Building a Bootable DVD to Deploy a Solaris™ Flash Archive

John S. Howard, Reference Architecture Engineering

Sun BluePrints™ OnLine—April 2004

http://www.sun.com/blueprints

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
650 960-1300

Part No. 817-6991-10
Revision 1.0, 4/15/04
Edition: April 2004

Please
Recycle

Building a Bootable DVD to Deploy a Solaris™ Flash Archive

In some situations, you might need to perform an automated (hands-free) installation of a Solaris™ Flash archive (flar), but are unable to use a JumpStart™ server. For example, deploying software stacks in environments with any of the following constraints can inhibit the use of a network-based JumpStart server:

- Disk space limitations
- Many subnets that cannot have their own JumpStart server or other networking limitations
- Remote locations with low-bandwidth network connections

In these situations, you can create a bootable installation DVD-ROM that essentially places a JumpStart server and complete software stack (the flar) on a DVD. You can then use this DVD to effect a standardized and fully automated installation of the software stack.

Using a DVD in this manner enables much more space to be used for the Solaris Flash archive than would be available with a CD-ROM. To execute the JumpStart framework and component installation programs, the DVD requires a Solaris miniroot that occupies approximately .5 gigabytes in slice 1 of the DVD. Standard DVD media has a capacity of approximately 4.7 gigabytes, therefore, after placing the miniroot and making the DVD bootable, approximately 4 gigabytes will be available for the flash archive (the *payload*) in slice 0 of the DVD.

This article assumes that you are familiar with the concepts of a “classic” JumpStart installation and Solaris Flash. It describes the boot and installation processes and explains how to adapt them to build bootable DVDs for custom system installations.

Note – The procedure described in this article is an extension of the procedure for building a bootable CD-ROM for the JumpStart installation described in the Sun BluePrints™ book *JumpStart™ Technology: Effective Use in the Solaris™ Operating Environment*, by John S. Howard and Alex Noordergraaf, ISBN 0-13-062154-4.

Building a Bootable Installation DVD-ROM

This section examines the structure of a bootable Solaris™ 9 Operating System (Solaris OS) (for SPARC®) DVD and discusses the appropriate modifications to the default installation scripts that allow you to perform a JumpStart installation from DVD. Further, this section describes how to create a bootable Solaris 9 OS installation DVD for the SPARC platform. Additionally, the Solaris 9 CD Read / Write (CDRW) utilities are used to write the Solaris 9 OS bootable installation DVD. Although the CDRW utilities are named "CD," they are applicable to both CDs and DVDs. While several approaches and software applications are available for writing DVDs, this article uses commands available only in the standard Solaris 9 OS to create and write the bootable installation DVD.

The structure of the bootable installation DVD might vary with different versions of the Solaris OS, partly because of changes required to support additional hardware architectures. Additionally, changes to the Solaris OS from version to version might necessitate changes in the structure or layout of the DVD.

While there might be structural variances across versions or updates of the Solaris OS, the concepts and procedures presented here can be adapted or extended to create a bootable installation DVD for any of the current versions of the Solaris OS.

Note – The techniques presented in this section can be used to produce configurations that are not supported by Sun Enterprise™ Services. However, this should not detract from their value.

Bootable DVD Structure

A bootable Solaris OS DVD has several components that are common to any other hard disk. The boot DVD is divided into several partitions (or slices) and a volume table of contents (VTOC) is used to provide the location and sizes of these slices. In addition to the VTOC, a typical installation DVD has six slices. Although the Solaris OS imposes the partitioning of the DVD into six slices, it is very important to note that the DVD is written as one *session*.

The following section describes the VTOC and the six slices of the Solaris 9 OS installation DVD.

