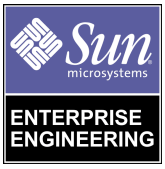
Building a Bootable JumpStart™ Installation CD-ROM

By John S. Howard - Enterprise Engineering

Sun BluePrints™ OnLine - March 2001



Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Part No.: 816-0092-10
Revision 01, 03/02/01
Edition: March 2001

Please
Recycle

Building a Bootable JumpStart™ Installation CD-ROM

In some situations it may not be possible to use a JumpStart™ server, yet an automated "hands-free" installation of the Solaris™ Operating Environment (Solaris OE) is needed. This article details how to create a bootable JumpStart installation CD-ROM, that is, essentially putting a JumpStart server onto a CD. This CD can be used to complete a standardized, hands-free Solaris OE installation in environments where the disk space or networking constraints do not allow for a JumpStart server. This article is intended for system administrators with a moderate to advanced knowledge of JumpStart technology who desire to use JumpStart technology in a constrained environment.

This article is an excerpt from the upcoming Sun BluePrints™ book titled "*JumpStart™ Technology: Effective use in the Solaris™ Operating Environment*" (ISBN# 0-13-062154-4) by John S. Howard and Alex Noordergraaf. This book is scheduled for publication by Prentice Hall in the Summer of 2001 and will be available through `http://www.sun.com/books`, amazon.com, fatbrain.com and Barnes & Noble bookstores.

In this article:

- "Bootable CD Structure" on page 2
- "Procedure Overview" on page 5
- "Procedure Specifics" on page 5
- "Testing the Bootable Installation CD" on page 10

This article examines the structure of a bootable Solaris 2.6 OE (for SPARC™ Platform Edition) CD and discusses modifications to the default installation scripts that allow performing a JumpStart installation from CD. Further, this article describes how to create a bootable Solaris 2.6 OE installation CD for the SPARC platform. Because of the availability of CDRW (CD Read and Write) utilities in the OS, a Solaris 8 OE system is used to write the Solaris 2.6 bootable installation CD. While several different approaches and software applications are available for writing CDs, this article uses commands available only in the standard Solaris 8 OE to write the bootable installation CD.

Bootable CD Structure

The structure of the bootable installation CD may vary with different versions of the Solaris OE, partly because of changes required for the support of additional hardware architectures. Additionally, changes to the Solaris OE from version to version may necessitate changes in the CD (or number of CDs) required to install the Solaris OE.

While there may be structural variances across versions of the Solaris OE, the concepts and procedures presented here can be adapted or extended to create a bootable installation CD for any current version of the Solaris OE.

A bootable Solaris OE CD has several components in common with any other hard disk. The boot CD 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 CD has six slices. Although the Solaris OE imposes the partitioning of the CD into six slices, it is important to note that the CD contains only one *session*; this distinction will be important when we write the CD. The following is an examination of the VTOC and the six slices of the Solaris 2.6 OE installation CD.

The VTOC

The VTOC is located at cylinder 0, sector 0 on the CD. The VTOC of any disk device may be examined with the `prtvtoc` command. The VTOC of the Solaris 2.6 OE Software CD (commonly referred to as the installation CD) is as follows:

```
caliburn# /etc/init.d/volmgt stop
caliburn# 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
*
* Unallocated space:
*   First  Sector    Last
*   Sector   Count    Sector
*   1091200   3200  1094399
*   1104000 206720  1310719
*
*                           FirstSector     Last
* Partition  Tag  Flags   Sector  Count   Sector  Mount Directory
       0      4   10           0  988160   988159
       1      2   10      988160  103040  1091199
       2      0   00     1091200    3200  1094399
       3      0   00     1094400    3200  1097599
       4      0   00     1097600    3200  1100799
       5      0   00     1100800    3200  1103999
caliburn# /etc/init.d/volmgt start
```

Note – CD and floppy volume management must be stopped to execute the `prtvtoc` command on a CD. Volume management is then restarted after executing the `prtvtoc` command.

