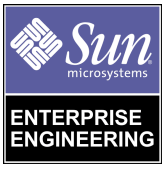
MR System for Rapid Recovery

By John S. Howard - Enterprise Engineering

Sun BluePrints™ OnLine - January 2001



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.: 806-7384-10
Revision 01, 01/12/01
Edition: January 2001

MR System for Rapid Recovery

Enterprise class (or mission-critical) systems have grown to such size and importance that they may no longer tolerate down time for even the most simple of recovery efforts. As the system uptime requirements have become more exacting, the length of time it takes to boot these larger and more complex systems has grown. A moderately sized Sun Enterprise™ 10000 server domain can take 45 minutes to boot using default self-test levels and substantially longer if a more stringent self-test is performed. Traditional recovery tactics on these systems can be frustratingly time consuming—in most part due to the requirement of multiple reboots during recovery.

The time consuming aspect of recovery procedures stems from the fact that the tools necessary to manage and run a system or domain are not in the fail safe boot image (typically a CD-ROM). The persistent storage on these mission-critical systems is almost always managed using products such as Solstice DiskSuite™ application or Veritas Volume Manager (VxVM) which are not found in the base Solaris™ Operating Environment (Solaris OE). These volume management tools are mandatory for the operation of any important UNIX system or datacenter, yet are not available on the CD-ROM so often used as fail safe boot media to effect emergency recoveries.

The configuration and tools described in this article were designed to augment the JumpStart™ boot image to provide a fully functional version of VxVM, available for booting over the network. Although it provides a set of recovery tools, the Mini-root Recovery (MR) system is also valuable as an example of a mechanism by which you may incorporate tools into the fail safe boot media, allowing system administrators to incorporate their site-specific or *home-grown* tools into the fail safe Solaris OE boot image.

MR is unsupported software available in the Tools section of the Sun BluePrints™ OnLine web site. As with any new tool (or existing tool being used in a different way), MR and the procedures provided in this article should be tested before being used in a production environment, or used to recover production systems or data.

The MR system is packaged for ease of installation and made available in the Scripts and Tools section of the Sun BluePrints Web site at `http://www.sun.com/blueprints/tools`.

JumpStart™ Technology and the Mini-Root

JumpStart technology is the framework developed by Sun Microsystems Inc. for automating, customizing, and controlling the Solaris OE installation process. The JumpStart software is responsible for providing two primary services. First, JumpStart software provides a framework to facilitate or completely automate all aspects of installation of the Solaris OE. The second service is to provide a hardware architecture neutral Solaris OE kernel called the mini-root. The mini-root is a small footprint version of the normal SunOS™ kernel, drivers, and support utilities required to boot the system.

The MR system leverages Sun's JumpStart technology and the mini-root to provide an enhanced fail safe boot image containing the tools necessary for system recovery.

Generally, the JumpStart software is a client/server mechanism for:

- Providing a Solaris OE kernel (the mini-root) to a client system via the network
- Providing installation information
- Standardizing and automating installation of the Solaris OE

The MR packages leverage the JumpStart framework to modify the mini-root image on a server. This MR-augmented mini-root is then served to clients to effect a recovery of that client system. For details on JumpStart technology, the architecture and constraints of the mini-root, please refer to "References" on page 12.

MR Overview

The MR system layers on top of a standard JumpStart boot image found on the Solaris OE installation media. The scripts controlling the mini-root boot sequence are modified to provide processing for the `recover` keyword, supplied as an argument to the Open Boot Prom (OBP) `boot` command:

```
racerx# ok boot net - recover
```

This argument will control which services start when the kernel reaches run level 2, and is processed in a similar manner to the `install` keyword used with a JumpStart software installation:

```
racerx# ok boot net - install
```

The recognition of the `recover` keyword causes the booting client to start network services not normally started on a JumpStart boot. Additionally, the client will enter into run level 2 without starting the Solaris OE installation mechanism (`pfinstall`), which is the default behavior for a JumpStart boot beyond run level 1.

Initial Set-up

The MR system was designed and implemented in such a manner as to preserve the default JumpStart mini-root boot behavior. An MR-modified boot image can still be used to install and upgrade clients without requiring special procedures—that is, the base JumpStart technology functionality still exists and functions normally.