Volume Table of Contents

The VTOC is located at cylinder 0, sector 0 on the DVD. You can examine the VTOC of any disk device with the `prtvtoc` command. The VTOC of the Solaris 9 OS Software DVD (the installation DVD) appears as follows.

```
lokoya# /etc/init.d/volmgt stop
lokoya# prtvtoc /dev/dsk/c0t6d0s0
* /dev/dsk/c0t6d0s0 partition map
*
* Dimensions:
*     512 bytes/sector
*     640 sectors/track
*       1 tracks/cylinder
*     640 sectors/cylinder
*    2048 cylinders
*    2048 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector    Count   Sector  Mount Directory
       0      4    10          0   6026240   6026239
       1      2    10    6026240   1050880   7077119
       2      0    00    7077120      5120   7082239
       3      0    00    7082240      5120   7087359
       4      0    00    7087360      5120   7092479
       5      0    00    7092480      5120   7097599
lokoya# /etc/init.d/volmgt start
```

Note – The DVD and floppy volume management must be stopped in order to execute the `prtvtoc` command on a DVD. Volume management is then restarted after executing `prtvtoc`. All file systems mounted from the DVD will be unmounted and become inaccessible while volume management is stopped.

In contrast to a hard disk, the disk geometry that the Solaris OS uses for a DVD provides no distinction between a cylinder and a track. As the `prtvtoc` output illustrates, the disk label used for a DVD defines a cylinder as being composed of one track. Further, the `prtvtoc` output verifies that each track is defined as having 640 sectors and one sector is equal to 512 bytes.

It is important to note that the Solaris OS requires that all UFS file systems align on a cylinder boundary. For a DVD, this means that all UFS file systems on the DVD must begin on a sector that is a multiple of 640.

Slices

By reading the VTOC, the Solaris OS sees the DVD as having six slices. These six slices are described as follows:

- Slice 0 contains the Solaris OS packages to be installed and is the on High Sierra File System (HSFS) partition of the DVD.
- Slice 1 contains the generic kernel and the file system the install client will mount as `/` (root) after boot.
- Slice 2 contains the boot block for the `sun4c` architecture.
- Slice 3 contains the boot block for the `sun4m` architecture.
- Slice 4 contains the boot block for the `sun4d` architecture.
- Slice 5 contains the boot block for the `sun4u` architecture.

Slices 2 through 5 exist only to provide hardware architecture-specific boot blocks. As new hardware architectures are added and old architectures reach their end-of-lives, the uses of these slices might change. The file `.slicemapfile` in the top-level directory of slice 0 contains the mapping of a slice to the architecture supported.

As noted earlier, slice 0 is on the HSFS partition, and all other slices are on the UFS partitions. Slice 0 is also the largest of the slices and can incorporate any unused space on the DVD. The procedures detailed in this section augment the installation procedures in slice 0. However, there is a fixed upper limit in available space for slice 0 that limits our modifications. (The total space available on a standard DVD is 4.7 gigabytes.)

The distribution media for Solaris 9 OS supports four architectures. If the bootable installation DVD being created only needs to support one architecture, the space (slices) used by the unneeded architectures can be incorporated into slice 0, enlarging slice 0 at the expense of losing the ability to boot other architectures from that DVD.

It is also interesting to note that, other than the boot block, the only contents of slices 2 through 5 is the file `.SUNW-boot-redirect` in the top-level directory of each of those partitions. This file contains the character 1, which redirects the OpenBoot™ PROM (OBP) boot loader to load the kernel from partition 1. This mechanism was added with Solaris 2.5 Operating Environment as a means of taking advantage of the hardware-independent nature of the kernel to optimize the utilization of space on the DVD.

Procedure Overview

Generally, this procedure extracts the contents of slice 0, then splices the desired installation behaviors into the contents of slice 0. The modifications made to slice 0 are to configure the bootable installation DVD to partition `c0t0d0` as the boot device. The modifications then enable a fully automated installation of the Solaris 9 OS. The profile specifies that a Solaris Flash archive will be installed.

The following steps are the high-level tasks involved in creating a bootable DVD. The specific steps are described in the following section.

1. Create and populate a work area, including copying in the flar.
2. Modify the installation behaviors of slice 0.
3. Assemble the individual slices into one DVD session and write them to the bootable installation DVD.
4. Test the bootable installation DVD.