In contrast to a hard disk, the disk geometry that the Solaris OE uses for a CD provides no distinction between a cylinder and a track. As is evident from the `prtvtoc` output, the disk label used for a CD defines a cylinder as being composed of 1 track. Further, the `prtvtoc` output verifies that each track is defined as having 640 sectors and 1 sector is 512 bytes.

Note – The Solaris OE requires that all UFS filesystems align on a cylinder boundary. For a CD, this means that all UFS filesystems on the CD must begin on a sector that is a multiple of 640.

The Slices

By reading the VTOC, Solaris OE sees the CD as having 6 slices. Those 6 slices are:

- Slice 0 contains the Solaris OE packages to be installed and is on the HSFS (High Sierra File System) partition of the CD.
- Slice 1 contains the generic kernel and what will be the systems `/` (`root`) directory 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 only exist to provide hardware architecture specific boot blocks. As new hardware architectures are added and old architectures reach their end-of-life, the uses of these slices may change. The file `.slicemapfile` in the top-level directory of slice 0 will contain the mapping of the slice to architecture supported.

As noted earlier, slice 0 is HSFS, with all other slices being UFS partitions. Slice 0 is also the largest of the slices and can incorporate any unused space on the CD. The procedures detailed in this article augment the installation procedures in slice 0. However, a fixed upper limit in available space for slice 0 limits your modifications. The total space available on a standard CD is 640mb. The distribution media for Solaris 2.6 OE supports 4 architectures. If the bootable installation CD being created is only required to support one architecture, the space (slices) used by the unneeded architectures may be incorporated into slice 0; thereby, enlarging slice 0 at the expense of losing the ability to boot other architectures from that CD.

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 each of those slices. 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 in the release of Solaris 2.5 OE to take advantage of the hardware-independent nature of the kernel to optimize the use of space on the CD.

Procedure Overview

Generally, the procedure discussed in this article extracts the contents of slice 0, then splices the desired installation behaviors into the contents of slice 0. The modifications made to slice 0 on the CD will configure the installation process from the CD to partition `c0t0d0` as the boot device, then do a fully-automated installation of the Solaris 2.6 OE. Finally, the Solaris OE will be installed from the CD.

The modified slice 0 is then written to the bootable installation CD being created. Since no changes to the mini-root or supported architectures are required, slices 1 through 5 are extracted from the Solaris 2.6 OE software CD and written (unchanged) to the bootable installation CD being created.

At a high level, the procedure to create a bootable CD is as follows:

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

This procedure can also be used to create a Bootable CD without the JumpStart Technology by omitting step 2.

Procedure Specifics

For this example, `caliburn` is an Ultra Enterprise™ 420R server running the Solaris 8 OE with the Solaris 8 OE CD creation utilities installed. The server `caliburn` has the a CD-ROM writer connected at `c3t2d0`.

1. The presence of the Solaris OE CD creation utilities is verified. The Solaris 2.6 OE installation media is already mounted and `/bicd` is used as the work area.

The filesystem `/bicd` is a 2 GB UFS filesystem; create this filesystem as follows:

```
caliburn# pkginfo SUNWmkcd SUNWcdrw
system      SUNWcdrw       CD read and write utility for Solaris
system      SUNWmkcd       CD creation utilities
caliburn# newfs -m 1 /dev/rdsk/c0t1d0s0
newfs: construct a new file system /dev/rdsk/c0t1d0s0: (y/n)? y
/dev/rdsk/c0t1d0s0: 4194828 sectors in 1452 cylinders of 27
tracks, 107 sectors
        2048.3MB in 46 cyl groups (32 c/g, 45.14MB/g, 7488 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 92592, 185152, 277712, 370272, 462832, 555392, 647952, 740512,
833072, 925632, 1018192, 1110752, 1203312, 1295872, 1388432,
1480992, 1573552, 1666112, 1758672, 1851232, 1943792, 2036352,
2128912, 2221472, 2314032, 2406592, 2499152, 2591712, 2684272,
2776832, 2869392, 2958368, 3050928, 3143488, 3236048, 3328608,
3421168, 3513728, 3606288, 3698848, 3791408, 3883968, 3976528,
4069088, 4161648,
caliburn# mkdir /bicd
caliburn# mount /dev/dsk/c0t1d0s0 /bicd
```

