

Gestión de los servicios del sistema en Oracle® Solaris 11.2



Referencia: E53840
Julio de 2014

Copyright © 2013, 2014, Oracle y/o sus filiales. Todos los derechos reservados.

Este software y la documentación relacionada están sujetos a un contrato de licencia que incluye restricciones de uso y revelación, y se encuentran protegidos por la legislación sobre la propiedad intelectual. A menos que figure explícitamente en el contrato de licencia o esté permitido por la ley, no se podrá utilizar, copiar, reproducir, traducir, emitir, modificar, conceder licencias, transmitir, distribuir, exhibir, representar, publicar ni mostrar ninguna parte, de ninguna forma, por ningún medio. Queda prohibida la ingeniería inversa, desensamblaje o descompilación de este software, excepto en la medida en que sean necesarios para conseguir interoperabilidad según lo especificado por la legislación aplicable.

La información contenida en este documento puede someterse a modificaciones sin previo aviso y no se garantiza que se encuentre exenta de errores. Si detecta algún error, le agradeceremos que nos lo comunique por escrito.

Si este software o la documentación relacionada se entrega al Gobierno de EE.UU. o a cualquier entidad que adquiera licencias en nombre del Gobierno de EE.UU. se aplicará la siguiente disposición:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Este software o hardware se ha desarrollado para uso general en diversas aplicaciones de gestión de la información. No se ha diseñado ni está destinado para utilizarse en aplicaciones de riesgo inherente, incluidas las aplicaciones que pueden causar daños personales. Si utiliza este software o hardware en aplicaciones de riesgo, usted será responsable de tomar todas las medidas apropiadas de prevención de fallos, copia de seguridad, redundancia o de cualquier otro tipo para garantizar la seguridad en el uso de este software o hardware. Oracle Corporation y sus filiales declinan toda responsabilidad derivada de los daños causados por el uso de este software o hardware en aplicaciones de riesgo.

Oracle y Java son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

Intel e Intel Xeon son marcas comerciales o marcas comerciales registradas de Intel Corporation. Todas las marcas comerciales de SPARC se utilizan con licencia y son marcas comerciales o marcas comerciales registradas de SPARC International, Inc. AMD, Opteron, el logotipo de AMD y el logotipo de AMD Opteron son marcas comerciales o marcas comerciales registradas de Advanced Micro Devices. UNIX es una marca comercial registrada de The Open Group.

Este software o hardware y la documentación pueden ofrecer acceso a contenidos, productos o servicios de terceros o información sobre los mismos. Ni Oracle Corporation ni sus filiales serán responsables de ofrecer cualquier tipo de garantía sobre el contenido, los productos o los servicios de terceros y renuncian explícitamente a ello. Oracle Corporation y sus filiales no se harán responsables de las pérdidas, los costos o los daños en los que se incurra como consecuencia del acceso o el uso de contenidos, productos o servicios de terceros.

Contenido

Uso de esta documentación	13
1 Introducción a la Utilidad de gestión de servicios	15
Capacidades de SMF	15
Nuevas funciones de esta versión	16
Conceptos y componentes SMF	17
Servicio SMF	18
Modelos de servicio	20
Nombres de servicio	20
Estados de servicio	21
Dependencias de servicio	21
Reiniciadores de servicio	22
Propiedades de servicio y grupos de propiedades	23
Repositorio de configuración de servicios	24
Archivos de configuración y servicios SMF	28
Privilegios de gestión de servicios	30
2 Obtención de información sobre los servicios	31
Enumeración de servicios en el sistema	31
Visualización de estado de servicio	31
Visualización de más información sobre los servicios	32
Visualización de información de servicio seleccionado	34
Visualización de dependencias de servicio	34
Agrupaciones de dependencia	35
Enumeración de instancias de las que depende un servicio	36
Enumeración de instancias que dependen de un servicio	37
Visualización de si un servicio se reiniciará automáticamente	38
Obtención de más información sobre estados de servicio	39
Visualización de archivos log de servicio	40
Inspección de configuración de servicio	42

