>
logical device	<p>A group of physical slices on one or more disks that appear to the system as a single device. A logical device is called a volume in Solaris Volume Manager. A volume is functionally identical to a physical disk for the purposes of an application or file system.</p>
manifest section	<p>A section of a Solaris Flash archive that is used to validate a clone system. The manifest section lists the files on a system to be retained, added to, or deleted from the clone system. This section is informational only. The section lists the files in an internal format and cannot be used for scripting.</p>
master system	<p>A system that you use to create a Solaris Flash archive. The system configuration is saved in the archive.</p>

MD5	(Message Digest 5) An iterative cryptographic hash function that is used for message authentication, including digital signatures. The function was developed in 1991 by Rivest.
media server	See <i>install server</i> .
menu.lst file	x86 only: A file that lists all the operating systems that are installed on a system. The contents of this file dictate the list of operating systems that is displayed on the GRUB menu. From the GRUB menu, you can easily boot an operating system without modifying the BIOS or <code>fdisk</code> partition settings.
metadevice	See <i>volume</i> .
miniroot	A minimal, bootable root (<code>/</code>) file system that is included in Solaris installation media. A miniroot consists of the Solaris software that is required to install and upgrade systems. On x86 based systems, the miniroot is copied to the system to be used as the failsafe boot archive. See <i>failsafe boot archive</i> .
mirror	See <i>RAID-1 volume</i> .
mount	The process of accessing a directory from a disk that is attached to a machine that is making the mount request or a remote disk on a network. To mount a file system, you need a mount point on the local system and the name of the file system to be mounted (for example, <code>/usr</code>).
mount point	A workstation directory to which you mount a file system that exists on a remote machine.
name server	A server that provides a naming service to systems on a network.
naming service	A distributed network database that contains key system information about all the systems on a network so that the systems can communicate with each other. With a naming service, the system information can be maintained, managed, and accessed on a network-wide basis. Without a naming service, each system has to maintain its own copy of the system information in the local <code>/etc</code> files. Sun supports the following naming services: LDAP, NIS, and NIS+.
network installation	A way to install software over the network from a system with a CD-ROM or DVD-ROM drive to a system without a CD-ROM or DVD-ROM drive. Network installations require a <i>name server</i> and an <i>install server</i> .
networked systems	A group of systems (called hosts) that are connected through hardware and software so that they can communicate and share information. Referred to as a local area network (LAN). One or more servers are usually needed when systems are networked.
NIS	The SunOS 4.0 (minimum) Network Information Service. A distributed network database that contains key information about the systems and the users on the network. The NIS database is stored on the master server and all the slave servers.
NIS+	The SunOS 5.0 (minimum) Network Information Service. NIS+ replaces NIS, the SunOS 4.0 (minimum) Network Information Service.
non-global zone	A virtualized operating system environment created within a single instance of the Solaris Operating System. One or more applications can run in a non-global zone without interacting with the rest of the system. Non-global zones are also called zones. See also <i>Solaris Zones</i> and <i>global zone</i> .
nonnetworked systems	Systems that are not connected to a network or do not rely on other systems.

/opt file system	A file system that contains the mount points for third-party and unbundled software.
OS server	A system that provides services to systems on a network. To serve diskless clients, an OS server must have disk space set aside for each diskless client's root (<i>/</i>) file system and swap space (<i>/export/root</i> , <i>/export/swap</i>).
package	A collection of software that is grouped into a single entity for modular installation. The Solaris software is divided into <i>software groups</i> , which are each composed of <i>clusters</i> and packages.
panel	A container for organizing the contents of a window, a dialog box, or applet. The panel might collect and confirm user input. Panels might be used by wizards and follow an ordered sequence to fulfill a designated task.
patch analyzer	A script that you can run manually or as part of the Solaris installation program. The patch analyzer performs an analysis on your system to determine which (if any) patches will be removed by upgrading to a Solaris update.
platform group	A vendor-defined grouping of hardware platforms for the purpose of distributing specific software. Examples of valid platform groups are <i>i86pc</i> and <i>sun4u</i> .
platform name	The output of the <code>uname -i</code> command. For example, the platform name for the Ultra 60 is <i>SUNW,Ultra-60</i> .
pool	A logical group of devices describing the layout and physical characteristics of the available ZFS storage. Space for datasets is allocated from a pool.
Power Management	<p>Software that automatically saves the state of a system and turns it off after it is idle for 30 minutes. When you install the Solaris software on a system that complies with Version 2 of the U.S. Environmental Protection Agency's Energy Star guidelines, the Power Management software is installed by default. A sun4u SPARC based system is an example of a system that has Power Management installed by default. After a subsequent reboot, you are prompted to enable or disable the Power Management software.</p> <p>Energy Star guidelines require that systems or monitors automatically enter a "sleep state" (consume 30 watts or less) after the system or monitor becomes inactive.</p>
primary boot archive	A boot archive that is used to boot the Solaris OS on a system. This boot archive is sometimes called the primary boot archive. See <i>boot archive</i> .
private key	The decryption key used in public-key encryption.
probe keyword	A syntactical element that extracts attribute information about a system when using the custom JumpStart method to install. A probe keyword does not require you to set up a matching condition and run a profile as required for a rule. See also <i>rule</i> .
profile	A text file that defines how to install the Solaris software when using the custom JumpStart method. For example, a profile defines which software group to install. Every rule specifies a profile that defines how a system is to be installed when the rule is matched. You usually create a different profile for every rule. However, the same profile can be used in more than one rule. See also <i>rules file</i> .
profile diskette	A diskette that contains all the essential custom JumpStart files in its root directory (JumpStart directory).
profile server	A server that contains all the essential custom JumpStart files in a JumpStart directory.

