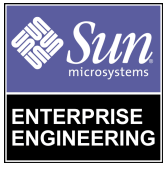
Rapid Recovery Techniques: Exploring the Solaris™ Software Registry

---

By Richard Elling - Enterprise Engineering

Sun BluePrints™ OnLine - October 1999



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.: 806-3753-10
Revision 01, October 1999

Please
Recycle

Rapid Recovery Techniques: Exploring the Solaris™ Software Registry

System recovery time, also described as mean time to recover (MTTR), is an important part of availability calculations. Reducing recovery time directly and positively impacts the overall availability of systems.

This is the first article in a series that will discuss rapid recovery techniques for the Solaris operating environment. The focus of this series is to show ways in which recovery time for repairing or restoring systems can be decreased thus increasing overall availability. This series will also deal with recovery techniques for problems specifically not related to failed hardware. Computer industry analysts observe failures of three main types: product, people, and process. In particular, this series will discuss how to use processes to recover from errors caused by people.

The Software Registry

The Solaris operating environment maintains a software registry for maintaining information on software packages and files installed on the system. This registry plays a critical role in software installation, upgrades, and patches. The registry is the primary location for software installation information.

The software and patch installation information is kept under the system administration directory, `/var/sadm`. The layout of the `/var/sadm` directory structure changed in Solaris 7 operating environment. Table 1 shows the `/var/sadm` directory contents.

Table 1: `/var/sadm` Directory Contents.

Description	Pre-Solaris 7 Location	Solaris 7 Location
Log files from installation and system configuration software	`.`	`system/logs/`
Data files generated by system installation and configuration utilities containing system-relevant configuration data	`.`	`system/data/`
Database of installed files	`install/contents`	`install/contents`
Directories for storing patched files and patch information	`patch/`	`patch/`
Directories for storing software package information	`pkg/`	`pkg/`
Log of Solaris operating environment installation including file system creation, Solaris operating environment package installations, and file system information	`install_data/install_log`	`system/logs/install_log`
Log of Solaris operating environment upgrade messages	`install_data/upgrade_log`	`system/logs/upgrade_log`
Log of Solaris operating environment post upgrade cleanup messages	`install_data/ upgrade_cleanup`	`system/data/ upgrade_cleanup`
File system size information from upgrade	`install_data/ upgrade_space_report`	`system/data/ upgrade_space_report`
Begin script output, usually empty unless customized for AutoInstall	`begin.log`	`system/logs/begin.log`

Table 1: `/var/sadm` Directory Contents.

Description	Pre-Solaris 7 Location	Solaris 7 Location
Finish script output, usually empty unless customized for AutoInstall	`finish.log`	`system/logs/finish.log`

The software registry is a key component of the Solaris operating environment because it contains information on what software is installed and where. This information is vital for recovering from errors that involve changes in the software or files on the system. The registry is also invaluable for auditing the system to determine what software has been installed, removed, or patched.

The software registry is located in the `/var` file system. `/var` may vary widely in its disk space consumption because it contains the `/var/tmp` directory where users can place temporary files, as well as `/var/adm` and `/var/log`, which collect accounting and logging information. The software registry requires space to grow as software is installed. Any patches that are installed without explicitly using the “do not backup the files” option will store the files needed for back out in the `/var/sadm/pkg` directory. Enough space should be available in the `/var` directory to hold user temporary files plus software installation requirements and patch management. For a large server, the software registry may require 50MB or more and will grow over time as patches are added.

Software for Solaris operating systems is typically installed as packages. A package contains files, installation scripts, copyright information, and removal scripts. Refer to the Solaris *Application Packaging Developer’s Guide* for a complete description of packages, package construction, and management. This guide is available in the Software Developer Collection in the Answerbook™ documentation and on the internet at `http://docs.sun.com`

Occasionally packages that are related in some way are collected into clusters. Clusters aid in systems management as it is easier to install a single cluster than the dozens of individual packages. Solaris clusters are named `SUNWC`*`name`*.

Solaris patches are distributed as sparse packages. A patch usually contains fewer files than a complete package. A patch will not normally include pre installation scripts.

Patches are not required to update packages. A complete package may be built that includes a few updated files. At installation time, only the files that have been changed are installed. Files which remain the same are not reinstalled. The software registry is used to manage the changes needed. The ability to update files by installing a later version of a package is a very useful feature that will be explored in detail in articles later in this series.

Tools for Administering the Software Registry

Starting with the Solaris 2.0 release, the Solaris operating environment has included tools for manipulating the software registry. These are: `installf`, `removef`, `pkgadd`, `pkgrm`, `pkginfo`, and `pkgchk`. These commands are intended for system administrators and software developers.

The Solaris 2.3 release added the software management tool, `swmtool`. `swmtool` has an X-window GUI that can be used to add and remove packages. The back end to `swmtool` is the normal `pkgadd`, `pkgrm`, and `pkginfo`. All interaction with the `pkg` commands is done in a text window. This function was later added to the `admintool` under the browse software menu.

