



An Oracle Technical White Paper  
July 2012

# Easier System Administration with the Oracle Solaris Service Management Facility

# Introduction

This white paper instructs system administrators unfamiliar with the Oracle Solaris 11 operating system on how to use the Service Management Facility (SMF) to monitor and manage an Oracle Solaris 11 system. The document starts with a brief overview of SMF 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 Oracle Solaris administrators to quickly and efficiently manage systems and their associated services.

After reading this paper, 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 Oracle Solaris 11 systems.

## Service Management Facility Overview

The Service Management Facility (SMF) is a core component of the new Oracle Solaris Predictive Self-Healing set of technologies first introduced in Oracle 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.

An Oracle 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 e-mail, handling FTP requests, and permitting remote command execution are a few examples of services typically provided within the Oracle Solaris environment.

In Oracle Solaris 11, each software service has an advertised state. If a failure occurs, the system automatically diagnoses it and pinpoints the source of the failure. Failing services are automatically restarted whenever possible, reducing the need for human intervention. If manual intervention is required, system administrators can quickly identify the root cause of the service's failure and significantly reduce the time required to repair and recover from the 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
- Enable and disable services persistently across system upgrades and patches
- Preserve compatibility with legacy services
- Automatically configure snapshots for backup, restore, and undo
- Provide consistent configuration handling

SMF preserves compatibility with legacy services. Legacy refers to `/etc/rc*.d`, `/etc/init.d`, and `/etc/inittab` scripts that have been used to manage ISV-provided or internally developed services. Legacy services will continue to work as they did in earlier releases of Oracle 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 tasks described previously are made possible by a key attribute of SMF: It understands the relationships and dependencies among software services on an Oracle Solaris system. This information is stored in a service manifest that SMF uses when managing services and determining the root causes of service failures. The service manifest also describes the conditions under which failed services can be automatically restarted. A separate service manifest is required for each service or application. Oracle provides some service manifests by default. Optionally, you can customize these manifests, or you can write your own for other services.

## SMF Commands

SMF has a limited yet powerful set of commands. Each command has several options that cover the tasks required to manage Oracle Solaris systems. Table 1 lists the SMF commands.

TABLE 1. SMF COMMANDS

COMMAND	DESCRIPTION
<code>svcs</code>	Reports a service's status
<code>svcadm</code>	Used for service management, for example, starting, stopping, and restoring services
<code>svccfg</code>	Used to list and set properties of a service
<code>svcprop</code>	Used to list and set properties of a service

This guide focuses on gathering information about the services running on an Oracle 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 in the “Managing Service (Overview)” section of the [Oracle Solaris Administration: Common Tasks](#) guide.

## SMF: An Example

This example begins by looking at all the services currently running on your Oracle 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

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

```
my-system# svcs -a
STATE      STIME    FMRI
legacy_run Jun_14   lrc:/etc/rcS_d/S99openconnect-clean
legacy_run Jun_14   lrc:/etc/rc2_d/S47pppd
legacy_run Jun_14   lrc:/etc/rc2_d/S81dodatadm_udapl
legacy_run Jun_14   lrc:/etc/rc2_d/S89PRESERVE
disabled   Jun_14   svc:/platform/i86pc/acpihp:default
disabled   Jun_14   svc:/network/ipsec/policy:default
disabled   Jun_14   svc:/network/nis/domain:default
online     Jun_14   svc:/system/early-manifest-import:default
online     Jun_14   svc:/system/svc/restart:default
online     Jun_14   svc:/network/tcp/congestion-control:vegas
online     Jun_14   svc:/network/tcp/congestion-control:highspeed
online     Jun_14   svc:/network/sctp/congestion-control:highspeed
online     Jun_14   svc:/network/sctp/congestion-control:vegas
online     Jun_14   svc:/network/tcp/congestion-control:newreno
online     Jun_14   svc:/network/sctp/congestion-control:cubic
online     Jun_14   svc:/network/tcp/congestion-control:cubic
....
online     Jun_14   svc:/system/zones:default
online     Jun_14   svc:/system/power:default
online     Jun_14   svc:/system/hal:default
online     Jun_14   svc:/application/texinfo-update:default
online     Jun_14   svc:/application/pkg/update:default
```

## 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.

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

```
my-system# svcs inetd
STATE      STIME      FMRI
online     2:24:23  svc:/network/inetd:default
```

To display information about the SSH server service, use the **svcs** command specifying the service by name.

```
my-system# svcs ssh
STATE      STIME      FMRI
online     2:24:27  svc:/network/ssh:default
```

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

```
my-system# svcs apache22
STATE      STIME      FMRI
online     8:13:41  svc:/network/http:apache22
```

## 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.

**TABLE 2. USEFUL OPTIONS FOR SVCS(1)**

OPTION	DESCRIPTION
-a	Displays all services, including those that 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 process IDs (PIDs) associated with a service

Note that the lowercase **-d** option and the uppercase **-D** option do 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.