Visualización de descripciones de propiedades y grupos de propiedades	42
Visualización de valores de propiedad de instancia y servicio	43
Visualización de propiedades en un tipo de grupo de propiedades	45
Visualización de la capa donde se define un valor	47
Visualización de valores en una instantánea especificada	48
Visualización de personalizaciones de configuración	48
Visualización de parámetros de notificación de eventos	49
3 Administración de servicios	53
Gestión de instancias de servicio SMF	53
Inicio de un servicio	53
Detención de un servicio	56
Reinicio de un servicio	58
Relectura de configuración de servicio	59
Supresión de un servicio	60
Configuración de notificación de transición de estado y eventos FMA	62
4 Configuración de servicios	65
Uso del comando de configuración de servicio	65
Invocación de un editor de propiedades	66
Invocación de svccfg de manera interactiva o con un archivo	67
Configuración de valores de propiedad	68
▼ Cómo modificar un valor de propiedad ttymon	70
▼ Cómo modificar una variable de entorno para un entorno de proceso de servicio	71
Agregación de grupos de propiedades, propiedades y valores de propiedad	73
Supresión de grupos de propiedades, propiedades y valores de propiedad	74
Supresión de configuración administrativa	75
Supresión de configuración no administrativa	77
Agregación de instancias de servicio	78
Reversión de instantáneas	79
Importación y la aplicación de manifiestos y perfiles	79
Configuración de varios sistemas	80
▼ Cómo crear un perfil mediante svcbundle	81
▼ Cómo crear un perfil mediante svccfg	81
Modificación de los servicios controlados por inetd	82
▼ Cómo cambiar un valor de propiedad para un servicio controlado por inetd	83
Modificación de servicios que están configurados por un archivo	84

5 Uso de SMF para controlar la aplicación	87
Creación de un servicio SMF	87
▼ Cómo crear un servicio SMF mediante la herramienta de generador de paquete de servicio	88
Creación de un servicio para iniciar o detener una instancia de Oracle Database	89
Servicios de nombres, instancias, propiedades y grupos de propiedades	96
▼ Cómo convertir una secuencia de comandos de control en ejecución en un servicio SMF	97
Uso de un cliché para crear un archivo de configuración	98
▼ Cómo crear un servicio de cliché	98
Ejemplos de servicio de cliché en Oracle Solaris	99
 A Prácticas recomendadas y solución de problemas de SMF	105
Prácticas recomendadas de SMF	105
Resolución de problemas de servicio	106
Comprensión de cambios de configuración	106
Reparación de una instancia degradada, fuera de línea o en mantenimiento	107
Marcación de una instancia como degradada o en mantenimiento	111
Diagnóstico y reparación de problemas del repositorio	111
Especificación de la cantidad de mensajes de inicio	114
Especificación de hito SMF con el cual reiniciar	115
Uso de SMF para investigar problemas de inicio de sistema	117
Conversión de servicios inetd a los servicios SMF	119
 Índice	121

Lista de figuras

FIGURA 1-1	Estructura de Utilidad de gestión de servicios	18
FIGURA 2-1	Relaciones de dependencia de servicio	35

Lista de tablas

TABLA 2-1	Reinicio automático de un servicio después de que se detiene una dependencia	39
TABLA A-1	Niveles de registro de mensajes de inicio SMF	115
TABLA A-2	Hitos de inicio SMF y niveles de ejecución correspondientes	116

