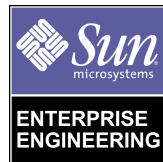




JumpStart™ Mechanics: Using JumpStart Application for Hands-Free Installation of Unbundled Software

By John S. Howard - Enterprise Engineering

Sun BluePrints™ OnLine - May 2000



<http://www.sun.com/blueprints>

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

Part No.: 806-5539-10
Revision 01, May 2000

Copyright 2000 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

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, Sun BluePrints, Sun Enterprise Volume Manager, JumpStart, SunOS and Solaris are trademarks, registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

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 2000 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, The Network Is The Computer, Sun BluePrints, Sun Enterprise Volume Manager, JumpStart, SunOS, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, 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.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



Please
Recycle



Adobe PostScript

JumpStart™ Mechanics: Using JumpStart Application for Hands-Free Installation of Unbundled Software

Introduction

Automating and standardizing the installation of the Solaris™ Operating Environment along with the associated unbundled software products and datacenter management tools is one of the largest challenges facing System Administrators of many systems. In this article, an overview of the mechanics of JumpStart™ application is presented, the JumpStart mini-root and boot process are examined and an example of using JumpStart application to perform a site-standard, hands-free install of Solaris 2.6 Operating Environment and Sun Enterprise Volume Manager™ (SEVM) 2.6 is given. This article is the first of several articles on advanced JumpStart software techniques, these articles will examine some of the lesser known (but more powerful) configuration options of JumpStart application and also cover customizing the JumpStart mini-root and boot process.

JumpStart Application Overview

JumpStart application is Sun Microsystems, Inc. framework for automating and simplifying the Solaris Operating Environment installation process. JumpStart application is responsible for providing two primary services. First JumpStart application provides a framework to facilitate or completely automate all aspects of installation of the Solaris Operating Environment. The second service is to provide a hardware architecture neutral Solaris Operating Environment kernel called the mini-

root. In addition to being used as the operating environment by the installation programs, the mini-root is also a failsafe boot media. As a failsafe boot media, the mini-root is used to provide a stable and "well known" operating environment from which to effect system recovery procedures.

The JumpStart Application Installation Framework

JumpStart application is based on a client/server architecture with a number of server components providing services throughout the installation process. From a high level, the components of a JumpStart system are:

TABLE 1 JumpStart server components

Install Client	Target system to be installed or upgraded.
Boot Server	The network entity which provides a failsafe OS to the installing client. The boot image is architecture neutral providing base OS services to all hardware supported by that OS release.
Configuration Server	Helps the client determine its unique profile. Partition sizes, the list of software to install, begin and finish scripts, etc. are all specified in a profile served by the configuration server.
Install Server	The source of the software packages which are to be installed on the client.

In practice, all three JumpStart servers are often co-located on the same host, but this is not a requirement.

Booting a client over the network is initiated when that client broadcasts a RARP request. Being an IP broadcast request, it is not typically forwarded beyond subnet boundaries. A server responding to this request, maps the client's MAC address to an IP and hostname, returning this data to the client. The booting client is now aware of two IP addresses: Its own, and that of a RARP server. In order to simplify the network booting algorithm, the client assumes that any server which responds to the RARP request is also capable of responding to subsequent requests for data. As such, the RARP server is now declared the server for kernel delivery and boot parameter requests. A system which provides all three services (RARP, TFTP, `bootparam`) is referred to the boot server. The mini-root kernel is read from the boot server via TFTP. Once initially loaded to the client, the kernel will mount its root file system and begin processing with `init`, just as it would if booted from a disk.

As with the CD-ROM failsafe boot image, all filesystems except `/tmp` are mounted read-only. The location of the root file system is determined by means of a `bootparam` request. The `bootparam` daemon on the boot server will respond with

the values configured into its boot parameter database (either `/etc/bootparams` or a bootparams NIS or NIS+ map as determined by `nsswitch.conf` on the boot server). A number of parameters are conventionally used in bootparams to help the client configure itself such as:

- root
- swap
- ns (name service)
- term (terminal type)

In addition to providing a failsafe boot device, the JumpStart servers provide a framework which allow the automatic determination of a system configuration and subsequent hands-free installation. This automation framework uses a rule set to match install clients to a pre-defined installation profile, allowing the system administrator to control and automate the installation of the Solaris Operating Environment and unbundled software.