public key	The encryption key used in public-key encryption.
public-key cryptography	A cryptographic system that uses two keys: a public key known to everyone, and a private key known only to the recipient of the message.
RAID-0 volume	A class of volume that can be a stripe or a concatenation. These components are also called submirrors. A stripe or concatenation is the basic building block for mirrors.
RAID-1 volume	A class of volume that replicates data by maintaining multiple copies. A RAID-1 volume is composed of one or more RAID-0 volumes called <i>submirrors</i> . A RAID-1 volume is sometimes called a <i>mirror</i> .
RAID-Z storage pool	A virtual device that stores data and parity on multiple disks that can be used as a ZFS storage pool. RAID-Z is similar to RAID-5.
Reduced Network Support Software Group	A software group that contains the minimum code that is required to boot and run a Solaris system with limited network service support. The Reduced Networking Software Group provides a multiuser text-based console and system administration utilities. This software group also enables the system to recognize network interfaces, but does not activate network services.
resilvering	For a ZFS storage pool, the process of transferring data from one device to another device is known as resilvering. For example, if a mirror component is replaced or taken offline, the data from the up-to-date mirror component is copied to the newly restored mirror component. This process is referred to as mirror resynchronization in traditional volume management products. For more information about ZFS resilvering, see “ Replacing a Device in a ZFS Storage Pool ” in <i>Oracle Solaris ZFS Administration Guide</i>
root	The top level of a hierarchy of items. Root is the one item from which all other items are descended. See <i>root directory</i> or <i>root (/) file system</i> .
root (/) file system	The top-level file system from which all other file systems stem. The root (/) file system is the base on which all other file systems are mounted, and is never unmounted. The root (/) file system contains the directories and files critical for system operation, such as the kernel, device drivers, and the programs that are used to start (boot) a system.
root directory	The top-level directory from which all other directories stem.
rule	A series of values that assigns one or more system attributes to a profile. A rule is used in a custom JumpStart installation.
rules file	A text file that contains a rule for each group of systems or single systems that you want to install automatically. Each rule distinguishes a group of systems, based on one or more system attributes. The <code>rules</code> file links each group to a profile, which is a text file that defines how the Solaris software is to be installed on each system in the group. A <code>rules</code> file is used in a custom JumpStart installation. See also <i>profile</i> .
rules.ok file	A generated version of the <code>rules</code> file. The <code>rules.ok</code> file is required by the custom JumpStart installation software to match a system to a profile. You <i>must</i> use the <code>check</code> script to create the <code>rules.ok</code> file.
Secure Sockets Layer	(SSL) A software library establishing a secure connection between two parties (client and server) used to implement HTTPS, the secure version of HTTP.
server	A network device that manages resources and supplies services to a client.