Lista de ejemplos

EJEMPLO 2-1	Enumeración de todos los servicios activados	31
EJEMPLO 2-2	Enumeración de todos los servicios instalados	32
EJEMPLO 2-3	Enumeración de todas las instancias de un servicio	32
EJEMPLO 2-4	Visualización de procesos iniciados por un servicio de contrato	33
EJEMPLO 2-5	Visualización de un reinicio de servicio de contrato automáticamente después de la detención de un proceso	33
EJEMPLO 2-6	Enumeración de instancias y propiedades heredadas actualmente en uso	43
EJEMPLO 2-7	Enumeración de propiedades especificadas o grupos de propiedades actualmente en uso	44
EJEMPLO 2-8	Enumeración de valores de instancia y servicios en la vista de edición	44
EJEMPLO 2-9	Enumeración de propiedades especificadas o grupos de propiedades en la vista de edición	45
EJEMPLO 2-10	Visualización de grupos de propiedades por sus tipos	46
EJEMPLO 2-11	Enumeración de propiedades de un tipo de grupo de propiedades	46
EJEMPLO 3-1	Activación de una instancia de servicio permanentemente	55
EJEMPLO 3-2	Activación de una instancia de servicio temporalmente	55
EJEMPLO 3-3	Desactivación de una instancia de servicio	58
EJEMPLO 3-4	Configuración de una notificación global para un evento de estado de servicio	62
EJEMPLO 3-5	Configuración de una notificación para una instancia de servicio especificada	63
EJEMPLO 3-6	Configuración de una notificación para un evento FMA	63
EJEMPLO 3-7	Supresión de valores de notificación	63
EJEMPLO 4-1	Configuración de un valor simple	68
EJEMPLO 4-2	Configuración de un valor que contiene un carácter de dos puntos	69
EJEMPLO 4-3	Configuración de un valor que contiene espacios	69
EJEMPLO 4-4	Configuración de un valor que es un conjunto de valores	69
EJEMPLO 4-5	Agregación de un valor	70
EJEMPLO 4-6	Uso de addpg para crear un nuevo grupo de propiedades	73
EJEMPLO 4-7	Uso de setprop para crear una nueva propiedad	74
EJEMPLO 4-8	Uso de addpropvalue para crear una nueva propiedad	74

EJEMPLO 4-9	Supresión de todos los valores de una propiedad	75
EJEMPLO 4-10	Supresión de todos los valores coincidentes de una propiedad	75
EJEMPLO 4-11	Supresión de una propiedad	76
EJEMPLO 4-12	Supresión de un grupo de propiedades	76
EJEMPLO 4-13	Supresión de personalizaciones	76
EJEMPLO 4-14	Supresión de configuración que admite paquetes	77
EJEMPLO 4-15	Desenmascaramiento de configuración	78
EJEMPLO 4-16	Instalación automática de un perfil mediante svcbundle	81
EJEMPLO 4-17	Modificación del comando para ejecutarse cuando se inicia un servicio controlado por inetd	83
EJEMPLO 5-1	Instalación automática de un manifiesto generado	89

Uso de esta documentación

- **Descripción general:** describe cómo utilizar la función Utilidad de gestión de servicios (SMF) de Oracle Solaris. SMF es uno de los componentes de la capacidad más amplia de reparación automática predictiva de Oracle Solaris.
- **Destinatarios:** los administradores del sistema que gestionan los servicios del sistema.
- **Conocimientos necesarios:** experiencia en la administración de sistemas Oracle Solaris.

Biblioteca de documentación del producto

En la biblioteca de documentación (<http://www.oracle.com/pls/topic/lookup?ctx=E56339>), se incluye información de última hora y problemas conocidos para este producto.

Acceso a My Oracle Support

Los clientes de Oracle tienen acceso a soporte electrónico por medio de My Oracle Support. Para obtener más información, visite <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> o, si tiene alguna discapacidad auditiva, visite <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>.

Comentarios

Envíenos comentarios acerca de esta documentación mediante <http://www.oracle.com/goto/docfeedback>.

♦ ♦ ♦ 1 C A P Í T U L O 1

Introducción a la Utilidad de gestión de servicios

La estructura de la Utilidad de gestión de servicios (SMF) de Oracle Solaris gestiona servicios de aplicación y del sistema. SMF gestiona servicios del sistema esenciales para la operación del sistema y gestiona servicios de aplicación como una base de datos o servidor web. SMF mejora la disponibilidad de un sistema asegurándose de que los servicios de aplicación y del sistema esenciales se ejecuten de manera continua incluso en caso de fallos de hardware o software.

SMF reemplaza el uso de archivos de configuración para gestionar servicios y es el mecanismo que se recomienda usar para iniciar aplicaciones. SMF reemplaza el mecanismo de inicio de secuencias de comando `init`, las configuraciones `inetd.conf` y la mayoría de las secuencias de comandos `rc?.d`. SMF mantiene la compatibilidad con las prácticas administrativas existentes siempre que sea posible. Por ejemplo, la mayoría de los clientes y secuencias de comandos `rc` proporcionadas por ISV siguen funcionando de la misma manera que han funcionado sin SMF.

