solaris

EASIER SYSTEM ADMINISTRATION with SERVICE MANAGEMENT FACILITY

> **Solaris™ 10** How To Guides

Jeff Victor, Solaris Marketing

Sun microsystems

About This Solaris How To Guide

This How to Guide instructs system administrators unfamiliar with the Solaris 10 OS on how to use the Service Management Facility (SMF) to monitor and manage a Solaris 10 system. The guide starts with a brief overview of the Service Management Facility and follows with several examples of using the facility to manage services. With SMF, system administration becomes easier, faster, and more reliable—enabling both novice and experienced Solaris administrators to quickly and efficiently manage systems and their associated services.

After reading this guide, the user will be able to use SMF to obtain information about system services and be able to apply this knowledge to perform similar tasks for other services on Solaris 10 systems.

Contents

Service Management Facility How To Guide

Service Management Facility: Overview

The Service Management Facility (SMF) is a core component of the new Predictive Self-Healing set of technologies introduced in Solaris 10. With SMF, system administrators can use simple command line utilities to easily identify, observe, and manage both the services provided by the system and the system itself.

A Solaris service is any long-lived software object with a well-defined state, start and stop, and relationship to other services on the system. Delivering email, handling ftp requests, and permitting remote command execution are a few examples of services typically provided within the Solaris environment.

In Solaris 10, each software service has an advertised state. Should a failure occur, the system automatically diagnoses it and locates/pinpoints the source of the failure. Failing services are automatically restarted whenever possible, reducing the need for human intervention. Should manual intervention be required, system administrators can quickly identify the root cause of the service's failure and significantly reduce the times-to-repair and recover from said failure.

Specifically, SMF enables administrators to do the following tasks easily and efficiently:

- Observe and manage system-wide services
- Identify "misbehaved" or failed services
- Securely delegate administrative tasks to non-root users
- Automatically restart failed services in the appropriate order of dependency
- Persist the enable/disable of services across system upgrades & patches
- Preserve compatibility with legacy services
- Automatically configure snapshots for backup, restore, undo
- Provide consistent configuration handling

SMF preserves compatibility with "legacy" services. Legacy refers to /etc/rc*.d, /etc/init.d, and /etc/inittab scripts which have been used to manage ISV-provided or internally developed services. Legacy services will continue to work as they did in earlier releases of Solaris, and you will be able to observe these services with SMF. However, they will not participate in or benefit from SMF's self-healing capabilities, such as service restart, until the scripts have been converted to SMF manifests.

Service Manifests

The above tasks are made possible by a key attribute of SMF: it understands the relationships and dependencies between software services on a Solaris system.

This information is stored in a *service manifest* which SMF uses when managing services as well as when determining root causes of service failures. The service manifest also describes the conditions under which failed services may be automatically restarted. A separate service manifest is required per service/application. Sun provides some service manifests by default. Optionally, you can customize these manifests, or write your own for other services.

SMF Commands

SMF has a limited yet powerful set of commands. Each command has several options which cover the tasks required to manage Solaris systems. The following table lists the SMF commands.

Command	Description
svcs	Reports service status
svcadm	Used for service management: e.g., starting, stopping and restoring services
svccfg	Used to list properties of a service
svcprop	Used to list properties of a service
inetadm	Used to manage inetd services

Table 1—SMF commands

This guide focuses on gathering information about the services running on a Solaris system and troubleshooting a failed service. These tasks are accomplished with the svcs and svcadm commands.

The other commands enable system administrators to manage, modify, and display service manifests. You can read more about them on docs.sun.com under the Solaris 10 Basic Administration Guide.

SMF: An Example

This example begins by looking at all the services currently running on your Solaris system and then examining a few of the services for more details. These details include the services upon which they depend and the services which depend upon them.