CD and floppy volume management is stopped, and the work area is populated by extracting the partitions from the Solaris 2.6 OE software CD. Because the contents of slice 0 need to be manipulated, use `cpio` to copy out partition 0. Because no changes are made to the contents of slices 1 through 5, use the `dd` command to remove these slices from the CD:

```
caliburn# mkdir /bicd/s0
caliburn# cd /cdrom/sol_2_6_598_sparc_smcc_svr/s0
caliburn# find . -print |cpio -pudm /bicd/s0
943076 blocks
caliburn# /etc/init.d/volmgt stop
caliburn# cd /bicd
caliburn# for i in 1 2 3 4 5
> do
> dd if=/dev/dsk/c0t6d0s${i} of=sol2.6.s${i} bs=512
> done
103040+0 records in
103040+0 records out
3200+0 records in
3200+0 records out
3200+0 records in
3200+0 records out
3200+0 records in
3200+0 records out
3200+0 records in
3200+0 records out
```

Additionally, because the slice layout of the bootable installation CD being created does not vary from the slice layout of the Solaris 2.6 OE Software CD, use the VTOC from the Software CD later for the bootable installation CD. Use the `dd` command to copy the VTOC from the CD and restart volume management:

```
caliburn# dd if=/dev/dsk/c0t6d0s0 \
of=/bicd/sol2.6.cdrom.vtoc \
bs=512 count=1
1+0 records in
1+0 records out
caliburn# /etc/init.d/volmgt start
```

Several choices are available if the slice layout of the CD being created needs to vary from that of the Software CD (i.e., if the VTOC needs to be changed). CD creation software, such as Gear Pro for UNIX® operating system or Young Minds, could be used to generate a correct and valid VTOC. Alternately, a new VTOC

and disk label can be created progammatically, by creating and writing the `dkl_vtoc` and `dk_label` structures, respectively. For detailed information on these structures and their use, see `/usr/include/sys/dklabel.h`.

2. Modify the default installation behaviors in slice 0 by removing the contents of the `.install_config` directory and add the desired JumpStart rules and profile to the following directory:

Note – 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, they should be placed in the `.install_config` directory as well.

```
caliburn# cd /jumpstart
caliburn# cp Profiles/S2.6-Standard.profile \
/bicd/s0/.install_config/S2.6-Standard.profile
caliburn# cat /bicd/s0/.install_config/S2.6-Standard.profile
install_type   initial_install
system_type    standalone
usedisk        c0t0d0
partitioning   explicit
filesys        c0t0d0s0    691:2040       /
filesys        c0t0d0s1    1:690          swap
cluster        SUNWCall
package        SUNWpmu     delete  # Power Management binaries
package       SUNWpmr     delete  # Power Mgmt config file and rc script
package      SUNWpmowr   delete  # Power Management OW Utilities, (Root)
package       SUNWpmowu   delete  # Power Management OW Utilities, (Usr)
package        SUNWpmowm   delete  # Power Management OW Util Man Pages
caliburn# cat rules
any - - S2.6-Standard.profile -
caliburn# ./check rules
Validating rules...
Validating profile Profiles/S2.6-Standard.profile...
The custom JumpStart configuration is ok.
caliburn# cp rules.ok /bicd/s0/.install_config/rules.ok
```

The setup of the installation profile directory is controlled by the `profind` script, which must be modified to point the configuration directory (that is used by the JumpStart software) to the `.install_config` directory on the bootable installation CD.

Perform this modification by editing the `/bicd/s0/Solaris_2.6/Tools/Boot/usr/sbin/install.d/profind` script and replacing the `cdrom()` function with the following function:

```
cdrom()
{
    gettext " <<< using CD default >>>"; echo
    rmdir ${SI_CONFIG_DIR}
    ln -s /cdrom/.install_config ${SI_CONFIG_DIR}
    exit 0
}
```