For hands-free installation in environments where name services are not used or to override defaults in the name services, the `sysidcfg` file is used. The `sysidcfg` file provides the information required by the install program for such configuration parameters, such as:

- the time host
- timezone
- root password
- locale information

Consult the `bootparams(4)` and `sysidcfg(4)` man pages for details. Additionally several previous Sun BluePrints™ Online articles have covered JumpStart server architecture and configuration in depth. However, to take advantage of the full power of JumpStart application we will first need to understand the mini-root; the pre-installation environment served up by the JumpStart server.

Mini-root

The mini-root is a small footprint version of the normal SunOS™ kernel, drivers and support utilities required to boot the system. As the Solaris Operating Environment mini-root is required to operate any supported Sun™ hardware configuration without modification, the mini-root is designed to be architecture and hardware neutral. The mini-root is used when a failsafe boot image or known good kernel is required for maintenance or repair procedures. Also the mini-root is used when a minimal operating system is required to perform some recovery or maintenance activity on the system. The mini-root is most widely used as the base operating system from which the system is booted at system install time - usually from a CD-ROM.

The mini-root must be generic enough and feature-rich enough to bring a system with an arbitrary hardware configuration up to a minimum threshold OS level. The mini-root must support all Solaris Operating Environment supported hardware configurations well enough to allow install and repair tasks.

In order for the mini-root to function in such diverse environments, its designers were forced to employ numerous workarounds and shortcuts. The majority of such workarounds use the now familiar software switch mechanism `/platform/`uname -m`` to locate architecture-specific binaries or configuration data. A number of items found on a conventional Solaris Operating Environment root file system may not be available on the mini-root image. This is most often due to the need to conserve space on the mini-root. Frequently, however, a set of features is simply not necessary or cannot be provided in an architecture neutral way. In such cases, the corresponding packages are not installed when the mini-root is created. For example, source and binary compatibility items are not available on the mini-root image.

Mini-root Kernel and Drivers

The mini-root kernel is not architecture specific, the software switch mechanism discussed earlier is used to help ensure that the same filesystem image will operate a variety of hardware platforms. In addition, the mini-root boot image does not provide a large number of device drivers. The mini-root kernel does not pre-load or force-load a large number of drivers default, but the mini-root operates in the same way as the normal kernel with respect to loading drivers. Drivers may be loaded and configured in the same manner and via the same mechanisms as with the full Solaris Operating Environment kernel (i.e., via `add_drv`).

Complex system resources such as shared memory or message queues may not work as expected, because these are not necessary to fulfill the charter of the mini-root filesystem. The inter-process communication (IPC) drivers are still present, however, and will load if an application calls for a message queue or shared memory segment.

Mini-root File Systems

The file systems of interest on a system booted with mini-root are `/` (root), `/usr`, and `/tmp`. The mini-root root file system has all of the fundamental utilities one might expect on a root file system, as well as those scripts and configuration files necessary to boot and install the system. However, the mini-root file systems differ slightly from the Solaris Operating Environment filesystems. Because the root and `/`

`usr` file systems are always served from read-only media (CD-ROM or NFS mounted read-only), the mini-root root file systems will always be read-only. It is important to note that `/var` is not its own filesystem, `/var` is symlinked into a space within `/tmp` to allow for a writable `/var/tmp`.

Some key files on root must be writable (such as the hosts file), these are symlinked to per-boot copies stored under `/tmp`. The `/etc/hosts` file, for example, is configured early in the boot process to give the system its network identity. Since `/etc` is on the read-only root file system, a symlink is used to refer to a writable hosts file in `/tmp/root`. `/usr` is mounted read-only via a loopback mount and has most of the standard UNIX® commands and tools. `/tmp` is the only filesystem mounted as a writable device by default. Every file or directory that must be writable at any stage during the boot is placed in a subdirectory of the temporary file system, usually as a subdirectory of `/tmp/root`. The original pathname (such as `/etc/hosts`) is a symbolic link to its writable equivalent in `/tmp/root`. This means that each such file must be built by a start script before it is first needed. UNIX expects `/var` (and in particular `/var/tmp`) to be a writable space. To provide this on a read-only root file system with no sub-mount for `/var`, the mini-root uses a symbolic link approach similar to that used for writable files. `/var` is a symlink to `/tmp/root/var`, which must be built early in the boot process. This is accomplished by making a subtree copy of `/var` from a prototype stored in `/.tmp_proto`. This is copied into `/var/tmp` using `cpio` very early in `rcS` startup.

Mini-root Boot Sequence

The JumpStart boot image and mini-root file system can be loaded using either `ufsboot` or `inetboot`, much as a conventional kernel is. In fact, the entire boot process is identical to a conventional kernel at a low level. The scripted implementation differs on the JumpStart filesystem, however. A conventional system has the need for easy customization and expansion. This is accomplished via the well known `rcS`, `rc2`, and `rc3` driver scripts. These driver scripts call various easily configured scripts within the appropriate `/etc/rc?.d` directory during shutdown and startup. The mini-root file system has no such need for configurability or ease of administration.

