



Solaris 10 Installation Guide: Solaris Flash Archives (Creation and Installation)

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 817-5668-11
December 2005

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, JumpStart, Solaris Flash, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certaines composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, JumpStart, Solaris Flash, Sun ONE Application Server, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.



051024@13215



Contents

Preface 11

1 Solaris Flash (Overview) 15

- Solaris Flash Introduction 15
 - Installing Clone Systems With an Initial Installation 15
 - Updating Clone Systems With a Solaris Flash Differential Archive 17

2 Solaris Flash (Planning) 21

- Planning Your Solaris Flash Installation 21
 - Designing an Initial Installation of the Master System 21
 - Planning the Creation of a Solaris Flash Archive 24
 - Planning the Installation of Solaris Flash Archives 29

3 Creating Solaris Flash Archives (Tasks) 31

- Task Map: Creating Solaris Flash Archives 31
 - Installing the Master System 32
 - ▼ To Install the Master System for an Initial Installation 33
 - Creating Customization Scripts 33
 - ▼ To Create a Precreation Script 34
 - Using a Precreation Script to Create a User-Defined Archive Section 35
 - ▼ To Create a Predeployment Script 36
 - ▼ To Create a Postdeployment Script 36
 - ▼ To Create a Reboot Script 37
 - Creating a Solaris Flash Archive 38
 - ▼ To Create a Solaris Flash Archive for an Initial Installation 38

Creating a Solaris Flash Archive (Examples)	39
▼ To Create a Solaris Flash Differential Archive With an Updated Master Image	43
▼ To Create a Solaris Flash Differential Archive by Using Solaris Live Upgrade	46

4	Installing and Administering Solaris Flash Archives (Tasks)	51
	Installing a Solaris Flash Archive With the Solaris Installation Program	51
	▼ Installing a Solaris Flash Archive	52
	References to Procedures for Installing Solaris Flash Archives	53
	Administering Solaris Flash Archives	53
	Splitting a Solaris Flash Archive	54
	Merging a Solaris Flash Archive	55
	Extracting Information From an Archive	56
5	Solaris Flash (Reference)	57
	Solaris Flash Archive Section Descriptions	57
	Solaris Flash Keywords	59
	General Keywords	59
	Identification Section Keywords	60
	User-Defined Section Keywords	63
	Solaris Flash <code>fclar create</code> Command	64
	<code>fclar create</code>	64
	Glossary	69
	Index	85

Tables

TABLE 2–1	Flash Archive Sections	28
TABLE 3–1	Task Map: Creating a Solaris Flash Archive to Install With an Initial Installation	31
TABLE 3–2	Task Map: Creating a Solaris Flash Archive to Update a Clone System	32
TABLE 5–1	Flash Archive Sections	57
TABLE 5–2	Values for <code>section_begin</code> and <code>section_end</code> Keywords	59
TABLE 5–3	Identification Section Keywords: General Keywords	60
TABLE 5–4	Identification Section Keywords: Contents of Archive Files Section	60
TABLE 5–5	Identification Section Keywords: User Describes the Archive	61
TABLE 5–6	Identification Section Keywords: Software Describes the Archive	63
TABLE 5–7	Command-Line Options for <code>fclar create</code>	64

Figures

FIGURE 1–1 Solaris Flash Initial Installation 17

FIGURE 1–2 Solaris Flash Update 19

Examples

EXAMPLE 3–1	Excerpts From a Precreation Script	34
EXAMPLE 3–2	Precreation Script	35
EXAMPLE 3–3	Predeployment Script	36
EXAMPLE 3–4	Postdeployment Script	37
EXAMPLE 3–5	Creating a reboot Script	37
EXAMPLE 3–6	Creating an Exact Duplicate Archive	39
EXAMPLE 3–7	Creating an Archive From an Alternate root (/) File System	40
EXAMPLE 3–8	Creating an Archive and Adding Keywords to Describe the Archive	40
EXAMPLE 3–9	Creating an Archive and Excluding and Including Files and Directories	41
EXAMPLE 3–10	Creating an Archive Excluding and Including Files and Directories by Using Lists	41
EXAMPLE 3–11	Creating an Archive Excluding Files and Directories by Using a List and Restoring a Directory	42
EXAMPLE 3–12	Creating an Archive Excluding and Including Files and Directories by Using a List With the -z Option	42
EXAMPLE 3–13	Creating a Differential Archive With the New Master Image on the Master System	45
EXAMPLE 3–14	Creating a Differential Archive With the Images Stored on an Inactive Boot Environment	46
EXAMPLE 3–15	Creating a Differential Archive by Using Solaris Live Upgrade	48
EXAMPLE 4–1	Splitting an Archive	54
EXAMPLE 4–2	Merging a Solaris Flash Archive	56
EXAMPLE 4–3	Merging a Solaris Flash Archive and Adding a User-Defined Section	56
EXAMPLE 4–4	Listing Files in an Archive Section	56

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.