Displaying System Services Information

1. To display all services on your Solaris system with their state information, use the svcs command along with the -a option:

```
my-system# svcs -a
STATE          STIME    FMRI
legacy_run     Apr_18   lrc:/etc/rcS_d/S51installupdates
legacy_run     Apr_18   lrc:/etc/rc2_d/S47pppd
legacy_run     Apr_18   lrc:/etc/rc2_d/S99audit
legacy_run     Apr_18   lrc:/etc/rc3_d/S76snmpdx
legacy_run     Apr_18   lrc:/etc/rc3_d/S90samba
disabled       Apr_18   svc:/network/ipfilter:default
disabled       Apr_18   svc:/network/rpc/keyserv:default
disabled       Apr_18   svc:/network/rpc/nisplus:default
disabled       Apr_18   svc:/application/print/server:default
disabled       Apr_18   svc:/network/dhcp-server:default
disabled       Apr_18   svc:/network/http:apache2
online         Apr_18   svc:/system/svc/restarter:default
online         Apr_18   svc:/network/pfil:default
online         Apr_18   svc:/network/physical:default
online         Apr_18   svc:/system/identity:domain
online         Apr_18   svc:/system/cryptosvc:default
```

```
online          Apr_18    svc:/network/inetd:default
online          Apr_18    svc:/network/telnet:default
online          Apr_18    svc:/network/ssh:default
online          Apr_18    svc:/system/zones:default
online          Apr_18    svc:/network/nfs/nlockmgr:default
offline         Apr_18    svc:/application/print/ipp-listener:default
offline         Apr_18    svc:/application/print/rfc1179:default
[Note: This is a truncated list]
```

Displaying Individual Services

You can look at individual services as well. This is especially useful during troubleshooting or when examining what is going on with a particular service.

1. To display information about the inetd service, use the svcs command specifying the service by name:

```
my-system# svcs inetd
STATE           STIME     FMRI
online          Apr_18    svc:/network/inetd:default
```

2. To display information about the Samba service, use the svcs command specifying the service by name:

```
my-system# svcs S90samba
STATE           STIME     FMRI
legacy_run      Apr_18    lrc:/etc/rc3_d/S90samba
```

3. To display information about the Apache service, use the svcs command specifying the service by name:

```
my-system# svcs apache2
STATE           STIME     FMRI
disabled        Apr_18    svc:/network/http:apache2
```

Retrieving Dependency Tree Information

SMF permits you to identify all the service dependencies for a given service. That is, the services upon which a given service depends, as well as the services that depend upon that service. The following options are used to provide additional detail on the services.

Option	Description
-a	Displays all services, including those which have been disabled
-d	Lists a service's dependencies
-D	Lists a service's dependents
-l	Displays all available information about the service
-p	Lists all processes (PID) associated with a service

Table 2—Useful options for svcs(1)

Note that a lowercase -d option and the uppercase -D option actually mean different things. The -d option results in a list of services on which the named service depends, while the -D option results in a list of services which depend upon the named service. Think of them as above and below the service on a dependency tree.

In Step 1, you used the -a option to list all services on the system. Now take a look at the list of services on which inetd depends.

1. To list the dependencies of the inetd service, use the svcs command, specifying the -d option:

```
my-system# svcs -d inetd
STATE          STIME    FMRI
disabled       Apr_18   svc:/network/inetd-upgrade:default
online         Apr_18   svc:/milestone/name-services:default
online         Apr_18   svc:/network/loopback:default
online         Apr_18   svc:/milestone/network:default
online         Apr_18   svc:/system/filesystem/local:default
online         Apr_18   svc:/network/rpc/bind:default
online         Apr_18   svc:/milestone/sysconfig:default
```

As you can see, inetd depends upon a number of different services including inetd-upgrade and name-services. The same command can be used to find out if any one of these services depends upon other services. With the information gathered you can sketch out the dependency tree for inetd. Figure 1 shows a partial graph of the dependency tree for inetd.