You can also use this procedure to create a bootable DVD without the JumpStart installation behaviors, by omitting step 2.

Procedure Specifics

For this example, `lokoya` is a Sun™ Fire 6800 server running the Solaris 9 12/03 OS with the Solaris 9 12/03 OS CD/DVD creation utilities installed and configured as a JumpStart server. `lokoya` has a DVD-ROM writer connected at `c3t2d0` (identified as `cdrom0` by the `cdrw -l` command).

1. Create and populate the work area, including copying in the flar.

Verify the presence of the Solaris OS CD/DVD creation utilities.The Solaris 9 12/03 OS installation media is already mounted, and you can use `/bidvd9` as the work area. `/bidvd9` is a 12-gigabyte UFS file system. In our example, we created `/bidvd9` as follows.

```
lokoya# pkginfo SUNWmkcd SUNWcdrw
system      SUNWcdrw       CD read and write utility for Solaris
system      SUNWmkcd       CD creation utilities
lokoya# newfs -m 1 /dev/rdsk/c0t1d0s0
newfs: construct a new file system /dev/rdsk/c0t1d0s0: (y/n)? y
[ ... additional output omitted for brevity ... ]
lokoya# mkdir /bidvd9
lokoya# mount /dev/dsk/c0t1d0s0 /bidvd9
```

a. Populate the work area by extracting the partitions from the Solaris 9 OS software DVD.

Because the contents of slice 0 will be manipulated, use the `cpio` command to copy out partition 0. Because no changes are made to the contents of slices 1 through 5, use `dd` to take those slices off the DVD. Before extracting slices 1 through 5, DVD and floppy volume management is stopped.

```
lokoya# cd /cdrom/sol_9_1203_sparc
lokoya# find s0 -print |cpio -pudm /bidvd9/s0
lokoya# cd /bidvd9
lokoya# /etc/init.d/volmgt stop
lokoya# for i in 1 2 3 4 5
> do
> dd if=/dev/dsk/c0t6d0s${i} of=s9u5.s${i} bs=512
> done
1050880+0 records in
1050880+0 records out
5120+0 records in
5120+0 records out
5120+0 records in
5120+0 records out
5120+0 records in
5120+0 records out
5120+0 records in
5120+0 records out
```

Note – All file systems mounted from the DVD are unmounted while volume management is stopped.

---

b. Use the `dd` command to take the VTOC from the DVD and restart volume management as follows.

Because the slice layout of the bootable installation DVD being created will not vary from the slice layout of the Solaris 9 OS DVD, the VTOC from the DVD can be used later for the bootable installation DVD.

```
lokoya# dd if=/dev/dsk/c0t6d0s0 of=s9u5.dvd.vtoc bs=512 count=1
1+0 records in
1+0 records out
lokoya# /etc/init.d/volmgt start
```

c. Remove the unneeded files from slice 0.

To make room on slice 0 for the Solaris Flash archive, you need to evacuate (or remove) the unneeded files from slice 0. The files you decide to remove will vary depending on your site's specific requirements. In general, because the flar will contain the complete software load to be installed, the package-based installation files in slice 0 are redundant. Specifically, the following files or directories are typically not needed when using a Solaris Flash archive:

- Solaris packages in `s0/Solaris_9/Product`
- Solaris supplemental software in `s0/SW_Supp_CD`
- Solaris documentation in `s0/1of2_Doc_CD` and `s0/2of2_Doc_CD`

The following sample shows how we removed the unneeded directories and how we copied the flar into the slice 0 workspace for our example. This step assumes that the Flash archive has already been created using `flarcreate` and was transferred to `lokoya`, the JumpStart server.

```
lokoya# cd /bidvd9/s0/Solaris_9/Product
lokoya# rm -rf *
lokoya# cd /bidvd9/s0
lokoya# rm -rf 1of2_Doc_CD 1of2_Doc_CD SW_Supp_CD
lokoya# cp /jumpstart/FlashArchives/s9.archive \
> /bidvd9/s0/Solaris_9/Product
```

For information about creating and using Solaris Flash archives, refer to the *Solaris 9 Installation Guide* available at `http://docs.sun.com` or the Sun BluePrints book *JumpStart™ Technology: Effective Use in the Solaris™ Operating Environment*, by John S. Howard and Alex Noordergraaf, ISBN 0-13-062154-4.