Note – This Solaris release supports systems that use the SPARC® and x86 families of processor architectures: UltraSPARC®, SPARC64, AMD64, Pentium, and Xeon EM64T. The supported systems appear in the *Solaris 10 Hardware Compatibility List* at <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” points out specific 64-bit information about AMD64 or EM64T systems.
- “32-bit x86” points out specific 32-bit information about x86 based systems.

For supported systems, see the *Solaris 10 Hardware Compatibility List*.

Who Should Use This Book

This book is intended for system administrators who are responsible for installing the Solaris operating system. 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 related information that you might need when you install the Solaris software.

TABLE P-1 Related Information

Information	Description
<i>Solaris 10 Installation Guide: Basic Installations</i>	Describes a basic OS installation with a graphical user interface (GUI).
<i>Solaris 10 Installation Guide: Network-Based Installations</i>	Describes how to perform a remote Solaris installation over a local area network or a wide area network.
<i>Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations</i>	Describes how to create the files and directories necessary to perform an unattended custom JumpStart™ installation. This book also describes how to create RAID-1 volumes.
<i>Solaris 10 Installation Guide: Solaris Live Upgrade and Upgrade Planning</i>	Provides planning information when using CD or DVD media to upgrade a system to the Solaris operating system. This book also describes how to use Solaris Live Upgrade to create and upgrade new boot environments.
<i>Solaris 10 Installation Guide: Solaris Flash Archives (Creation and Installation)</i>	Provides instructions for creating Solaris Flash archives and using Solaris Flash archives to install the Solaris OS on multiple systems.
Chapter 24, "Backing Up and Restoring File Systems (Overview)," in <i>System Administration Guide: Devices and File Systems</i>	Describes how to back up system files and other system administration tasks.
<i>Solaris Release Notes</i>	Describes any bugs, known problems, software that is being discontinued, and patches that are related to the Solaris release.
<i>SPARC: Solaris Sun Hardware Platform Guide</i>	Contains information about supported hardware.
<i>x86: Solaris Hardware Compatibility List</i>	Contains supported hardware information and device configuration.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (<http://www.sun.com/documentation/>)
- Support (<http://www.sun.com/support/>)

- Training (<http://www.sun.com/training/>)
-

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 .login file. Use ls -a to list all files. machine_name% you have mail.
AaBbCc123	What you type, contrasted with onscreen computer output	machine_name% su Password:
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is rm filename.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . A <i>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 the C shell, Bourne shell, and Korn shell.

TABLE P-3 Shell Prompts

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

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.

Note – If you want an overview of all Solaris installation methods, see Part I, “Overall Planning of Any Solaris Installation or Upgrade,” in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

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.

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.

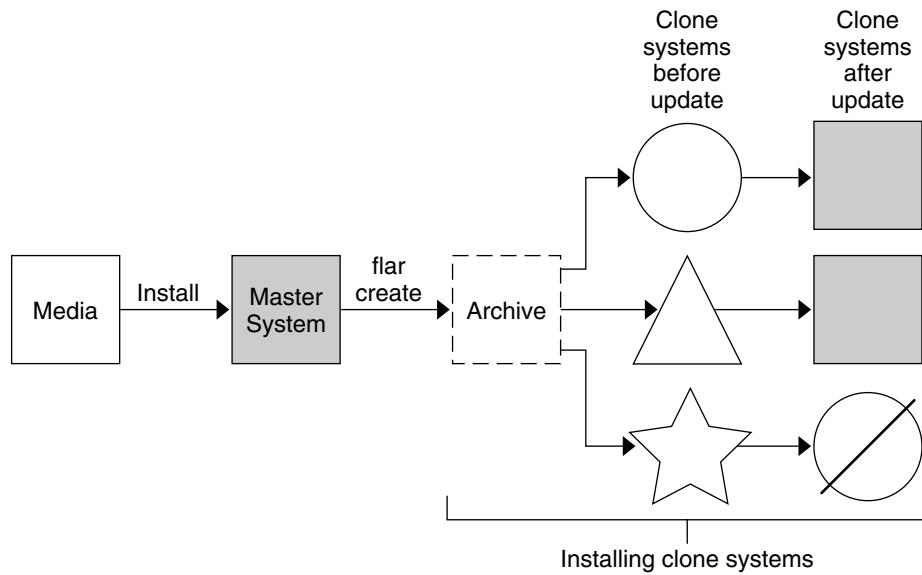
2. (Optional) Prepare customization scripts to reconfigure or customize the clone system before or after installation.
3. Create the Solaris Flash archive. The Solaris Flash archive contains a copy of all of the files on the master system, unless you excluded some nonessential files.
4. Install the Solaris Flash archive on clone systems. The master system and the clone system must have the same kernel architecture.