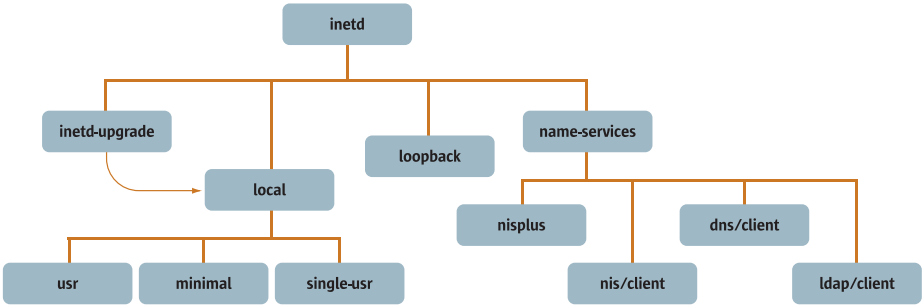

Figure 1—Partial dependency tree for inetd

2. To generate a similar list for Apache2, use the svcs command and the -d option, specifying Apache2 by name:

```
my-system# svcs -d apache2
STATE          STIME    FMRI
online         Apr_18   svc:/network/loopback:default
online         Apr_18   svc:/network/physical:default
```

Next, use the -D option to identify services which depend upon inetd and Apache.

3. To discover the services which depend upon inetd, use the svcs command with the -D option, specifying inetd by name:

```
my-system# svcs -D inetd
STATE          STIME    FMRI
online         Apr_18   svc:/milestone/multi-user:default
```

In this example you can see that multi-user depends upon inetd.

Next, find the services which, in turn, depend upon multi-user.

4. To discover the services which depend upon multi-user, use the svcs command with the -D option, specifying multiuser by name:

```
my-system# svcs -D multi-user
STATE           STIME     FMRI
disabled        Apr_18    svc:/network/dhcp-server:default
online          Apr_18    svc:/milestone/multi-user-server:default
```

Notice that there are two services which depend upon multi-user, dhcp-server and multi-user-server.

Next, examine the dhcp-server.

5. To discover the services which depend upon dhcp-server, use the svcs command with the -D option, specifying multiuser by name. Follow through the whole dependency tree in the same way:

```
my-system# svcs -D dhcp-server
STATE           STIME     FMRI
online          Apr_18    svc:/milestone/multi-user-server:default
```

Find the services which depend upon multi-user-server:

```
my-system# svcs -D multi-user-server
STATE           STIME     FMRI
online          Apr_18    svc:/system/zones:default
```

Find services which depend upon zones:

```
my-system# svcs -D zones
STATE           STIME     FMRI
```

In this case, there are no services which depend on zones, so this is the end of the dependency tree. See the dependency tree below.

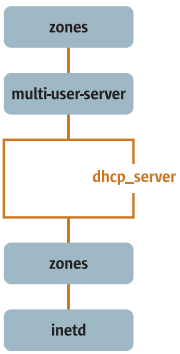

Figure 2—Tree listing the dependents of inetd

Listing Service Processes

Another common task is to list the processes comprising a service. In this example, we will look at the Apache2 service.

1. First, check if the service is running, using the svcs command:

```
my-system# svcs apache2
STATE          STIME    FMRI
online         11:25:33 svc:/network/http:apache2
```

2. To list the process IDs of this service, use the svcs command with the -p option.

```
my-system# svcs -p apache2
STATE          STIME    FMRI
online         11:25:33 svc:/network/http:apache2
                11:25:33     2438 httpd
                11:25:34     2439 httpd
                11:25:34     2440 httpd
                11:25:34     2441 httpd
                11:25:34     2442 httpd
                11:25:34     2443 httpd
```

Viewing all Service Information

1. To view all available information about a particular service, use the svcs command with the -l option, specifying the service by name:

```
my-system# svcs -l apache2
fmri          svc:/network/http:apache2
name          Apache 2 HTTP server
enabled       true
state         online
next_state    none
state_time    Wed Jun 07 11:47:04 2006
logfile       /var/svc/log/network-http:apache2.log
restarter     svc:/system/svc/restarter:default
contract_id   297
dependency    require_all/error svc:/network/loopback:default (online)
dependency    optional_all/error svc:/network/physical:default (online)
```

This option gives a number of useful details about the service. For example, the service is enabled, is online, and has been online since June 7 11:47:04 2006. The logfile, the restarter, and the service dependencies are given as well.