3. At this point, the VTOC and the individual slices are combined into one image to be written to a blank CD. To accomplish this, use the `mkisofs` command (`mkisofs` is a component of the `SUNWmkcd` Solaris 8 OE software package) to convert the modified slice 0 in the `/bicd/s0` work area into an HSFS filesystem:

```
caliburn# mkisofs -R -d -L -l -o /bicd/sol2.6.S0 /bicd/s0

             .
             .
             .
Total extents actually written = 247000
Total translation table size: 0
Total rockridge attributes bytes: 2196003
Total directory bytes: 8787968
Path table size(bytes): 55842
Max brk space used b84000
247000 extents written (482 Mb)
caliburn# dd if=/bicd/sol2.6.S0 of=/bicd/new.sol2.6.s0 \
bs=512 skip=1
987999+0 records in
987999+0 records out
caliburn# rm /bicd/sol2.6.S0
```

Note – The `mkisofs` command creates a VTOC at offset 0 within this image. Because the newly created HSFS image is combined with the other slices, the VTOC created for this HSFS image is invalid. Use the `dd` command to remove this invalid VTOC from the modified slice 0 HSFS image.

Because the VTOC specifies a size for slice 0, it must be padded to maintain the validity of the VTOC and also to maintain correct cylinder boundaries. The size of the pad is computed by adding 1 to the number of sectors in the HSFS slice 0 image (to account for the added VTOC), then subtracting that sum from the number of sectors (reported by `prtvtoc` in step 1) in the unmodified slice 0 on the CD. The pad is created with the `dd` command to read the appropriate number of 0's from `/dev/zero`:

```
caliburn# bc
988160 - (987999 + 1)
160
^d
caliburn# dd if=/dev/zero of=pad.s0 bs=512 count=160
160+0 records in
160+0 records out
```

The VTOC, HSFS image, padding, and unmodified images of slices 1 through 5 are then concatenated into one image and written to the CD-ROM writer on device `c3t2d0` by the `cdrw` command:

```
caliburn# cd /bicd
caliburn# cat sol2.6.cdrom.vtoc new.sol2.6.s0 pad.s0 \
sol2.6.s1 sol2.6.s2 sol2.6.s3 sol2.6.s4 sol2.6.s5 >bicd2.6.image
caliburn# cdrw -d /dev/rdsk/c3t2d0s2 -i bicd2.6.image
Initializing device...done.
done.
Finalizing (Can take upto 4 minutes)...done.
```

Testing the Bootable Installation CD

To validate the newly created bootable installation CD, place it in the CD drive of the installation client, `wasabi`. After issuing the OBP `boot cdrom` command, `wasabi` boots from the CD and performs a hands-free installation of the Solaris 2.6 OE. As with any installation, `sysidtool` needs all installation client identification information such as hostname, ip address, time zone, etc. For a hands-free 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. For clarity and simplicity in this example, these test

systems are not NIS+ clients (no name services are used) and the minimal information in the `/etc/bootparams` and `/etc/ethers` files are set-up on the server, `caliburn`:

```
caliburn# cat /etc/bootparams
wasabi sysid_config=caliburn:/jumpstart/Sysidcfg/Solaris_2.6
caliburn# cat /etc/ethers
8:0:20:7c:ff:dd wasabi
caliburn# cat /etc/hosts
#
# Internet host table
#
127.0.0.1    localhost
10.9.5.62    caliburn  loghost
10.9.5.10    wasabi
caliburn# cat /jumpstart/Sysidcfg/Solaris_2.6/sysidcfg
#
# Copyright (c) 2000 by Sun Microsystems, Inc.
# All rights reserved.
#
#ident"@(#)sysidcfg 1.100/09/12   SMI"
#

system_locale=en_US
timezone=US/Pacific
network_interface=hme0 {netmask=255.255.255.0}
terminal=vt100
root_password=P.rYjsh8AcRbg
name_service=NONE
timeserver=localhost
```

Note – The server `caliburn` is not acting as a JumpStart server, it is only serving the `sysidcfg` file which is pointed to by the `sysid_config` parameter in the `/etc/bootparams` file.