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 32.

[Figure 1–1](#) shows an installation of clone systems with an initial installation. All files are overwritten.



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

FIGURE 1–1 Solaris Flash Initial Installation

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 `flar create` 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.

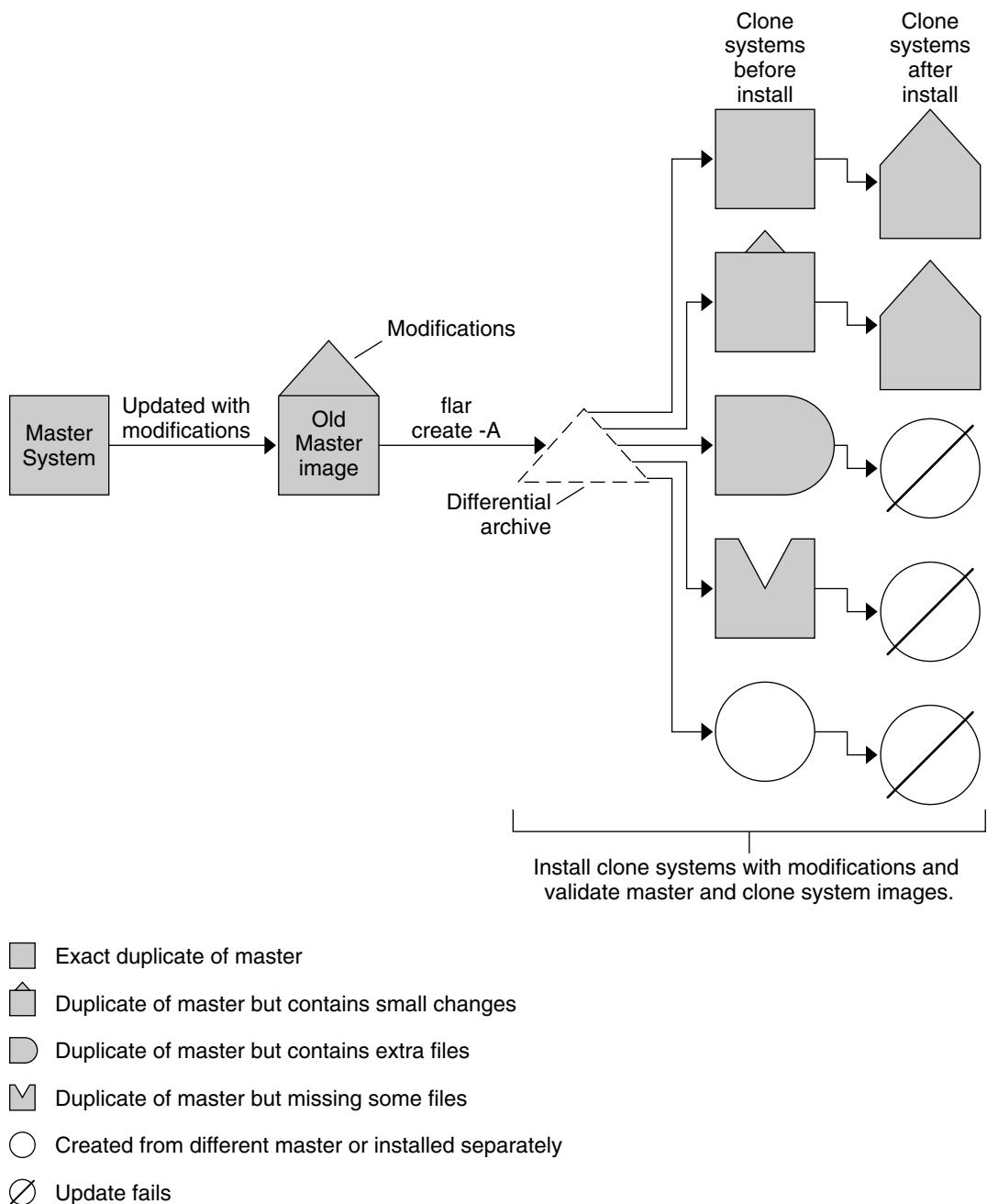


FIGURE 1–2 Solaris Flash Update

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.