Mini-root, as previously described, is most often used from read-only media. Because it is not maintained in the traditional sense, the start scripts are rolled into a very small number of directly called scripts. All Solaris Operating Environment kernels eventually launch `init` as process ID 1 to kick off the OS run. `init` consults its `inittab` to determine what processes to spawn at various run levels. This is where the mini-root and the conventional kernel begin to differ. A conventional kernel will launch the driver scripts for the correct run level (`/sbin/rcS` for run level S, followed by `rc2` and `rc3` for run levels 2 and 3, respectively). Although mini-root launches `/sbin/rcS`, this is not the same driver

script as found in a full (or conventional) Solaris Operating Environment. The mini-root version of `rcS` configures the system directly, then exits. The `inittab` on mini-root also calls `rc2` when moving to multi-user mode. `rc2` is a stub, however, `inittab` instructs `init` to also run a second script when entering run level two: `/sbin/startup` is where the remaining configuration takes place.

A Hands-Free Install With Unbundled Packages

Following is an example of a hands-free install of Solaris 2.6 Operating Environment and SEVM 2.6. In the example, the JumpStart server is `blackmesa` and the install client is `ravenswood`. `ravenswood` will be installed with Solaris 2.6 3/98 Operating Environment and SEVM. Note that SEVM will be installed but not configured. The topic of using JumpStart application to automatically encapsulate the boot disk and populate `rootdg` will be covered in a future Sun BluePrints OnLine article.

▼ To install Solaris 2.6 Operating Environment and SEVM 2.6

1. Configure `blackmesa` as a JumpStart server

```
blackmesa# share -F nfs -o ro,anon=0 /JumpStart
blackmesa# mkdir /JumpStart/Solaris_2.6_0398
blackmesa# cd \
> /cdrom/sol_2_6_hw2_sparc_smcc_dt/s0/Solaris_2.6/Tools
blackmesa# ./setup_install_server /JumpStart/Solaris_2.6_0398
Verifying target directory...
Calculating the required disk space for the Solaris_2.6
product
Copying the CD image to disk...
Install Server setup complete
```

2. Add the SEVM packages into the JumpStart server tree on `blackmesa`

```
blackmesa# cd /cdrom/sun_sevm_2_6
blackmesa# find . -print |cpio -pudm /JumpStart/sun_sevm_2_6
```

3. Add ravenwood as an install client of blackmesa, specifying the directory containing the sysidcfg file

```
blackmesa# ./add_install_client \  
> -p blackmesa:/JumpStart/Solaris_2.6_0398/Solaris_2.6/Configs \  
> ravenwood \  
> sun4u
```

4. Add the location of the profiles and rules to /etc/bootparams

```
blackmesa# echo \  
> "* install_config=blackmesa:/JumpStart/Solaris_2.6_0398/  
Solaris_2.6/Configs" >> /etc/bootparams  
blackmesa# cat /etc/bootparams  
ravenwood root=blackmesa:/JumpStart/Solaris_2.6_0398/  
Solaris_2.6/Tools/Boot install=blackmesa:/JumpStart/  
Solaris_2.6_0398 boottype=:in sysid_config=blackmesa:/  
JumpStart/Solaris_2.6_0398/Solaris_2.6/Configs  
rootopts=:rsize=32768  
* install_config=blackmesa:/JumpStart/Solaris_2.6_0398/  
Solaris_2.6/Configs
```

5. Create or modify sysidcfg

In order to have a hands-free install, all client configuration information must be

```
blackmesa# cd /JumpStart/Solaris_2.6_0398/Solaris_2.6/Configs  
blackmesa# vi sysidcfg  
blackmesa# cat sysidcfg  
name_service=NONE  
network_interface=hme0 {netmask=255.255.255.0}  
root_password=jsPrln7T7Ed4U  
system_locale=en_US  
system_locale=C  
timezone=US/Pacific  
timeserver=blackmesa  
terminal=xterm
```

available to the client installation program at boot. The configuration information can be made available from an NIS+ server, NIS server, the `sysidcfg` file, or by any combination of those sources.