The Solaris 2.6 release added the `patchadd` and `patchrm` command. Until the Solaris 2.6 release, each patch was bundled with a script, `installpatch`, to install the patch and manage the registry. A back out script, `backoutpatch`, included with each patch is now replaced by `patchrm`. However, since the scripts were built on the shell `pkg` commands, the process of adding patches was slow. For a large server with lots of software installed, the patch process was painfully slow. The `patchadd` program is a binary executable and manages the registry through a C API. The result is a more efficient patch process.

The Solaris 2.6 release added a new software installation procedure based on technology from InstallShield. InstallShield is the most popular interface for installing software in the Microsoft Windows environments. While the InstallShield front end looks very Windows-like, the back end is still the familiar `pkgadd` command. Hence, the registry remains the same.

The Solaris 7 5/99 release added a new Java™ technology-based application, `prodreg`. `prodreg` is a GUI interface for manipulating packages and clusters of packages. `prodreg` is an updated version of `admintool` using Java Abstract Windowing Toolkit (AWT) or SWING GUI interface instead of Motif.

The Database of Installed Files

Every file installed on a Solaris operating system using the pkgadd command has an entry in the database of installed files, `/var/sadm/install/contents`. The `contents` is a text file that contains one line per installed file. Information about each file includes:

- full pathname where the file is installed
- type of file: normal file, editable, volatile, directory, link, raw, cooked, etc.
- a secondary descriptor field, as needed
- file permissions (mode) in octal (see `chmod(1)`)
- owner id
- group id
- 16-bit checksum from the `sum` command
- size in bytes
- package name(s) which installed the file

Using `pkgchk` to Provide Information about Specific Files

The `pkgchk` command can provide a verbose description of the information in the `contents` file. For example:

```
$ pkgchk -l -p /usr/bin/ls
Pathname: /usr/bin/ls
Type: regular file
Expected mode: 0555
Expected owner: bin
Expected group: bin
Expected file size (bytes): 18120
Expected sum(1) of contents: 53113
Expected last modification: Oct 06 00:43:05 1998
Referenced by the following packages:
    SUNWcsu
Current status: installed
```

```
$ pkgchk -l -p /etc/hosts
Pathname: /etc/hosts
Type: symbolic link
Source of link: ./inet/hosts
Referenced by the following packages:
    SUNWcsr
Current status: installed
```

The equivalent information can be retrieved in raw form using grep. For example:

```
$ grep /usr/bin/ls /var/sadm/install/contents
/usr/bin/ls f none 0555 bin bin 18120 53113 907659785 SUNWcsu
```

All updates to the `contents` file must be carefully controlled to avoid corruption of the database. The Solaris operating environment commands `installf` and `removef` are used to help ensure controlled updating of the `contents` file and proper format of the database entries. Commands that read the database, such as `pkgchk`, properly follow the locking semantics enforced by `installf` and `removef`. The `contents` file is a text file, so Solaris operating environment commands that process text files will work as expected. However, they do not recognize the locking mechanism used to avoid corruption of the database. There is no mechanism to repair a corrupted `contents` file other than to restore from a backup. Use only the Solaris `pkg` commands to interface with the `contents` file.

`grep` can be useful for locating the locations of installed files. For example, if there is a file called "`pkgchk`" installed somewhere on the system, it can be located easily without using `find`:

```
$ grep pkgchk /var/sadm/install/contents
/usr/sbin/pkgchk f none 0555 root sys 157836 25516 907660189
SUNWcsu
/usr/share/man/sman1m/pkgchk.1m f none 0444 bin bin 12016 56338
904250063 SUNWman
```

A more readable example may be:

```
$ grep pkgchk /var/sadm/install/contents | awk '{print $1}'
/usr/sbin/pkgchk
/usr/share/man/sman1m/pkgchk.1m
```

In these examples, the entire pathname for the installed file and the manual page are found.

Using `pkginfo` to Provide Information about Specific Packages

The `pkginfo` command can provide information about specific packages.This information is useful for discovering the packages that are installed and detailed information about them. For example:

```
$ pkginfo
systemSUNWcsrCore Solaris, (Root)
systemSUNWcsuCore Solaris, (Usr)
...

$ pkginfo -l SUNWcsr
 PKGINST:  SUNWcsr
    NAME:  Core Solaris, (Root)
CATEGORY:  system
    ARCH:  sparc
 VERSION:  11.7.0,REV=1998.09.01.04.16
 BASEDIR:  /
  VENDOR:  Sun Microsystems, Inc.
    DESC:  core software for a specific instruction-set
architecture
  PSTAMP:  on998-patch19981105094318
INSTDATE:  Jul 08 1999 17:36
 HOTLINE:  Please contact your local service provider
  STATUS:  completely installed
   FILES:    511 installed pathnames
              38 shared pathnames
              64 linked files
              79 directories
             168 executables
           23482 blocks used (approx)
```

Using `prodreg` to Manage Packages

Figure 1 shows the `prodreg` main screen. `prodreg` allows easy exploration of the installed clusters and packages.