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.
- 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 25.

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 26.

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.

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>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>

Type of Installation	Description
Install selected packages	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.

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 CD



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

Starting in the Solaris 9 9/04 release, you can create and install Solaris Flash archives created from a master system that has RAID-1 volumes configured. With the Solaris 9 12/03 and Solaris 9 4/04 release, you must install a patch. To get a the patch that fixes CR 4838219, see sunsolve.sun.com.

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 *Solaris 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 *Solaris 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.

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 to 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 6, “Solaris Live Upgrade (Overview),” in *Solaris 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 43.

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 `flar 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 `flar 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>-z list_filename</code>	<code>-f list_filename</code> <code>-z list_filename</code>

For descriptions of these options, see [Table 5–7](#).

For examples of customizing an archive, see “[Creating a Solaris Flash Archive and Customizing Files \(Examples\)](#)” on page 41.

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 6, “Solaris Live Upgrade (Overview),” in *Solaris 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 5](#).

TABLE 2–1 Flash Archive Sections

Section Name	Informational Only	Description
Archive cookie	X	The first section contains a <code>cookie</code> 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 by using options to the <code>fclar create</code> command.
User-defined		This section follows the 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.
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.

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 51
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>Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations</i>

Installation Program	Archives Storable on This Media	For Step-by-Step Instructions
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>Solaris 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.

- “Task Map: Creating Solaris Flash Archives” on page 31
 - “Installing the Master System” on page 32
 - “Creating Customization Scripts” on page 33
 - “Creating a Solaris Flash Archive” on page 38
-

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 33

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

Task	Description	For Instructions
(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 33
Create the Solaris Flash archive	Use the <code>flar create</code> command to create an archive.	“To Create a Solaris Flash Archive for an Initial Installation” on page 38
(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 38

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 32
(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 33
Create the Solaris Flash differential archive	1. Mount the unchanged master image. 2. Use the <code>flar create</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 43

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 43.

▼ To Install the Master System for an Initial Installation

- Steps**
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 *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.
 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 `flar create` command to create the archive.

- To create custom scripts, see “[Creating Customization Scripts](#)” on page 33.
- To create the archive, see “[Creating a Solaris Flash Archive](#)” on page 38.

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 27.

▼ 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.

- Steps**
1. **Create the precreation script.** Follow the guidelines that are described in “[Guidelines for Creating a Custom Script](#)” on page 27.
 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 `f1check` command. This command cannot be used at the command line. The syntax for this command is as follows:

```
f1check software component files and directories . . . | -
```

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

```
f1check 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:

```
f1check software component files and directories
If Not in selection include by force
f1include 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
else
    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.

- Steps**
1. **Create the predeployment script.** Follow the guidelines that are described in “[Guidelines for Creating a Custom Script](#)” on page 27.
 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.

- Steps**
1. Create the postdeployment script. Follow the guidelines that are described in “[Guidelines for Creating a Custom Script](#)” on page 27.
 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.

- Steps**
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 25.

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.

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

2. **To create the archive, use the `flar create` command.**

`flar create -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 create](#)” on page 64.

- path* The path to the directory in which you want to save the archive file. If you do not specify a path, `flar create` saves the archive file in the current directory.
- filename* The name of the archive file.
- If the archive creation is successful, the `flar create` command returns an exit code of 0.
 - If the archive creation fails, the `flar create` command returns a nonzero exit code.
3. **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/ccc/ddd
/aaa/bbb/fff
/aaa/eee
/ggg
```



Caution – Use the `flar 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.

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`.

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

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

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 3–7 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.

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

EXAMPLE 3–8 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 59.

```
# flar create -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 identification section that contains the detailed description. An example of an 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
  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
```

EXAMPLE 3-8 Creating an Archive and Adding Keywords to Describe the Archive
(Continued)

```
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 3-9 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/ccc remains.

```
# flar create -n archive2 -x /aaa -y /aaa/bbb/ccc 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/ccc
aaa/bbb/ccc/ddd
aaa/bbb
ggg
```

EXAMPLE 3-10 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/ccc
```

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

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

EXAMPLE 3-10 Creating an Archive Excluding and Including Files and Directories by Using Lists *(Continued)*

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 3-11 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.

```
# flar create -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 3-12 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 3-12 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.

```
# flar create -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.

- Steps** 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 33.
- 4. 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 28, “UFS Backup and Restore Commands (Reference),” in *System Administration Guide: Devices and File Systems*.

5. Create the differential archive.