Common Administrative Tasks

This examples starts the Apache2 service and performs some common administrative tasks on the service.

1. To start the apache2 service, use the svcadm command with the enable option:

```
my-system# svcadm enable apache2
```

2. To display its status, use the svcs command:

```
my-system# svcs http
STATE          STIME    FMRI
online         11:26:46 svc:/network/http:apache2
```

3. To examine the process IDs associated with the service, use the scvs command with the -p option:

```
my-system# svcs -p http
STATE          STIME    FMRI
online         11:26:46 svc:/network/http:apache2
                11:26:46     2463 httpd
                11:26:47     2464 httpd
                11:26:47     2465 httpd
                11:26:47     2466 httpd
                11:26:47     2467 httpd
                11:26:47     2468 httpd
```

4. To kill the service, use the pkill command. Then check the status again:

```
my-system# pkill http

my-system# svcs http
STATE          STIME    FMRI
online         11:28:05 svc:/network/http:apache2
```

Note that in this example the service did indeed stop, but was restarted automatically. STIME in the two cases are different indicating that the service was restarted. SMF increases the uptime of the service and also makes this information easy to retrieve.

5. Get more information using the -x option.

```
my-system# svcs -x http
svc:/network/http:apache2 (Apache 2 HTTP server)
 State: online since Tue Jun 06 11:28:05 2006
   See: apache2(1M)
   See: /var/svc/log/network-http:apache2.log
Impact: None.
```

With SMF it is easy to get additional information about services using the -x and -v options of the svcs(1) command. This is particularly helpful when you are investigating the reason why a particular service has failed.

Option	Description
-x	Displays explanations for service states
-v	With -x, displays extra information for each explanation

Table 3—Additional options for svcs(1)

Determining System Faults

In this example you will investigate an Apache service failure.

Note: This is a contrived example because with SMF it is difficult to cause a failure that lasts long enough to warrant investigation. As seen in the previous example, the restarter starts Apache almost immediately and does not give us time to go through this exercise. For the purposes of this guide we have forced a failure of the Apache2 service. Details on creating this failure are included in the next section.

1. To examine the current state of the Apache2 service, use the svcs command:

```
my-system# svcs apache2
STATE           STIME    FMRI
maintenance    11:38:37 svc:/network/http:apache2
```

   Notice that the state is “maintenance” not online.

2. To examine the current state of the Apache2 service, use the svcs command:

```
my-system# svcs -xv apache2
svc:/network/http:apache2 (Apache 2 HTTP server)
 State: maintenance since Tue Jun 06 11:38:37 2006
Reason: Method failed.
   See: http://sun.com/msg/SMF-8000-8Q
   See: man -M /usr/share/man -s 1M apache2
   See: /var/svc/log/network-http:apache2.log
Impact: This service is not running.
```

   Notice that the service is not running and that a message ID and URL are given to learn more about the failure. You also can check the log file.