2. **Modify the installation behavior of slice 0 by deleting the contents of the `.install_config` directory and adding the desired JumpStart rules and profile to this directory.**

   It is important to note the `archive_location` profile keyword (shown in bold in the following example). This keyword instructs the JumpStart framework that the archive is to be found on a device local to the install client (`local_file`) and the location of the archive (`/cdrom/Solaris_9/Product/s9.archive`).

   Also, note that the parsed `rules.ok` file (the output from the `check` script), not the `rules` file, must be placed in the `.install_config` directory. If any begin or finish scripts are being used, place them in the `.install_config` directory as well.

```
lokoya# cd /jumpstart
lokoya# rm /bidvd9/s0/.install_config/*
lokoya# cat bidvd9.profile
install_type flash_install
archive_location local_file /cdrom/Solaris_9/Product/s9.archive
partitioning    explicit
#
# 2.0GB swap on a 36GB disk
# 4 cylinders on slice 7 for SVM's MetaData
#
filesys         rootdisk.s0     free            /
filesys         rootdisk.s1     1:1450           swap
filesys         rootdisk.s7     1451:4           unnamed
lokoya# cp /jumpstart/Profiles/bidvd9.profile \
> /bidvd9/s0/.install_config
lokoya# cat rules
any - - bidvd9.profile -
lokoya# ./check
Validating rules...
Validating profile bidvd9.profile...
The custom JumpStart configuration is ok.
lokoya# cp rules.ok /bidvd9/s0/.install_config
```

   The setup of the installation profile directory is controlled by the `profind` script. Modify this script to redirect the configuration directory environment variable (`${SI_CONFIG_DIR}`) used by the JumpStart software to the `.install_config`

directory on the bootable installation DVD. To make this change, edit the `/bidvd9/s0/Solaris_9/Tools/Boot/usr/sbin/install.d/profind` shell script and replace the `cdrom()` function with the following code.

```
cdrom()
{
   # Factory JumpStart is only allowed with factory
   # stub images, indicated by the file /tmp/.preinstall
   #
   if [ -f /tmp/.preinstall ]; then
      mount -o ro -F lofs ${CD_CONFIG_DIR} ${SI_CONFIG_DIR} >/dev/null 2>&1
      if [ $? -eq 0 ]; then
         verify_config "defaults" "CDROM"
      fi
   fi
   gettext " <<< using DVD install_config >>>"; echo   # bidvd9
   rmdir ${SI_CONFIG_DIR}                              # bidvd9
   ln -s /cdrom/.install_config ${SI_CONFIG_DIR}       # bidvd9
}
```

This modification instructs the installation process to use the `.install_config` directory that was populated with the desired JumpStart profiles and `rules` file.

3. **Assemble and write individual slices to the DVD.**

   At this point, write the VTOC, the modified slice 0, and the unmodified slices 1 through 5 to the bootable installation DVD being created.

   a. **Convert the modified slice 0.**

      Before combining the slices and writing them to the DVD, use the `mkisofs` command to convert the modified slice 0 in the `/bidvd9/s0` work area into an HSFS (iso9660) file system. Keep in mind that the iso9660 file system has some overhead, which increases the image (created by `mkisofs`).

   b. **Combine the individual slices into one image to be written to a blank DVD.**

   c. **Write the VTOC, the modified slice 0, and the unmodified, combined slices 1 through 5 to the bootable installation DVD.**

      Because no changes to the miniroot or supported architectures are required, slices 1 through 5 are extracted from the Solaris 9 OS software DVD and are written, unchanged, to the bootable installation DVD.

d. It is important to note that `mkisofs` creates a VTOC at offset 0 within this image. Use the `dd` command to remove this invalid VTOC from the HSFS image by skipping the first 512 byte block.