```
# flar create -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.

<code>-A <i>unchanged_master_image_dir</i></code>	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 (/). You can change the default with the <code>-R</code> option. <i>unchanged_master_image_dir</i> is a directory where the unchanged system image is stored or mounted through UFS, NFS, or the <code>lumount</code> command.
<i>options</i>	You can include and exclude some files by using the options for contents selection. For a list of options, see “ flar create ” on page 64.
<i>path</i>	For a description of options, see “ flar create ” on page 64.
<i>filename</i>	Specifies the path to the directory in which you want to save the archive file. If you do not specify a path, <code>flar create</code> saves the archive file in the current directory.

Specifies the name of the archive file.

- If the differential archive creation is successful, the `flar create` command returns an exit code of 0.
- If the differential archive creation fails, the `flar create` 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 *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

Example 3–13 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.

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

Example 3–14 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  
# flar create -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 6, “Solaris Live Upgrade (Overview),” in *Solaris 10 Installation Guide: Solaris Live Upgrade and Upgrade Planning*.

Steps 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. Check the status of the two boot environments.

```
# lustatus copy_BE
```

boot environment	Is Complete	Active Now	Active OnReboot	Can Delete	Copy Status

master_BE	yes	yes	yes	no	-
copy_BE	yes	no	no	yes	-

3. 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.

4. (Optional) Create custom scripts. See “[Creating Customization Scripts](#)” on page 33.

5. Create the differential archive.

a. Mount the newly created boot environment.

```
# luumount BE_name /a
```

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

```
# flar create -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 create](#)” on page 64.

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

filename Specifies the name of the archive file.

c. Unmount the new boot environment.

```
# luumount copy_BE
```

The *flar create* 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.

6. 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
```

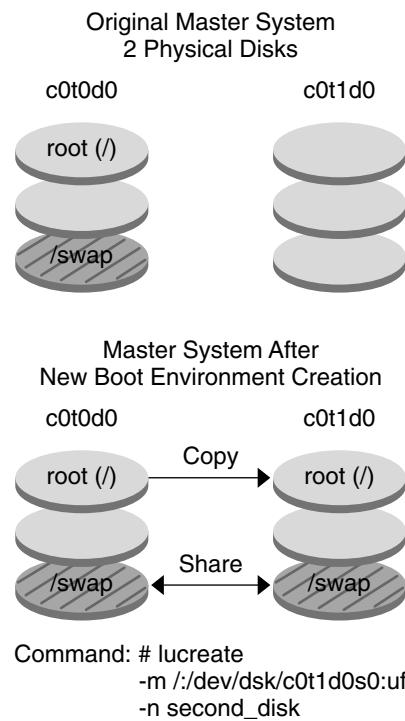
Example 3–15 Creating a Differential Archive by Using Solaris Live Upgrade

`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 `flar create` 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
# flar create -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 *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

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



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.



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 51.
 - 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 53.
 - To split or merge an archive, see “[Administering Solaris Flash Archives](#)” on page 53.
-

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

- Steps** 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" in *Solaris 10 Installation Guide: Basic Installations*
- x86: "Performing an Installation or Upgrade With the Solaris Installation Program" in *Solaris 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 to complete the installation.

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.

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 51■ Solaris Live Upgrade – See “Installing Solaris Flash Archives on a Boot Environment” in <i>Solaris 10 Installation Guide: Solaris Live Upgrade and Upgrade Planning</i>.■ Custom JumpStart installation program – See “Creating a Profile” in <i>Solaris 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>Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations</i>.■ WAN boot installation method – See Chapter 11, “WAN Boot (Overview),” in <i>Solaris 10 Installation Guide: Network-Based Installations</i>.
An update with a Solaris Flash differential archive	<ul style="list-style-type: none">■ Custom JumpStart installation program – See “Creating a Profile” in <i>Solaris 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>Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations</i>.■ Solaris Live Upgrade – See “To Install a Solaris Flash Archive With a Profile (Command-Line Interface)” in <i>Solaris 10 Installation Guide: Solaris Live Upgrade and Upgrade Planning</i>.