Previously, you saw how to use the **-a** option to list all the services on the system. Now let's look at the list of services on which **inetd** depends.

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   2:22:36 svc:/network/inetd-upgrade:default
online     2:23:02 svc:/network/loopback:default
online     2:23:35 svc:/milestone/network:default
online     2:23:38 svc:/system/filesystem/local:default
online     2:24:26 svc:/milestone/name-services:default
online     3:24:25 svc:/network/rpc/bind:default
```

As you can see, **inetd** depends upon a number of different services. The same command can be used to find out whether any one of these services depends upon other services.

To discover the services that depend upon **inetd**, use the **svcs** command with the **-D** option, specifying **inetd** by name. In this example, you can see that **multi-user** depends upon **inetd**.

```
my-system# svcs -D inetd
STATE      STIME      FMRI
online     2:24:40 svc:/milestone/multi-user:default
```

As before, the same command can be used to find out what services depend on **multi-user**. Using these two commands, we can easily map out a dependency tree of services, as required.

## Listing Service Processes

Another common task is to list the processes comprising a service. In this example, we will look at the **apache22** service.

First, check whether the service is running by using the **svcs** command.

```
my-system# svcs apache22
STATE      STIME      FMRI
online     8:13:41 svc:/network/http:apache22
```

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

```
my-system# svcs -p apache22
STATE      STIME     FMRI
online      8:13:41  svc:/network/http:apache22
              8:13:41    1660 httpd
              8:13:42    1662 httpd
              8:13:42    1663 httpd
              8:13:42    1664 httpd
              8:13:42    1665 httpd
              8:13:42    1666 httpd
```

## Viewing All Service Information

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 apache22
fmri        svc:/network/http:apache22
name        Apache 2.2 HTTP server
enabled     true
state       online
next_state  none
state_time  June 30, 2012 08:13:41 AM PDT
logfile     /var/svc/log/network-http:apache22.log
restarter   svc:/system/svc/restart:default
contract_id 133
manifest    /lib/svc/manifest/network/http-apache22.xml
dependency  require_all/error svc:/milestone/network:default (online)
dependency  require_all/none svc:/system/filesystem/local:default (online)
dependency  optional_all/error svc:/system/filesystem/autofs:default (online)
```

This option provides a number of useful details about the service. In this example, it reports that the service is enabled, is online, and has been online since June 30 08:13:41 2012. The log file, the rewriter, and the service dependencies are provided as well.

## Performing Common Administrative Tasks

The following example starts the **sendmail** service and performs some common administrative tasks on the service.

To display the status of the **sendmail** service, use the **svcs** command.

```
my-system# svcs sendmail
STATE      STIME      FMRI
disabled   2:28:08 svc:/network/smtp:sendmail
```

To enable the **sendmail** service, use the **svcadm** command with the **enable** option.

```
my-system# svcadm enable sendmail
my-system# svcs sendmail
STATE      STIME      FMRI
online    8:32:11 svc:/network/smtp:sendmail
```

To examine the process IDs associated with the service, use the **svcs** command with the **-p** option.

```
my-system# svcs -p sendmail
STATE      STIME      FMRI
online    8:32:11 svc:/network/smtp:sendmail
          8:32:11     1731 sendmail
```

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

```
my-system# pkill sendmail
my-system# svcs sendmail
STATE      STIME      FMRI
online    8:33:28 svc:/network/smtp:sendmail
```

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

Get more information using the **-x** option.

```
my-system# svcs -x sendmail
svc:/network/smtp:sendmail (sendmail SMTP mail transfer agent)
  State: online since June 30, 2012 08:33:28 AM PDT
  See: sendmail(1M)
  See: /var/svc/log/network-smtp:sendmail.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 (see Table 3). This is particularly helpful when you are investigating the reason why a particular service has failed.

**TABLE 3. ADDITIONAL OPTIONS FOR SVCS(1)**

OPTION	DESCRIPTION
<b>-x</b>	Displays explanations for service states
<b>-v</b>	When used with <b>-x</b> , displays extra information for each explanation

## 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 **apache22** service. Details on creating this failure are included in the next section.

To examine the current state of the **apache22** service, use the **svcs** command. Notice that the state is **maintenance** not **online**.