Capacidades de SMF

La estructura de SMF está siempre activa en el sistema Oracle Solaris 11. SMF proporciona las siguientes capacidades:

- Iniciar más rápido. SMF acelera el inicio en sistemas grandes, ya que inicia los servicios en paralelo según las dependencias de los servicios.
- Reiniciar servicios con fallos. Los servicios SMF tienen relaciones de dependencia bien definidas con otros servicios. Si falla un servicio, SMF informa a cualquiera de los servicios dependientes afectados. SMF intenta automáticamente reiniciar los servicios con fallos en orden de dependencia.
- Inspeccionar servicios. Ver las relaciones entre los servicios y los procesos. Ver los valores de las propiedades de servicio.
- Gestionar servicios. Activar, desactivar y reiniciar servicios. Estos cambios pueden persistir durante actualizaciones y reinicios, o puede especificar cambios temporales.
- Configurar servicios.
 - Cambiar los valores de las propiedades de servicio.
 - Agregar y suprimir propiedades personalizadas.

- Auditar cambios de servicio. SMF escribe registros de auditoría de Solaris para cada cambio administrativo para un servicio o sus propiedades. SMF puede mostrar si un valor de propiedad o estado de servicio fue establecido por un administrador.
- Delegar de manera segura tareas a usuarios que no sean root, incluida la posibilidad de modificar propiedades y activar, desactivar o reiniciar servicios.
- Crear nuevos servicios. Crear fácilmente una nueva instancia de un servicio existente, copiar y modificar un servicio existente, o bien utilizar una herramienta de creación de servicios.
- Depurar problemas de servicio. Mostrar fácilmente una explicación de por qué un servicio activado no se está ejecutando o por qué un servicio impide que otro servicio se ejecute.
- Configurar la manera en que se le notificará de eventos de software en particular o fallos de hardware.
- Convertir configuraciones `inetd.conf` a servicios SMF.
- Convertir propiedades de servicio SMF a archivos de configuración. Este mecanismo proporciona un puente para servicios que se gestionan mediante SMF pero interactúan con aplicaciones que aún requieren archivos de configuración.

Nuevas funciones de esta versión

Las siguientes funciones SMF son nuevas en esta versión:

Reinicio sincrónico y operaciones de refrescamiento

Una nueva opción `-s` para los subcomandos `restart` y `refresh` del comando `svcadm` especifica una operación sincrónica similar a la de la opción `-s` existente para los subcomandos `enable` y `disable`.

Mostrar archivos log

Una nueva opción `-L` para el comando `svcs` muestra el archivo log completo, las últimas líneas del archivo log o el nombre de ruta completo del archivo log para la instancia de servicio especificada.

Operación de lote en un conjunto de instancias de servicio

Nuevas llamadas de función `libscf(3LIB)` permiten que operaciones administrativas se agrupen y ejecuten en un lote de manera sincrónica o asincrónica. Puede activar, desactivar, reiniciar, refrescar, marcar y borrar el estado degradado o de mantenimiento, y la transición entre hitos para un conjunto de instancias de servicio.