Administering Solaris Flash Archives

The `f1ar` 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
```

<code>-d dir</code>	Retrieves the sections to copy from <i>dir</i> , 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 <i>archive</i> , rather than placing it in a file with the name <i>archive</i> .
<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.

EXAMPLE 4-1 Splitting an Archive *(Continued)*

- **identification** – A copy of the Archive Identification section with all keyword-value pairs.
- **archive** – The cpio 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 a Solaris Flash Archive

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. Each section is assumed to be in a separate file, the names of which are the section names. At a minimum, these three files must be present:

- Archive Cookie (**cookie**)
- Archive Identification (**identification**)
- Archive Files (**archive**)

When combining sections, remember the following points:

- If **archive** is a directory, **flar** uses **cpio** to archive the directory before including it in the combined archive.
- If the Archive Identification section specifies to compress the archive, **flar** compresses the contents of the newly combined archive.
- No validation is performed on any of the sections. In particular, no fields in the Archive Identification section are validated or updated.

```
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 get 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
```

Solaris Flash (Reference)

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

- “Solaris Flash Archive Section Descriptions” on page 57
 - “Solaris Flash Keywords” on page 59
 - “Solaris Flash `flar create` Command” on page 64
-

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 5–1 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 5–1 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 cpio■ 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 “Identification Section Keywords” on page 60.</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>f1ar create -M</code> option. Because no validation of the archive occurs, excluding this section is not recommended.</p>	No	No
Predeployment, Postdeployment, Reboot	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.	Yes	No
Summary	This section contains messages about the archive creation and records the activities of predeployment scripts.	Yes	Content is generated by both user and the software
User-defined	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.	No	Yes

TABLE 5–1 Flash Archive Sections (Continued)

Section Name	Description	Required by Archive?	Requires Input From User?
Archive files	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.	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 5–2 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> .

TABLE 5–2 Values for section_begin and section_end Keywords (Continued)

Archive Section	Value for section_begin and section_end keywords
Archive files	archive

Identification Section Keywords

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 5–3 to delimit each section.

TABLE 5–3 Identification Section Keywords: General Keywords

Keywords	Value Definitions	Value	Required
section_begin	These keywords are used to delimit sections in the archive and are not limited exclusively to the identification section. For a description of these keywords, see “General Keywords” on page 59.	Text	Yes
section_end			

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

TABLE 5–4 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 F1AsH-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 the keyword is present, it has the value of cpio.■ 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 the files_compressed_method is present, the compression method is applied to the archive file that is created by the archive method.	Text	No

TABLE 5–4 Identification Section Keywords: Contents of Archive Files Section (Continued)

Keywords	Value Definitions	Value	Required
<code>files_archived_size</code>	This keyword value is the size of the archived files section in bytes.	Numeric	No
<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 `flar create` command to include these keywords. For an example, see [Example 3–8](#).

TABLE 5–5 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>flar create</code> command with the <code>-i</code> option to create the date. ■ If you do not specify a creation date with the <code>flar create</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

TABLE 5–5 Identification Section Keywords: User Describes the Archive *(Continued)*

Keywords	Value Definitions	Value	Required
creation_master	This keyword value is the name of the master system you used to create the archive. You can use the <code>flar create -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
content_name	This keyword identifies the archive. The value is generated from the <code>flar 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>flar 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>flar 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>flar 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 5–6 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 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 identification section, see [Example 3–8](#).

Solaris Flash `flar create` Command

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

`flar create`

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

The master system should be in as stable a state as possible when you create a Solaris Flash archive. The syntax of the command is as follows:

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

In this command line, *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 5-7 Command-Line Options for `flar create`

Option	Description
Required Options	
<code>-n archive_name</code>	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	
<code>-c</code>	Compresses the archive by using <code>compress (1)</code> .
Options for Directories and Sizes	
<code>-R root</code>	Creates the archive from the file system tree that is rooted at <i>root</i> . If you do not specify this option, <code>flar create</code> creates an archive from a file system that is rooted at <code>/</code> .

TABLE 5-7 Command-Line Options for `flar create` (Continued)

Option	Description
<code>-S</code>	Omits sizing information in the archive.
<code>-H</code>	Does not generate the hash identifier.
Options for Creating a Differential Archive	
<code>-A</code>	<p>Creates a differential archive by comparing a new system image <i>unchanged_master_image_dir</i> with the image that is specified by the <i>unchanged_master_image_dir</i> argument. By default, the new system image is root (/). 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 lumount.</p> <p>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.</p>
<code>-M</code>	<p>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.</p>
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.	
<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>

TABLE 5–7 Command-Line Options for `fclar create` (Continued)

Option	Description