Setting up MR is merely a matter of installing a JumpStart server, then splicing in the additional behaviors that you would like from this fail safe boot image. The steps to accomplish this are:

- Copy the JumpStart server image to write enabled media
- Modify the startup routines to accept extra boot arguments
- Install any required device drivers into the mini-root
- Install any application software into the mini-root

Copying JumpStart Image

Copy the JumpStart server software from the CD as normal. For example:

```
racerx# BASEDIR=/JumpStart/Solaris2.6_0599
racerx# export BASEDIR
racerx# cd /cdrom/cdrom0/Solaris_2.6/Tools
racerx# ./setup_install_server $BASEDIR
```

Modifying Boot Startup

The processing for keywords specified at the end of the boot command line is embedded in the shell scripts launched by `init`. MR modifies these scripts such that additional keywords are accepted and the triggers are set for the functions that they control. This is necessary because you want the host to travel beyond single user mode, yet not start the JumpStart install process (`pfinstall`).

Additionally, because most utilities require network services, such as `inetd`, MR configures the mini-root to automatically start network services in `/sbin/sysconfig`.

Installing Device Drivers

A number of additional device drivers may be needed by the tools used in a recovery boot. The VxVM devices, for example, are activated to facilitate the manipulation of volumes while booted from the MR augmented mini-root. This technique can be copied or adapted to install other desired device drivers. For example, by adding the FDDI driver to the mini-root image to permit booting from an FDDI network interface.

Addition of device drivers is performed using the standard Solaris OE mechanism for adding a device driver; the `add_drv` command. However, some additional configuration may be necessary to a few system and driver configuration files to make the added drivers function correctly.

Installing Application Software

Any application software necessary for a recovery should be installed into the JumpStart image as if its root directory were the base of the OS. For most applications this is very straight forward. However, some applications, especially those requiring a license key or a write enabled workspace, may require workarounds or artful manipulations of the mini-root environment.

Pre-Packaged MR Enhancements

The suite of MR packages is designed to handle the majority of the setup and customizations necessary in a consistent and automated manner. As mentioned previously, MR is unsupported software available in the Scripts and Tools section of the Sun BluePrints OnLine web site at `http://www.sun.com/blueprints/`

`tools`. As with any new tool (or existing tool being used in a different way), MR and the procedures provided in this article should be tested before being used in a production environment or used to recover production systems or data.

Each application is bundled into its own separately installable package in order to allow modular addition to the fail safe boot image. You only need to install those components necessary for *your* environment. A few of the current packages are listed in the following sections.

MRbase

MRbase modifies the mini-root start scripts described above and MRbase is a prerequisite for all other packages. The modifications that MRbase makes to the startup scripts provide additional argument processing during entry into run level 2, to detect and process the MR arguments to the booting kernel. MRbase retains the default ability to boot for installation services (the usual JumpStart boot, starting `suninstall`).

The second modification made to the boot image by MRbase is the addition of extra network device drivers. The standard boot image can only boot from a few network interface types. With the exception of ATM, this restriction is not due to any technical reasons, it is only because the non-bootable network interface types do not have drivers on the install CD. The device drivers MRbase adds to the boot image are the drivers for the FDDI networking support. Booting from these additional network interface types still requires adherence to the rules governing placement of Reverse Address Resolution Protocol (RARP) servers on the same subnet or having directed broadcasts configured on the router for that subnet.

Please note that just as Solaris OE does not support booting from an ATM interface, MRbase also does not support booting from an ATM interface.

MRbase packages are specific to the Solaris OE version, MRbase packages are available for versions 2.6, 7, and 8 of the Solaris OE.

MRvx

The MRvx package provides VxVM (version 2.6 or 3.0.4) support to the mini-root kernel. This package includes everything normally found in the SUNWvxvm or VRTSvxvm packages installed on a host. However, no man pages or GUI utilities are included.

The MRvx package provides the boot image with the ability to directly access and manipulate VxVM objects. MRvx includes the drivers and utilities normally found in VxVM, relocated into the correct places for use while booted on the mini-root. In addition, MRvx provides a setup script which facilitates starting and using VxVM on a read-only root filesystem as found on the mini-root.

Additionally, it is important to note that because the VxVM setup is done in the mini-root some devices may be completely inaccessible or accessible via a different controller number than they are when booted off of the system's usual system disk. In general, any device that required changes to the `sd.conf` file will not be usable by MRvx above lun 0 or may not be visible at all. Devices such as EMC storage units or Sun StorEdge™ A3500 arrays configured for multi-lun support are the most common devices that fall into this category.

MRvx provides a quick and ready method to boot off of a device other than the system disk and the ability to mount and modify the root filesystem without having to unencapsulate. This saves you the time of having to unencapsulate and then re-encapsulate and the several associated reboots involved in that procedure. It also saves you the time of a mirror re-synchronization, as a re-synchronization is no longer necessary since MRvx enables you to work with the mirrored devices directly. Additionally, the use of MRvx avoids the potential of human error during the complicated and often manual unencapsulation/re-encapsulation procedure.