3. To examine the log file, use the tail command with the -3 option to print the last 3 lines of the file:

```
my-system# tail -3 /var/svc/log/network-http:apache2.log
[ Jun  6 11:38:37 Stopping because all processes in service exited. ]
[ Jun  6 11:38:37 Executing stop method ("/lib/svc/method/http-apache2
stop") ]
[ Jun  6 11:38:37 Method "stop" exited with status 96 ]
```

   Notice there are no processes associated with the Apache2 service. The following example shows the expected output when a process is attached to the service:

```
my-system# svcs -p apache2
STATE             STIME    FMRI
maintenance    17:57:36 svc:/network/http:apache2
```

If the service were working and healthy the output would be:

```
my-system# svcs -p apache2
STATE          STIME    FMRI
online         Jun_07   svc:/network/http:apache2
                Jun_07       2880 httpd
                Jun_07       2881 httpd
                Jun_07       2882 httpd
                Jun_07       2883 httpd
                Jun_07       2884 httpd
                Jun_07       2885 httpd
my-system#
```

4. Look up the message ID (the highlighted line under step 2 above):

```
http://sun.com/msg/SMF-8000-8Q
```

When you look up the message ID on the URL provided, you will learn that this failure could be due to either a missing or broken file.

In our example, we look in the directory for the Apache file and find that indeed it is missing. We then replace it. In this example, we simulated a lost file by changing the name of the file, so we change it back using the cp command:

```
my-system# cp /etc/apache2/httpd.conf-example /etc/apache2/httpd.conf
```

5. To re-check the service status, use the svcs command:

```
my-system# svcs apache2
STATE          STIME    FMRI
maintenance    11:38:37 svc:/network/http:apache2
```

The service is still in maintenance.

6. To restore the service, use the svcadm command, with the clear option:

```
my-system# svcadm clear apache2
```

7. To confirm it is back online, once again use the svcs command:

```
my-system# svcs http
STATE          STIME    FMRI
online         11:40:45 svc:/network/http:apache2
```

8. Get more details using the -xv option of the svcs command:

```
my-system# svcs -xv apache2
svc:/network/http:apache2 (Apache 2 HTTP server)
 State: online since Tue Jun 06 11:40:45 2006
   See: man -M /usr/share/man -s 1M apache2
   See: /var/svc/log/network-http:apache2.log
Impact: None.
```

You can see that the Apache service has been online since Jun 06 11:40:45 2006. By looking the time provided by the State line, you can determine whether a service has been restarted. In our example, we know the service has been restarted. On a production system this type of information can be very useful.

Failing the Apache Service

A failure of the Apache service can be induced using the following commands.

1. To check that the service is running, use the svcs command:

```
my-system# svcs apache2
STATE          STIME    FMRI
disabled       11:37:45 svc:/network/http:apache2
```

   Apache is disabled.

2. To enable Apache2, use the svcadm command with the enable option:

```
my-system# svcadm enable apache2
```

3. To check that the service was enabled, use the svcs command again:

```
my-system# svcs apache2
STATE          STIME    FMRI
online         11:38:15 svc:/network/http:apache2
```

4. To induce failure, first remove the file using the rm command and then kill the service using the pkill command:

```
my-system# rm /etc/apache2/httpd.conf

my-system# pkill http
```

   The service is now stopped and will not be automatically restarted since the config file has been removed. We already saved a copy of the file under a different name for recovery.

For More information

This guide is just a brief introduction to SMF. For more information on SMF and Predictive Self-Healing, please visit the following websites:

Web Resources	
Solaris Website	sun.com/solaris/
SMF Community page	opensolaris.org/os/community/smf/
PSH (SMF/FMA) Bigadmin page	sun.com/bigadmin/content/selfheal/
SMF Quickstart	sun.com/bigadmin/content/selfheal/smf-quickstart.html
SMF Service Developer Introduction	sun.com/bigadmin/content/selfheal/sdev_intro.html
System Administration Guide: Managing Services	docs.sun.com/app/docs/doc/817-1985/6mhm8o5rl?a=view
Peter Baer Galvin's SMF article in Sys Admin	samag.com/documents/s=9766/sam0506i/0506i.htm
Blueprint: Service Management Facility (SMF) in the Solaris 10 Operating System	sun.com/blueprints/0206/819-5150.pdf

sun.com/solaris

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA Phone 1-650-960-1300 or 1-800-555-9SUN Web sun.com