<code>-f list_filename</code>	Adds files and directories from a list to the archive. <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. <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>"-"</code>, <code>fclar create</code> reads standard input as the list of files. When you use the value <code>"-"</code>, the archive size is not calculated.
<code>-F</code>	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.
<code>-x exclude_dir/filename</code>	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. <i>exclude_dir/filename</i> is the name of the directory or file to be excluded.
<code>-X list_filename</code>	Excludes a list of files or directories from the archive. <i>list_filename</i> is the full path to a file that contains the list. <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>"-"</code>, <code>fclar create</code> reads standard input as the list of files. When you use the value <code>"-"</code>, the archive size is not calculated.
<code>-z list_filename</code>	Excludes or includes a list of files or directories from the archive. Each file or directory in the list is noted with a plus <code>"+"</code> or minus <code>"-"</code> . A plus indicates an included file or directory and the minus indicates an excluded file or directory. <i>list_filename</i> is the full path to a file that contains the list. <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 5-7 Command-Line Options for `flar create` (Continued)

Option	Description
<code>-I</code>	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 <code>-I</code> option.
Options Used With User-Defined Sections	
<code>-u section</code>	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.
<code>-d dir</code>	Retrieves the section file that is specified with <code>-u</code> from <i>dir</i> .
Options Used With Tape Archives	
<code>-t</code>	Creates an archive on a tape device. The <i>filename</i> argument is the name of the tape device.
<code>-p posn</code>	Use only with the <code>-t</code> 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.
<code>-b blocksize</code>	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 identification section of the archive.	
<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.

TABLE 5–7 Command-Line Options for `flar create` (Continued)

Option	Description
<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 type, <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	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. 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.
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	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 (/) file system is mounted. Two boot archives are maintained on a system:

	<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 (/) 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.
bootlog-cgi	The CGI program that enables a web server to collect and store remote client-booting and installation console messages during a WAN boot installation.
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.
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	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 installed 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 <i>vfstab</i> of the active and inactive boot environments. Example file systems are <i>root</i> (/), <i>/usr</i> , <i>/var</i> , and <i>/opt</i> . 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 <i>rules</i> 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 <i>\$1</i> 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> .
decryption	The process of converting coded data to plain text. See also encryption .
derived profile	A profile that is dynamically created by a begin 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 disc .
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 pfinstall 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 10 release.
Entire Solaris Software Group Plus OEM Support	A software group that contains the entire Solaris 10 release, plus additional hardware support for OEMs. This software group is recommended when installing Solaris software on SPARC based servers.
/etc	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	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 brings the system up 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 fdisk partition on an x86 based system. x86 based systems allow up to four different fdisk partitions on a disk. These partitions can be used to hold individual operating systems. Each operating system must be located on a unique fdisk partition. A system can only have one Solaris fdisk 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 rules 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 Solaris Zones and non-global zone.
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 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 fdisk partition settings.
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.
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	An installation that overwrites the currently running software or initializes a blank disk.
	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.
install server	A server that provides the Solaris DVD or CD images from which other systems on a network can install Solaris (also known as 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.
IP address	(Internet protocol address) In TCP/IP, a unique 32-bit number that identifies each host in a network. An IP address consists of four numbers that are separated by periods (192.168.0.0, for example). Most often, each part of the IP address is a number between 0 and 225. However, the first number must be less than 224 and the last number cannot be 0.

IP addresses are logically divided into two parts: the network (similar to a telephone area code), and the local system on the network (similar to a phone number). The numbers in a Class A IP address, for example, represent “network.local.local.local” and the numbers in a Class C IP address represent “network.network.network.local.”

Class	Range (xxx is a number 0 to 255)	Number of Available IP Addresses
Class A	1.xxx.xxx.xxx - 126.xxx.xxx.xxx	Over 16 million
Class B	128.0.xxx.xxx - 191.255.xxx.xxx	Over 65,000
Class C	192.0.0.xxx - 223.255.255.xxx	256

IPv6

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.

IPv6 is described in more detail in Part I, “Introducing System Administration: IP Services,” in *System Administration Guide: IP Services*.

job

A user-defined task to be completed by a computer system.

JumpStart directory

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.

JumpStart installation

A type of installation in which the Solaris software is automatically installed on a system by using the factory-installed JumpStart software.

Kerberos

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.

key

The code for encrypting or decrypting data. See also [encryption](#).

keystore

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.

LAN

(local area network) A group of computer systems in close proximity that can communicate by way of some connecting hardware and software.

LDAP	(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.
locale	A geographic or political region or community that shares the same language, customs, or cultural conventions (English for the U.S. is en_US, and English for the U.K. is en_UK).
logical device	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 in the view of an application or file system.