In our example, we removed the unneeded power management packages from the `Product` directory before creating the iso9660 file system from `/bidvd9/s0`.

```
lokoya# cd /bidvd9
lokoya# mkisofs -R -D -d -L -l -o s9u5.S0 s0
Warning: creating filesystem that does not conform to ISO-9660.
[ ... additional output omitted for brevity ... ]
 99.36% done, estimate finish Tue Apr 6 15:02:24 2004
 99.71% done, estimate finish Tue Apr 6 15:02:26 2004
Total translation table size: 0
Total rockridge attributes bytes: 7882055
Total directory bytes: 22704128
Path table size(bytes): 109482
Max brk space used 3726000
1409040 extents written (2752 Mb)
lokoya# dd if=/bidvd9/s9u5.S0 of=/bidvd9/s9u5.s0 bs=512 skip=1
5636159+0 records in
5636159+0 records out
lokoya# rm /bidvd9/s9u5.S0
```

e. Pad the size for slice 0 to match the size specified in the VTOC.

Because the VTOC specifies a size for slice 0, you must pad it to maintain the validity of the VTOC and to maintain the correct cylinder boundaries. The size of the pad is computed by adding one to the number of sectors in the HSFS slice 0 image (this accounts for the VTOC), and then subtracting that sum from the number of sectors (reported by `prtvtoc` earlier) in the unmodified slice 0 on the DVD. In our example, we created the pad using the `dd` command to read the appropriate number of zeros from `/dev/zero` as follows.

```
lokoya# bc
6026240-(5636159+1)
390080
lokoya# dd if=/dev/zero of=pad.s0 bs=512 count=390080
390080+0 records in
390080+0 records out
```

f. Provide neecessary installation client identification information.

As with any automated installation, `sysidtool` needs all installation client identification information such as hostname, IP address, and timezone. The location of this information varies depending on whether the installation client is connected to a network or off-network during the installation.

If the installation client is connected to a network during installation, this information must be available from a name service such as NIS+ or NIS, or provided from the `/etc/bootparams`, `/etc/ethers`, and `sysidcfg` files from a host on the network. The minimum entries required in the `/etc/bootparams` file are shown.

```
lokoya# cat /etc/bootparams
barossa sysid_config=lokoya:/jumpstart/Sysidcfg/Solaris_9
```

The `sysidcfg` file specified by `/etc/bootparams` appears as follows.

```
lokoya# cat /jumpstart/Sysidcfg/Solaris_9/sysidcfg
system_locale=en_US
timezone=US/Pacific
network_interface=primary {netmask=255.255.255.0
                           default_route=none
                           protocol_ipv6=no}
terminal=vt100
security_policy=NONE
root_password=Q7jsh1m6IztTU
name_service=NONE
timeserver=localhost
```

g. Automate the installation.

To perform an automated installation without network connectivity, a `sysidcfg` file must be placed in the `etc` directory of the file system image taken from slice 1 of the Solaris 9 OS DVD. You can mount the file system image file using the Solaris 9 OS loopback file driver administration commands. After the file system image has been mounted, use standard Solaris OS commands to remove the symbolic link for the default `sysidcfg` file and a complete `sysidcfg` file can be copied to the file system image.

```
lokoya# cat /bidvd9/sysidcfg
system_locale=en_US
timezone=US/Pacific
network_interface=primary {hostname=barossa
                           ip_address=10.1.1.9
                           netmask=255.255.255.0
                           default_route=none
                           protocol_ipv6=no}
terminal=vt100
security_policy=NONE
root_password=Q7jsh1m6IztTU
```
(continued on next page)

```
(continued from previous page)

name_service=NONE
timeserver=localhost
lokoya# lofiadm -a /bidvd9/s9u5.s1
/dev/lofi/1
lokoya# mount /dev/lofi/1 /mnt
lokoya# ls -al /mnt/etc/sysidcfg
lrwxrwxrwx   1 root     other          24 Nov 28 16:38 /mnt/etc/
sysidcfg -> ../tmp/root/etc/sysidcfg
lokoya# rm /mnt/etc/sysidcfg
lokoya# cp /bidvd9/sysidcfg /mnt/etc/sysidcfg
lokoya# umount /mnt
lokoya# lofiadm -d /dev/lofi/1
```

Note – For the off-network automated installation, the host name, IP address, netmask, and IPv6 specification *must* be in the `sysidcfg` file.

h. Concatenate and write the required information to the DVD writer.