6. Create the installation profile..

```
blackmesa# vi server.profile
blackmesa# cat server.profile
install_type    initial_install
system_type     standalone
partitioning    explicit
filesystems     rootdisk.s0 1516 /
                rootdisk.s1 512 swap
cluster         SUNWCall
package SUNWpmowm      delete
package SUNWpmowr     delete
package SUNWpmowu     delete
package SUNWpmmr      delete
package SUNWpmu       delete
package SUNWpmux      delete
blackmesa# /usr/sbin/install.d/pfinstall -D \
> -c /JumpStart/Solaris_2.6_0398
> server.profile
```

7. Use pkgask(1) to create the response files for automating the installation of the SEVM SUNWasevm and SUNWvmsa packages

Installation of the SUNWasevm and SUNWvmsa prompts the user for location and configuration information. The use of a response file will allow installation of these packages to be automated.

```
blackmesa# pkgask -r SUNWasevm.response \  
> -d /JumpStart/sun_sevm_2_6/Product SUNWasevm  
  
Processing package instance <SUNWasevm> from </JumpStart/  
sun_sevm_2_6/Product>  
  
Sun StorEdge Volume Manager 2.6 AnswerBook  
(all) 263.1.3  
  
The installation options are as follows:  
Option: Description:  
-----  
1. nil:    less than 1 Megabyte disk space required [slowest  
performance].  
2. heavy:  21.97 Megabytes disk space required [best  
performance].  
  
Note: If the install option which you choose below fails  
due to lack of space, try another location, or  
choose a lower install option number.  
Enter the number of an installation option from the list  
above (1 or 2).  
Select an installation option: 2  
  
Installation option: heavy selected.  
  
The next request for input asks you to specify the parent  
directory of AnswerBook  
Make sure to choose a parent directory on a file system big  
enough to accommodate all the files to be moved for the  
INSTALL OPTION you selected.  
  
Specify the parent of the AnswerBook home directory: /opt  
For the heavy option all files will be placed under /opt/  
SUNWasevm.  
  
Response file </JumpStart/Solaris_2.6_0398/Solaris_2.6/  
Configs/SUNWasevm.response> was created.
```

```
Processing of request script was successful.
# pkgask -r SUNWvmsa.response -d /JumpStart/sun_sevm_2_6/
Product SUNWvmsa

Processing package instance <SUNWvmsa> from </JumpStart/
sun_sevm_2_6/Product>

Sun StorEdge Volume Manager, Server Administration
(sparc) 2.6,REV=1.0.1

Where should this package be installed? (default: /opt)
[?,q] /opt

Should the Apache HTTPD (Web Server) included in this package
be installed?
(default: n) [y,n,?,q] n

Should the StorEdge Volume Manager Server be installed on
this system?
(The StorEdge Volume Manager Client will be installed
regardless) (default:
y) [y,n,?,q] y

Response file </JumpStart/Solaris_2.6_0398/Solaris_2.6/
Configs/SUNWvmsa.response> was created.

Processing of request script was successful.
```

8. Create the finish script

```
blackmesa# vi add.vxvm.fin
blackmesa# cat add.vxvm.fin
#!/bin/sh

BASEDIR=/a
SLASHAMNT=${BASEDIR}/mnt
VXPRODUCT=${SLASHAMNT}/Product

mount -F nfs blackmesa:/JumpStart/sun_sevm_2_6 ${SLASHAMNT}
if [ $? != 0 ]; then
    exit 1
fi

echo "Installing SUNWvxvm..."
pkgadd -a ${SI_CONFIG_DIR}/noask -d ${VXPRODUCT} \
-R ${BASEDIR} SUNWvxvm
echo "Installing SUNWvxva..."
pkgadd -a ${SI_CONFIG_DIR}/noask -d ${VXPRODUCT} \
-R ${BASEDIR} SUNWvxva
echo "Installing SUNWasevm..."
pkgadd -a ${SI_CONFIG_DIR}/noask \
-r ${SI_CONFIG_DIR}/SUNWasevm.response -d ${VXPRODUCT} \
-R ${BASEDIR} SUNWasevm
echo "Installing SUNWvmsa..."
pkgadd -a ${SI_CONFIG_DIR}/noask \
-r ${SI_CONFIG_DIR}/SUNWvmsa.response -d ${VXPRODUCT} \
-R ${BASEDIR} SUNWvmsa
echo "Installing SUNWvmman..."
pkgadd -a ${SI_CONFIG_DIR}/noask -d ${VXPRODUCT} \
-R ${BASEDIR} SUNWvmman
echo "Installing SUNWvmdev..."
pkgadd -a ${SI_CONFIG_DIR}/noask -d ${VXPRODUCT} \
-R ${BASEDIR} SUNWvmdev
```

The finish script is what will do the actual installation of the unbundled product packages. The finish script name is specified in the rules file where it will be automatically executed after the installation of the Solaris Operating Environment. The environment variable `SI_CONFIG_DIR` is set by the JumpStart server to evaluate to the path to the JumpStart directory.