MRvx is essential for those instances when unencapsulating is impossible or dangerous. If the system administrator has done anything that has invalidated the disk's saved Volume Table of Contents (VTOC), such as resizing a filesystem on the system disk, replacing the encapsulated boot device without making the underlying partitions, or moving a subdisk on the system disk, then the VTOC saved by the encapsulation process no longer matches the slice boundaries on the system disk. In these situations, unencapsulation may be impossible or runs a high probability of data loss or corruption.

MRvx packages are specific for specific Solaris OE and VxVM versions: MRvx packages are available for VxVM version 2.6; version 2.6 of the Solaris OE; as well as VxVM version 3.0.4 on versions 7 and 8 of the Solaris OE.

MRmd

MR support for Solstice DiskSuite (SDS) 4.2 under the Solaris 2.6 and Solaris 7 OEs is provided by the MRmd642 and MRmd742 packages, respectively. These packages incorporate the SDS command line interface, SDS patch 106627, and the SDS Graphical User Interface (GUI) into MR.

MRsbu

MRsbu provides Solstice Backup™ client software, Version 5.1.Build.103 to the mini-root. All client software necessary to recover files from a specified Solstice Backup storage server is installed in the appropriate locations of the mini-root. No man pages or storage server software are included.

The MRsbu package is available for version 2.6 of the Solaris OE.

Using MR

The software packages provided by the MR system are, in most cases, used in precisely the same way from the fail safe booted client as it is from any other client. The various packages should be installed on the JumpStart server much as the original CD image was. When the JumpStart server image was copied, a base directory was specified to `setup_install_server`. That same location is given to MRbase during its installation so that it may modify the correct files. When installing the MRbase package (by executing the standard Solaris OE `pkgadd` command), the JumpStart server image location is prompted for. The location of the JumpStart server image will be stored in a state file, so that subsequent MR package installations may suggest an intelligent default for this location

A Sample Installation

The following is a sample installation of the MRbase and MRvx packages onto a Solaris 2.6 OE JumpStart server:

```
racerx# cd /var/tmp
racerx# ls *.pkg
MRbase.pkg MRvx.pkg
racerx# pkgadd -d ./MRbase.pkg all
Processing package instance <MRbase> from </var/tmp/MRbase.pkg>
Recovery Tools for Solaris_2.6 CD Image
(sparc,i386) 1.1.8
I need to know the base directory of your Solaris 2.6 CD image.
This is the same directory that you specified when you created the
install server using 'setup_install_server'.
What is the base directory for your CD image [?,q] /JumpStart/Solaris2.6_0599
## Executing checkinstall script.
Using </JumpStart/Solaris2.6_0599/Solaris_2.6/Tools/Boot> as the package base
directory.
## Processing package information.
## Processing system information.
5 package pathnames are already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.
This package contains scripts which will be executed with super-user
permission during the process of installing this package.
Do you want to continue with the installation of <MRbase> [y,n,?] y
Installing Recovery Tools for Solaris_2.6 CD Image as <MRbase>
## Installing part 1 of 1.
/JumpStart/Solaris2.6_0599/Solaris_2.6/Tools/Boot/kernel/drv/fddi
/JumpStart/Solaris2.6_0599/Solaris_2.6/Tools/Boot/kernel/drv/nf
[ verifying class <none> ]
Modifying /etc/.MR_BASEDIR_2.6
[ verifying class <build> ]
Modifying /JumpStart/Solaris2.6_0599/Solaris_2.6/Tools/Boot/sbin/rcS
Modifying /JumpStart/Solaris2.6_0599/Solaris_2.6/Tools/Boot/sbin/sysconfig
[ verifying class <sed> ]
## Executing postinstall script.
Reboot client to install driver.
Note: major number maximum based on server, not client
Installation of <MRbase> was successful.
```

```
racerx# pkgadd -d ./MRvx.pkg all
Processing package instance <MRvx> from </var/tmp/MRvx.pkg>
VxVM Drivers for Solaris_2.6 CD Image
(sparc,i386) 1.1.1
I need to know the base directory of your Solaris 2.6 CD image.
This is the same directory that you specified when you created the
install server using 'setup_install_server'.
What is the base directory for your CD image?
Default=/JumpStart/Solaris2.6_0599/Solaris_2.6 [?,q]
## Executing checkinstall script.
Using </JumpStart/Solaris2.6_0599/Solaris_2.6/Tools/Boot> as the package base
directory.
## Processing package information.
## Processing system information.
7 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.
The following files are being installed with setuid and/or setgid permissions:
/JumpStart/Solaris2.6_0599/Solaris_2.6/Tools/Boot/usr/sbin/vxprint <setuid
root>
Do you want to install these as setuid/setgid files [y,n,?,q] y
This package contains scripts which will be executed with super-user
permission during the process of installing this package.
Do you want to continue with the installation of <MRvx> [y,n,?] y
Installing VxVM Drivers for Solaris_2.6 CD Image as <MRvx>
## Installing part 1 of 1.
...Long list of files installed ...
[ verifying class <none> ]
Modifying /JumpStart/Solaris2.6_0599/Solaris_2.6/Tools/Boot/etc/devlink.tab
Modifying /JumpStart/Solaris2.6_0599/Solaris_2.6/Tools/Boot/etc/system
[ verifying class <sed> ]
## Executing postinstall script.
Reboot client to install driver.
Note: major number maximum based on server, not client
Installation of <MRvx> was successful.
```