SHA1	(Secure Hashing Algorithm) The algorithm that operates on any input length less than 2^{64} to produce a message digest.
shareable file systems	File systems that are user-defined files such as <code>/export/home</code> and <code>/swap</code> . These file systems are shared between the active and inactive boot environment when you use Solaris Live Upgrade. Shareable file systems contain the same mount point in the <code>vfstab</code> file in both the active and inactive boot environments. Updating shared files in the active boot environment also updates data in the inactive boot environment. Shareable file systems are shared by default, but you can specify a destination slice, and then the file systems are copied.
slice	The unit into which the disk space is divided by the software.
snapshot	A read-only image of a ZFS file system or volume at a given point in time.
software group	A logical grouping of the Solaris software (clusters and packages). During a Solaris installation, you can install one of the following software groups: Core, End User Solaris Software, Developer Solaris Software, or Entire Solaris Software, and for SPARC systems only, Entire Solaris Software Group Plus OEM Support.
Solaris DVD or CD images	The Solaris software that is installed on a system, which you can access on the Solaris DVDs or CDs or an install server's hard disk to which you have copied the Solaris DVD or CD images.
Solaris Flash	A Solaris installation feature that enables you to create an archive of the files on a system, called the <i>master system</i> . You can then use the archive to install other systems, making the other systems identical in their configuration to the master system. See also <i>archive</i> .
Solaris installation program	A graphical user interface (GUI) or command-line interface (CLI) installation program that uses wizard panels to guide you step-by-step through installing the Solaris software and third-party software.
Solaris Live Upgrade	An upgrade method that enables a duplicate boot environment to be upgraded while the active boot environment is still running, thus eliminating downtime of the production environment.
Solaris Zones	A software partitioning technology used to virtualize operating system services and provide an isolated and secure environment for running applications. When you create a non-global zone, you produce an application execution environment in which processes are isolated from all other zones. This isolation prevents processes that are running in a zone from monitoring or affecting processes that are running in any other zones. See also <i>global zone</i> and <i>non-global zone</i> .
standalone	A computer that does not require support from any other machine.
state database	A database that stores information about the state of your Solaris Volume Manager configuration. The state database is a collection of multiple, replicated database copies. Each copy is referred to as a <i>state database replica</i> . The state database tracks the location and status of all known state database replicas.
state database replica	A copy of a state database. The replica ensures that the data in the database is valid.
submirror	See <i>RAID-0 volume</i> .
subnet	A working scheme that divides a single logical network into smaller physical networks to simplify routing.
subnet mask	A bit mask that is used to select bits from an Internet address for subnet addressing. The mask is 32 bits long and selects the network portion of the Internet address and 1 or more bits of the local portion.

superuser	A special user who has privileges to perform all administrative tasks on the system. The superuser has the ability to read and write to any file, run all programs, and send kill signals to any process.
swap space	A slice or file that temporarily holds the contents of a memory area till it can be reloaded in memory. Also called the <code>/swap</code> or <code>swap</code> volume.
sysidcfg file	A file in which you specify a set of special system configuration keywords that preconfigure a system.
system configuration file	(<code>system.conf</code>) A text file in which you specify the locations of the <code>sysidcfg</code> file and the custom JumpStart files you want to use in a WAN boot installation.
time zone	Any of the 24 longitudinal divisions of the earth's surface for which a standard time is kept.
truststore file	A file that contains one or more digital certificates. During a WAN boot installation, the client system verifies the identity of the server that is trying to perform the installation by consulting the data in the <code>truststore</code> file.
unmount	The process of removing access to a directory on a disk that is attached to a machine or to a remote disk on a network.
update	An installation, or to perform an installation, on a system that changes software that is of the same type. Unlike an upgrade, an update might downgrade the system. Unlike an initial installation, software of the same type that is being installed must be present before an update can occur.
upgrade	An installation that merges files with existing files and preserves modifications where possible. An upgrade of the Solaris OS merges the new version of the Solaris OS with the existing files on the system's disk or disks. An upgrade saves as many modifications as possible that you have made to the previous version of the Solaris OS.
upgrade option	An option that is presented by the Solaris installation program. The upgrade procedure merges the new version of Solaris with existing files on your disk or disks. An upgrade also saves as many local modifications as possible since the last time Solaris was installed.
URL	(Uniform Resource Locator) The addressing system used by the server and the client to request documents. A URL is often called a location. The format of a URL is <i>protocol://machine:port/document</i> . A sample URL is <code>http://www.example.com/index.html</code> .
/usr file system	A file system on a standalone system or server that contains many of the standard UNIX programs. Sharing the large <code>/usr</code> file system with a server rather than maintaining a local copy minimizes the overall disk space that is required to install and run the Solaris software on a system.
utility	A standard program, usually furnished at no charge with the purchase of a computer, that does the computer's housekeeping.
/var file system	A file system or directory (on standalone systems) that contains system files that are likely to change or grow over the life of the system. These files include system logs, <code>vi</code> files, mail files, and UUCP files.
virtual device	A logical device in a ZFS pool, which can be a physical device, a file, or a collection of devices.
volume	A group of physical slices or other volumes that appear to the system as a single logical device. A volume is functionally identical to a physical disk for the purposes of an application or file system.