The installation client `wasabi` is at the OBP `ok` prompt. Issue the `boot cdrom` command with the `- install` parameters to initiate the hands-free installation, as follows:

```
ok boot cdrom - install
Resetting ...

Sun Ultra 1 UPA/SBus (UltraSPARC 167MHz), No Keyboard
OpenBoot 3.25, 128 MB memory installed, Serial #8191962.
Ethernet address 8:0:20:7c:ff:dd, Host ID: 807cffdd.

Initializing Memory
Rebooting with command: boot cdrom - install
Boot device: /sbus/SUNW,fas@e,8800000/sd@6,0:fFile and args: -
install
SunOS Release 5.6 Version Generic_105181-05 [UNIX(R) System V
Release 4.0]
Copyright (c) 1983-1997, Sun Microsystems, Inc.
Configuring devices...
Using sysid configuration file 10.9.5.62:/jumpstart/Sysidcfg/
Solaris_2.6/sysidcfg
The system is coming up.  Please wait.
Starting remote procedure call (RPC) services: sysidnis done.
Starting Solaris installation program...
Searching for JumpStart directory...
<<< using CD default >>>
Checking rules.ok file...
Using profile: S2.6-Standard.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)

Processing profile
    - Selecting cluster (SUNWCall)
    - Deselecting package (SUNWpmu)
    - Deselecting package (SUNWpmr)
    - Deselecting package (SUNWpmowr)
    - Deselecting package (SUNWpmowu)
    - Deselecting package (SUNWpmowm)
    - Selecting locale (en_US)
```

{continued ...}

```
                .
                . ( package and patch installation listing deleted for brevity )
                .

Finish script patch_finish execution completed.
Executing JumpStart postinstall phase...

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

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

syncing file systems... done
rebooting...
Resetting ...

Sun Ultra 1 UPA/SBus (UltraSPARC 167MHz), No Keyboard
OpenBoot 3.25, 128 MB memory installed, Serial #8191962.
Ethernet address 8:0:20:7c:ff:dd, Host ID: 807cffdd.
Initializing Memory
Rebooting with command: boot
Boot device: disk:a  File and args:
SunOS Release 5.6 Version Generic_105181-05 [UNIX(R) System V
Release 4.0]
Copyright (c) 1983-1997, Sun Microsystems, Inc.
configuring network interfaces: hme0.
Hostname: wasabi
Configuring the /devices directory
Configuring the /dev directory
Configuring the /dev directory (compatibility devices)
The system is coming up.  Please wait.
Configuring network interface addresses: hme0.
starting rpc services: rpcbind keyserv done.
Setting netmask of hme0 to 255.255.255.0
Setting default interface for multicast: add net 224.0.0.0:
gateway wasabi
syslog service starting.
Print services started.
volume management starting.
The system is ready.

wasabi console login:
```

Summary

This article provided techniques to augment a CD-ROM based installation with the services and behaviors provided by a JumpStart server. The techniques provided here are suitable for situations when a hands-free Solaris OE installation is necessary but the use of a JumpStart server is not possible.

Additionally, this article detailed a procedure to create a bootable installation CD, examined the structure of a bootable Solaris OE CD, and provided specifics on the modification of the installation behaviors.

References

Advanced Installation Guide, `http://docs.sun.com`

Noodergraaf, Alex, *JumpStart Architecture and Security Scripts for the Solaris Operating Environment - Part 1*, Sun Blueprints OnLine, July 2000.
`http://www.sun.com/blueprints/0700/jssec.pdf`

Snevely, Rob, *JumpStart: NIS and sysidcfg*. Sun Blueprints OnLine, October 1999.
`http://www.sun.com/blueprints/1099/jumpstart.pdf`

Author's Bio: John S. Howard

John S. Howard has over 19 years experience in software engineering and systems administration. As a Senior Staff Engineer in the Enterprise Engineering group at Sun Microsystems he is currently working on projects for enhancing system availability and serviceability.

Prior to his position in Enterprise Engineering, John worked as an Area System Support Engineer with Sun Enterprise Services. As an ASSE, he was responsible for providing escalation management and backline system support for problem isolation and resolution in the areas of clustered systems, storage subsystems and the Solaris kernel. In addition to these support functions, he developed and performed Reliability, Accessibility, and Serviceability (RAS) studies of customer datacenters.