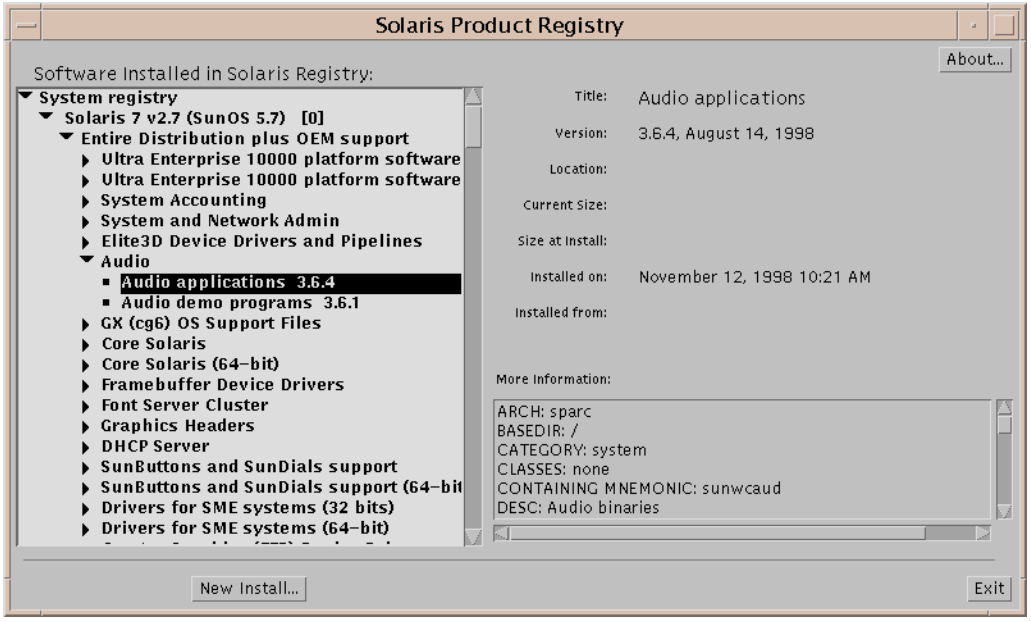


FIGURE 1 `prodreg` Sample Display

In this example, software clusters are shown in the software registry window with triangle icons. Individual packages are shown with square icons. Cluster and package details are immediately visible on the right side when selected.

Using *`admintool`* to Manage Packages

Figure 2 shows the `admintool` browsing screen. `admintool` allows easy exploration of the installed clusters and packages.

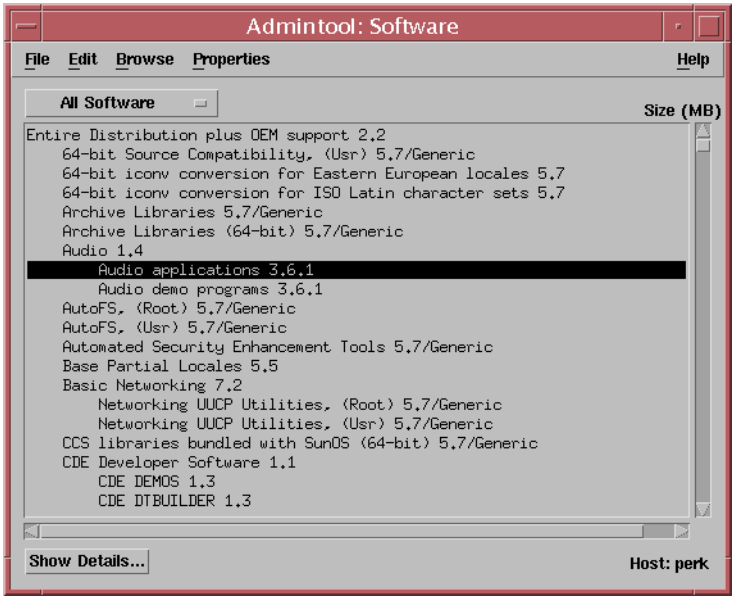


FIGURE 2 `admintool` Sample Display

However, clusters are not as obvious in `admintool` as they are in `prodreg`. Individual packages are the right most indented items. It is difficult in `admintool` to determine at a glance whether an item is a cluster or a package. You must select an item and press the *`Show Details...`* button to see any significant detail about the cluster or package.

Conclusion

This article describes the Solaris Operating Environment software registry. The registry contains records and logs of all software packages installed and patched on a Solaris operating system. This information is important for maintaining the software configuration of a Solaris operating system. Techniques for using the registry to rapidly recover a system will be discussed in a future Sun BluePrints™ OnLine article.

Author's Bio: Richard Elling

Richard is a Senior Engineer in Enterprise Engineering for the Computer Systems at Sun Microsystems in San Diego, California. Richard had been a field systems engineer at Sun for five years. Richard was the Sun Worldwide Field Systems Engineer of the year in 1996. Prior to Sun, Richard was the Manager of Network Support for the College of Engineering at Auburn University, a design engineer for a startup microelectronics company, and worked for NASA doing electronic design and experiments integration for Space Shuttle missions