In some command-line utilities, a volume is called a metadvice. Volume is also called *pseudo device* or *virtual device* in standard UNIX terms.

Volume Manager	A program that provides a mechanism to administer and obtain access to the data on DVD-ROMs, CD-ROMs, and diskettes.
WAN	(wide area network) A network that connects multiple local area networks (LANs) or systems at different geographical sites by using telephone, fiber-optic, or satellite links.
WAN boot installation	A type of installation that enables you to boot and install software over a wide area network (WAN) by using HTTP or HTTPS. The WAN boot installation method enables you to transmit an encrypted Solaris Flash archive over a public network and perform a custom JumpStart installation on a remote client.
WAN boot miniroot	A miniroot that has been modified to perform a WAN boot installation. The WAN boot miniroot contains a subset of the software in the Solaris miniroot. See also miniroot .
WAN boot server	A web server that provides the configuration and security files that are used during a WAN boot installation.
wanboot -cgi program	The CGI program that retrieves and transmits the data and files that are used in a WAN boot installation.
wanboot.conf file	A text file in which you specify the configuration information and security settings that are required to perform a WAN boot installation.
wanboot program	The second-level boot program that loads the WAN boot miniroot, client configuration files, and installation files that are required to perform a WAN boot installation. For WAN boot installations, the wanboot binary performs tasks similar to the ufsboot or inetboot second-level boot programs.
ZFS	A file system using storage pools to manage physical storage.
zone	See <i>non-global zone</i>

Index

A

- administering, Solaris Flash archives, 58
- archive
 - See also* scripts
 - administering, 58
 - compressing, 33
 - creating an archive, 41
 - examples, 43
 - requirements for platforms, 27
 - customizing
 - description, 30
 - with scripts, 31
 - description of process, 15–20
 - `flarcreate` command, 76–80
 - getting information, 61
 - installing
 - description, 15–20
 - how to install, 55–61
 - installation programs, 33–34
 - Solaris installation program, SPARC, 56
 - installing non-global zones, 24, 42
 - keywords
 - archive identification section, 71–75
 - description, 71
 - `section_begin` and `section_end`, 71
 - user-defined, 75
 - planning
 - creating a differential archive, 30
 - creating an archive, 28
 - installing an archive, 33–34
 - master system, 24–28

archive (*Continued*)

- sections
 - archive cookie, description, 69
 - archive identification, description, 70
 - archives files, description, 70
 - description, 32, 69–71
 - manifest, description, 70
 - summary, description, 70
 - user-defined, description, 70
 - task map, 35–36
 - updating a clone
 - description, 19
- ## C
- clone systems
 - See also* archive
 - description, 15–20
 - creating
 - limitations, 23
 - Solaris Flash archives
 - customizing, 30
 - Solaris Flash archives
 - initial installation, procedure, 41
 - Solaris Flash archives
 - planning, 28
 - requirements for platforms, 27
 - Solaris Flash archives
 - task map, 35–36
 - update, procedure, 48, 51
 - customizing files, example, 46, 47

customizing Solaris Flash archives

- master system, 26
- with scripts, 31

D

differential archive

- See also* archive
- description, 19
- planning, 30

F

files

- customizing, 30
 - excluding, example, 46, 47
 - excluding and including, example, 47
 - including, example, 46
 - large file handling, 30, 44
- `flarcreate` command, 76–80
- Flash, *See* archive

I

installation, Solaris Flash archives, description, 15–20

installing

- limitations, 23
 - Solaris Flash archives
 - how to install, 55–61
 - references to procedures, 57
 - with Solaris installation program, 55
- installing clone systems
- initial installation, 18
 - updating, 19
- installing master systems, 36

K

- keywords, Solaris Flash archives, 71

L

- large file handling, 30, 44
- limitations, 23

M

master system

- See also* archive
 - customizing an installation of, 26
 - description, 24–28
 - peripheral devices, 27–28
- merging a Solaris Flash archive, 59

N

non-global zone

- installing with a Solaris Flash archive, 24, 42

P

- planning, for a Solaris Flash archive installation, 23
- predeployment script, description, 70

R

- recovery image, creating, 63–67
- restoring system, recovery image, 63–67

S

scripts

- Solaris Flash archives
 - creating, 37
 - customizing, 31
- Solaris Flash archives
 - guidelines, 32
- Solaris Flash archives, *See* archive
- Solaris Live Upgrade
 - differential archive creation, example, 52
 - differential archive creation, procedure, 51

Solaris Zones partitioning technology
 installing with a Solaris Flash archive, 24, 42
 splitting a Solaris Flash archive, 58

U

updating a clone system, description, 19