In our example, we concatenated the VTOC, HSFS image, padding, and unmodified images of slices 1 through 5 into one image and wrote the image to the DVD writer on device `c3t2d0` by the `cdrw` command as follows.

```
lokoya# cat s9u5.dvd.vtoc s9u5.s0 pad.s0 \
s9u5.s1 s9u5.s2 s9u5.s3 s9u5.s4 s9u5.s5 >bidvd9.image
lokoya# cdrw -l
Looking for CD devices...
    Node            Connected Device            Device type
---------------------+-------------------------------+---------------
 cdrom0               | TOSHIBA  DVD-ROM SD-R6112 1731 | CD Reader/Writer

lokoya# cdrw -M
Device : TOSHIBA  DVD-ROM SD-R6112
Firmware : Rev. 1731 (05/27/03)
Media is blank
lokoya# cdrw -d cdrom0 -i /bidvd9/bidvd9.image
Initializing device...done.
Preparing to write DVD
Writing track 1...done.
Finalizing (Can take upto 4 minutes)...done.
```

4. Test the bootable installation DVD.

To validate the newly created bootable installation DVD, place it in the DVD drive of the installation client, `barossa`. For our example, the client is off-network while the installation occurs and the `sysidcfg` file in the `/etc` directory of slice 1 of the DVD was modified, as shown in Step 3 on page 9. After issuing the `boot cdrom` command with the `- install` options to initiate the automated installation, `barossa` boots from the DVD and performs an automated installation of the Solaris 9 OS.

For our example, this process appeared as follows.

```
{2} ok boot cdrom - install
Resetting ...

screen not found.
Can't open input device.
Keyboard not present.  Using ttya for input and output.

Sun Ultra 60 UPA/PCI (2 X UltraSPARC-II 450MHz), No Keyboard
OpenBoot 3.27, 2048 MB memory installed, Serial #13409146.
Ethernet address 8:0:20:c7:7:ff, Host ID: 80c707ff.



Initializing Memory
Rebooting with command: boot cdrom - install
Boot device: /pci@1f,4000/scsi@3/disk@6,0:f  File and args: - install
SunOS Release 5.9 Version Generic_112233-10 64-bit
Copyright 1983-2003 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
Configuring /dev and /devices
Using RPC Bootparams for network configuration information.
```

(continued on next page)

(continued from previous page)

```
Skipping interface hme0
Searching for configuration file(s)...
Search complete.
syslog service starting.
savecore: no dump device configured
Running in command line mode

Please wait while the system information is loaded...


Please wait while the system is configured with your settings...
Generating software table of contents [this may take a few minutes...]
Table of contents complete.
Starting Solaris installation program...
Searching for JumpStart directory...
 <<< using DVD install_config >>>
not found
Checking rules.ok file...
Using profile: bidvd.profile
Executing JumpStart preinstall phase...
Searching for SolStart directory...
Checking rules.ok file...
Using begin script: install_begin
Using finish script: patch_finish
Executing SolStart preinstall phase...
Executing begin script "install_begin"...
Begin script install_begin execution completed.

Processing default locales
    - Specifying default locale (en_US.ISO8859-1)

Processing profile
    - Opening Flash archive
    - Validating Flash archive
    - Selecting all disks
    - Configuring boot device
    - Using disk (c0t0d0) for "rootdisk"
    - Configuring swap (c0t0d0s1)
    - Configuring  (c0t0d0s7)
    - Configuring / (c0t0d0s0)
```

(continued on next page)