After all components are installed, add an install client by using the `add_install_client` command, as would be done to configure a standard JumpStart client (specifics on the use of `add_install_client` may be found in the References sections). The client is then booted with the recover keyword, invoking the MR augmentations. The following is a sample boot with the MR augmented mini-root:

```
{0} ok boot net - recover
Boot device: /sbus@3,0/SUNW,hme@3,8c00000  File and args: - recover
2ec00
hostname: blackmesa
domainname: EE_Lab
root server: racerx
root directory: /export/Install/2.6/Solaris_2.6_0599/Tools/Boot
SunOS Release 5.6 Version Generic [UNIX(R) System V Release 4.0]
Copyright (c) 1983-1997, Sun Microsystems, Inc.
socal1: port 0: Fibre Channel Loop is ONLINE
socal0: port 0: Fibre Channel Loop is ONLINE
Configuring devices...
The system is coming up.  Please wait.
Starting remote procedure call (RPC) services: sysidnis done.
Launching K shell...
#
# /sbin/vxstartup.sh
Enter the VxVM host ID: [blackmesa]
   If you do not have a photon or SSA on this host, you will
   need a temporary license.

Do you need a license key? [yes] yes
Please enter your key: xxxx xxxx xxxx xxxx xxxx xxx
vxvm:vxserial: INFO: Feature name: CURRSET [95]
vxvm:vxserial: INFO: Number of licenses: 1 (non-floating)
vxvm:vxserial: INFO: Expiration date: Sun Aug 01 03:00:00 1999 (35.7 days
from now)
vxvm:vxserial: INFO: Release Level: 20
vxvm:vxserial: INFO: Machine Class: All
vxvm:vxserial: INFO: Key successfully installed in /etc/vx/elm/95.
```

```
Do you need another license key? [no] no
Starting up VxVM . . . . . done
Do you wish to start volumes in the rootdg disk group? [yes] yes
dg rootdg usetype root: start rootvol
dg rootdg usetype swap: start swapvol
Do you wish to start volumes in all other disk groups? [no] no
You can start all volumes in all known disk groups using the command:
    vxrecover -svn
# fsck /dev/vx/rdsk/rootdg/rootvol
** /dev/vx/rdsk/rootdg/rootvol
** Last Mounted on /
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
28054 files, 449629 used, 661858 free (1226 frags, 82579 blocks,  0.1%
fragmentation)
# mount /dev/vx/dsk/rootdg/rootvol /a
```

Conclusion

This article is an introduction to the MR system for rapid recovery. An overview of the MR architecture has been presented as well as an examination of several of the more commonly used MR packages.

By implementing MR on your JumpStart servers it may be possible to reduce the number of reboots required during a system recovery or service event. This minimization of reboots will speed recovery and service time as well as help enable the system administrator to use datacenter tools during system recovery procedures.

Acknowledgements

MR began as the work of Gene Trantham, Gene was responsible for the initial architecture of MR as well as being the motivating force behind MR. Chris Elfers was responsible for the Solstice DiskSuite versions of MR as well as automating the packaging of the MR components.

References

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

Howard, John S., *JumpStart Mechanics: Using JumpStart Application for Hands-Free Installation of Unbundled Software - Part 1*, Sun Blueprints OnLine, May 2000.
`http://www.sun.com/blueprints/0500/jsmech1.pdf`

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 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, the storage subsystem and the Solaris kernel. In addition to these support functions, he developed and performed Reliability, Accessibility, and Serviceability (RAS) studies of customer datacenters.