```
my-system# svcs apache22
STATE      STIME     FMRI
maintenance 8:37:03 svc:/network/http:apache22
```

To examine the current state of the **apache22** service, use the **svcs** command. Notice that the service is not running and that a message ID and URL are given so you can learn more about the failure. You also can check the log file.

```
my-system# svcs -xv apache22
svc:/network/http:apache22 (Apache 2.2 HTTP server)
  State: maintenance since June 30, 2012 08:37:03 AM PDT
  Reason: Method failed.

    See: http://sun.com/msg/SMF-8000-8Q
    See: man -M /usr/apache2/2.2/man -s 8 httpd
    See: http://httpd.apache.org
    See: /var/svc/log/network-http:apache22.log

  Impact: This service is not running.
```

To examine the log file, use the **tail** command.

```
my-system# tail /var/svc/log/network-http\:apache22.log
[ Jun 30 08:13:39 Executing start method ("/lib/svc/method/http-apache22
start"). ]
Apache version is 2.2
[ Jun 30 08:13:41 Method "start" exited with status 0. ]
[ Jun 30 08:37:03 Stopping because service restarting. ]
[ Jun 30 08:37:03 Executing stop method ("/lib/svc/method/http-apache22
stop"). ]
Apache version is 2.2
httpd: Could not open configuration file /etc/apache2/2.2/httpd.conf: No such
file or directory
Server failed to start. Check the error log (defaults to
/var/apache2/2.2/logs/error_log) for more information, if any.
[ Jun 30 08:37:03 Method "stop" exited with status 95. ]
```

In our example, we look in the directory for the Apache configuration 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/apache2/2.2/httpd.conf.bk /etc/apache2/2.2/httpd.conf
```

To recheck the service status, use the **svcs** command. The service is still in **maintenance** state. SMF does not dynamically check for the existence of the configuration file, so we will need to manually notify it that an administrator has manually corrected the error.

```
my-system# svcs apache22
STATE          STIME      FMRI
maintenance    8:37:03  svc:/network/http:apache22
```

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

```
my-system# svcadm clear apache22
```

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

```
my-system# svcs apache22
STATE          STIME      FMRI
online         8:48:31  svc:/network/http:apache22
```

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

```
my-system# svcs -xv apache22
svc:/network/http:apache22 (Apache 2.2 HTTP server)
  State: online since June 30, 2012 08:48:31 AM PDT
    See: man -M /usr/apache2/2.2/man -s 8 httpd
    See: http://httpd.apache.org
    See: /var/svc/log/network-http:apache22.log
  Impact: None.
```

You can see that the Apache service has been online since June 30 08:48:31 2012. By looking at 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.

## For More Information

This guide is just a brief introduction to the Oracle Solaris Service Management Facility. For more information on SMF, please visit the following Websites.

**TABLE 4. WEB RESOURCES**

Oracle Solaris 11 information on oracle.com	<a href="http://www.oracle.com/solaris">http://www.oracle.com/solaris</a>
Oracle Solaris 11 information on Oracle Technology Network	<a href="http://www.oracle.com/technetwork/server-storage/solaris11/overview/index.html">http://www.oracle.com/technetwork/server-storage/solaris11/overview/index.html</a>
SMF community page	<a href="http://hub.opensolaris.org/bin/view/Community+Group+smf/WebHome">http://hub.opensolaris.org/bin/view/Community+Group+smf/WebHome</a>
SMF technology page on Oracle Technology Network	<a href="http://www.oracle.com/technetwork/server-storage/solaris11/technologies/smf-1690889.html">http://www.oracle.com/technetwork/server-storage/solaris11/technologies/smf-1690889.html</a>
"Managing Services (Overview)" section of the <i>Oracle Solaris Administration: Common Tasks</i> guide, which provides information about using SMF	<a href="http://docs.oracle.com/cd/E23824_01/html/821-1451/hbrunlevels-25516.html">http://docs.oracle.com/cd/E23824_01/html/821-1451/hbrunlevels-25516.html</a>



Easier System Administration with the  
Oracle Solaris Service Management Facility  
July 2012, Revision 1.0  
Author: Glynn Foster

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200

[oracle.com](http://oracle.com)



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2012, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0611

**Hardware and Software, Engineered to Work Together**