9. Create the rules file (or edit an existing rules file). Then verify the profile and rules file, generating `rules.ok`

```
blackmesa# cat rules
#
# The following rule matches any system:
#
any - - server.profile add.vxvm.fin
blackmesa# ./check
Validating rules...
Validating profile server.profile...
The custom JumpStart configuration is ok.
```

10. Begin the install

ravenswood is at the OpenBoot Prom prompt, initiate the installation.

```
ok boot net - install
Resetting ...

Sun Ultra 1 UPA/SBus (UltraSPARC 167MHz), No Keyboard
OpenBoot 3.0, 128 MB memory installed, Serial #8191962.
Ethernet address 8:0:20:7c:ff:ff, Host ID: 807cffff.

Rebooting with command: boot net - install
Boot device: /sbus/SUNW,hme@e,8c00000 File and args: - install
2ec00
hostname: ravenswood
root server: blackmesa
root directory: /JumpStart/Solaris_2.6_0398/Solaris_2.6/Tools/
Boot
SunOS Release 5.6 Version Generic_105181-03 [UNIX(R) System V
Release 4.0]
Copyright (c) 1983-1997, Sun Microsystems, Inc.
Configuring devices...
Using sysid configuration file blackmesa:/JumpStart/
Solaris_2.6_0398/Solaris_2.6/Configs/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 rules.ok from blackmesa:/JumpStart/Solaris_2.6_0398/
Solaris_2.6/Configs.
Checking rules.ok file...
Using profile: server.profile
Using finish script: add.vxvm.fin
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.
```

11. Verify the installation

The installation has completed, ravenwood has Solaris 2.6 Operating Environment and SEVM 2.6 installed.

```
ravenwood console login: root
Password:
```

```
ravenwood# uname -a
SunOS ravenwood 5.6 Generic_105181-03 sun4u sparc SUNW,Ultra-1
ravenwood# cat /etc/release
Solaris 2.6 HW:2 s297s_hw2smccDesktop_09 SPARC
Copyright 1998 Sun Microsystems, Inc. All Rights Reserved.
Assembled on 03 February 1998
ravenwood# pkginfo |grep -y storedge
application SUNWasevm      Sun StorEdge Volume Manager 2.6
AnswerBook
system      SUNWvmdev      Sun StorEdge Volume Manager, Header
and Library Files
system      SUNWvmmman     Sun StorEdge Volume Manager, Manual
Pages
application SUNWvmsa      Sun StorEdge Volume Manager, Server
Administration
system      SUNWvxva      Sun StorEdge Volume Manager, Visual
Administrator
system      SUNWvxvm      Sun StorEdge Volume Manager
ravenwood# ls -al /etc/vx/reconfig.d/state.d
total 4
drwxr-xr-x  2 root      sys          512 May  1 17:06 .
drwxr-xr-x  5 root      sys          512 May  1 17:06 ..
-rw-r--r--  1 root      other        0 May  1 17:06 install-db
```

It is important to note that SEVM is not configured (i.e. rootdg does not exist). The automatic configuration and encapsulation of the root disk after a JumpStart installation will be covered in the next article in this series.

Conclusion

This article has served as an introduction to understanding and utilizing the full capabilities of JumpStart application. In addition, this article has provided the procedures to fully automate the installation of Solaris Operating Environment and unbundled software products. JumpStart application is a powerful and flexible tool that enables the system administrator to have a fully automated and customizable method to standardize Solaris Operating Environment installations. Additionally, JumpStart application provides a framework to help ensure uniformity and adherence to site standards (or conventions) for all Solaris Operating Environment installations. However, this power and flexibility requires careful planning and testing by the System Administrator.

Author's Bio: John S. Howard

John S. Howard is currently a Staff Engineer in the Enterprise Engineering group at Sun Microsystems in San Diego, California. He has worked as a software engineer and systems administrator for the past 19 years. Prior to Enterprise Engineering, John worked in Enterprise Services as an Area System Support Engineer for five years. As an ASSE, he was responsible for developing and performing Reliability, Accessibility, and Serviceability (RAS) studies of customer datacenters and the development of proactive Enterprise RAS Services. Prior to Sun, John held engineering positions at: The Chicago Board of Trade Clearing Corporation, Datalogics Inc, and Rand McNally. Throughout his career he has developed: pagination and publishing software, loose-leaf publishing systems, extensive SGML systems development, database publishing systems, text editors and WYSIWIG systems, and device drivers.