manifest section	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.
master system	A system that you use to create a Solaris Flash archive. The system configuration is saved in the archive.
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.1st 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 fdisk partition settings.
metadevice	See <i>volume</i> .
miniroot	The smallest possible bootable Solaris root (/) file system. A miniroot contains a kernel and just enough software to install the Solaris environment on a hard disk. The miniroot is the file system that is copied to a machine in the initial installation.
mirror	See RAID-1 volume.
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, /usr).
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 name service to systems on a network.

name 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 name service, the system information can be maintained, managed, and accessed on a network-wide basis. Without a name service, each system has to maintain its own copy of the system information in the local /etc files. Sun supports the following name services: LDAP, NIS, and NIS+.
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.
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> .
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 Solaris Zones and global zone.
nonnetworked systems	Systems that are not connected to a network or do not rely on other systems.
/opt	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 (/) file system and swap space (/export/root, /export/swap).
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 i86pc and sun4u.
platform name	The output of the <code>uname -i</code> command. For example, the platform name for the Ultra 60 is SUNW,Ultra-60.
Power Management	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—a sun4u SPARC system, for example—the Power Management software is installed by default. After a subsequent reboot, you are prompted to enable or disable the Power Management software.
	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.
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> .
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.
private key	The decryption key used in public-key encryption.
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-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 submirrors. A RAID-1 volume is sometimes called a mirror.
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.
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.
root	The top level of a hierarchy of items. Root is the one item from which all other items are descended. See root directory or root (/) file system.
root directory	The top-level directory from which all other directories stem.
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 dismounted. 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.
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 rules 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 rules file is used in a custom JumpStart installation. See also <i>profile</i> .
rules.ok file	A generated version of the rules file. The rules.ok file is required by the custom JumpStart installation software to match a system to a profile. You <i>must</i> use the check script to create the rules.ok 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> 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.
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, known as the master system. 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 the 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 disk 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 state database replica. 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 RAID-0 volume.
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 loaded back into memory. Also called the <code>/swap</code> or <code>swap</code> file system.
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	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 saves 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 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	A file system on a standalone system or server that contains many of the standard UNIX programs. Sharing the large /usr 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	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, vi files, mail files, and uucp files.
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 in the view of an application or file system.
	In some command-line utilities, a volume is called a metadevice. 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 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 <code>wanboot</code> binary performs tasks similar to the <code>ufsboot</code> or <code>inetboot</code> second-level boot programs.
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.
zone	See non-global zone

Index

A

administering, Solaris Flash archives, 53
archive
See also scripts
administering, 53
compressing, 29
creating an archive, 38
 examples, 39
 requirements for platforms, 22
customizing
 description, 26
 with scripts, 27
description of process, 15-19
`fclar create` command, 64-68
getting information, 56
installing
 description, 15-19
 how to install, 51-56
 installation programs, 29-30
 Solaris installation program, SPARC, 52
installing non-global zones, 38
keywords
 description, 59
 identification section, 60-63
 section_begin and
 section_end, 59-60
 user-defined, 63
planning
 creating a differential archive, 25
 creating an archive, 24
 installing an archive, 29-30
master system, 21-24

archive (Continued)

sections
 archive cookie, description, 57
 archive identification, description, 58
 archives files, description, 59
 description, 28, 57-59
 manifest, description, 58
 summary, description, 58
 user-defined, description, 58
task map, 31-32
updating a clone
 description, 17

C

clone systems
See also archive
 description, 15-19
creating
 Solaris Flash archives
 customizing, 26
 initial installation, procedure, 38
 planning, 24
 requirements for platforms, 22
 task map, 31-32
 update, procedure, 43, 46
 customizing Solaris Flash archives
 master system, 22
 with scripts, 27

D

differential archive

See also archive

description, 17

planning, 25

P

planning, for a Solaris Flash archive

installation, 21

predeployment script, description, 58

F

`flar create` command, 64-68

Flash, *See* archive

I

installation, Solaris Flash archives,

description, 15-19

installing

Solaris Flash archives

how to install, 51-56

references to procedures, 53

with Solaris installation program, 51

installing clone systems

initial installation, 15

updating, 17

installing master systems, 32

S

scripts

Solaris Flash archives

creating, 33

customizing, 27

guidelines, 27

Solaris Flash archives, *See* archive

Solaris Live Upgrade

differential archive creation, example, 48

differential archive creation, procedure, 46

Solaris Zones partitioning technology, installing

with a Solaris Flash archive, 38

splitting a Solaris Flash archive, 54

K

keywords, Solaris Flash archives, 59

M

master system

See also archive

customizing an installation of, 22

description, 21-24

peripheral devices, 22-24

merging a Solaris Flash archive, 55

N

non-global zone, installing with a Solaris Flash

archive, 38