(continued from previous page)

```
    - Deselecting unmodified disk (c0t1d0)
    - Deselecting unmodified disk (c1t8d0)
    - Deselecting unmodified disk (c1t9d0)
    - Deselecting unmodified disk (c1t10d0)
    - Deselecting unmodified disk (c1t11d0)
    - Deselecting unmodified disk (c1t12d0)
    - Deselecting unmodified disk (c1t13d0)
    - Deselecting unmodified disk (c2t0d0)
    - Deselecting unmodified disk (c2t1d0)
    - Deselecting unmodified disk (c2t2d0)
    - Deselecting unmodified disk (c2t3d0)
    - Deselecting unmodified disk (c2t4d0)
    - Deselecting unmodified disk (c2t5d0)

Verifying disk configuration
    - WARNING: Unused disk space (c0t0d0)

Verifying space allocation
    NOTE: 1 archives did not include size information

Preparing system for Flash install

Configuring disk (c0t0d0)
    - Creating Solaris disk label (VTOC)

Creating and checking UFS file systems
    - Creating / (c0t0d0s0)

Beginning Flash archive extraction

Extracting archive: s9
    Extracted    0.00 MB (  0% of 1025.28 MB archive)
    Extracted    1.00 MB (  0% of 1025.28 MB archive)
[ ... additional output omitted for brevity ... ]

    Extracted 1024.00 MB ( 99% of 1025.28 MB archive)
    Extracted 1025.00 MB ( 99% of 1025.28 MB archive)
    Extracted 1025.28 MB (100% of 1025.28 MB archive)
    Extraction complete
```

(continued on next page)

(continued from previous page)

```
Customizing system files
    - Mount points table (/etc/vfstab)
    - Unselected disk mount points (/var/sadm/system/data/
vfstab.unselected)
    - Network host addresses (/etc/hosts)

Cleaning devices

Customizing system devices
    - Physical devices (/devices)
    - Logical devices (/dev)

Installing boot information
    - Installing boot blocks (c0t0d0s0)

Installation log location
    - /a/var/sadm/system/logs/install_log (before reboot)
    - /var/sadm/system/logs/install_log (after reboot)

Flash installation complete
Executing JumpStart postinstall phase...

The begin script log 'begin.log'
is located in /var/sadm/system/logs after reboot.

Apr 6 16:32:13 rpcbind: rpcbind terminating on signal.
syncing file systems... done
rebooting...
Resetting ...

screen not found.
Can't open input device.
Keyboard not present.  Using ttya for input and output.
Sun Ultra 60 UPA/PCI (2 X UltraSPARC-II 450MHz), No Keyboard
OpenBoot 3.27, 2048 MB memory installed, Serial #13409146.
Ethernet address 8:0:20:c7:7:ff, Host ID: 80c707ff.
```

(continued on next page)

(continued from previous page)

```
Initializing Memory
Rebooting with command: boot
Boot device: /pci@1f,4000/scsi@3/disk@0,0:a  File and args:
SunOS Release 5.9 Version Generic_112233-10 64-bit
Copyright 1983-2003 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
configuring IPv4 interfaces: hme0.
Hostname: barossa
Configuring /dev and /devices
Configuring the /dev directory (compatibility devices)
The system is coming up.  Please wait.
Configuring network interface addresses: hme0.
starting rpc services: rpcbind done.
Setting netmask of hme0 to 255.255.255.0
Setting default IPv4 interface for multicast: add net 224.0/4:
gateway barossa
syslog service starting.
volume management starting.
Creating new RSA public/private host key pair
Creating new DSA public/private host key pair
The system is ready.

barossa console login:
```

About the Author

John S. Howard is a Senior Staff Engineer with Sun's Reference Architecture Engineering group. He has over 20 years of experience in software engineering and systems administration on a diversity of platforms. John is currently designing data center architectures as well as products and tools to deploy and manage integrated software stacks.

John is the author of numerous technical papers and co-author of the books *JumpStart Technology: Effective Use in the Solaris™ Operating Environment* and *Boot Disk Management: A Guide for the Solaris™ Operating Environment*.

Ordering Sun Documents

The SunDocs℠ program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at: `http://www.sun.com/blueprints/online.html`