Conversión de las propiedades de servicio SMF para archivos de configuración

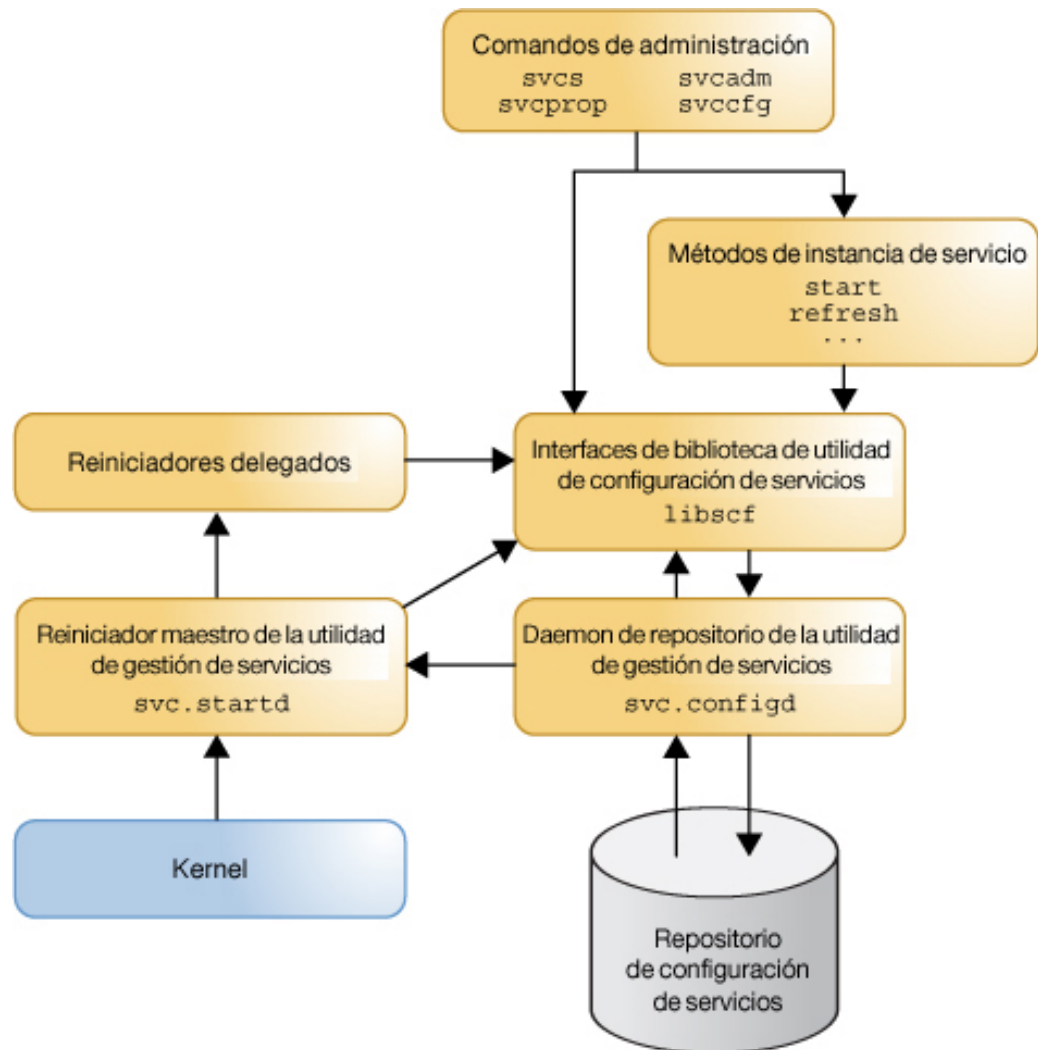
Para los servicios gestionados por SMF pero que interactúan con aplicaciones que aún requieren archivos de configuración, se crea una nueva capacidad de cliché y se actualiza

un archivo de configuración a partir de los valores de propiedad definidos para el servicio en el repositorio de configuración del servicio. Un nuevo archivo denominado archivo de cliché contiene la definición estructural del archivo de configuración que SMF creará. Una nueva utilidad `svcio` genera el archivo de configuración a partir de las definiciones en el archivo de cliché y las propiedades del servicio SMF. Esta función le permite aprovechar las ventajas de la gestión de configuración SMF con un mínimo de cambios en la aplicación existente.

Conceptos y componentes SMF

En esta sección, se definen los términos que se utilizan en el resto de esta guía. La figura siguiente muestra los componentes principales de la estructura SMF. Cuando inicia una imagen, SMF actualiza el repositorio de configuración de servicio, si es necesario, lee los datos del repositorio e inicia las instancias de servicio activadas. En la siguiente figura, `libscf` es la interfaz de biblioteca que los reiniciadores utilizan para interactuar con el repositorio de configuración de servicio. La interacción entre el repositorio de configuración de servicio y las interfaces de biblioteca `libscf` es gestionada por el daemon `svc.configd`. Los comandos `svcs`, `svccprop`, `svccadm` y `svccfg` son la interfaz que los administradores utilizan para interactuar con el repositorio de configuración de servicio.

FIGURA 1-1 Estructura de Utilidad de gestión de servicios



Servicio SMF

Un *servicio* SMF es una aplicación que se ejecuta de manera persistente que representa una entidad de sistema como la siguiente:

- Servicios de aplicación como una base de datos o un servidor web
- Servicios del sistema esenciales
- El estado del software de un dispositivo
- Información de configuración de núcleo
- Hitos que corresponden a un estado `init` del sistema

Una *instancia de servicio* es un elemento secundario de un servicio y proporciona capacidades y relaciones de dependencia para las aplicaciones y otras instancias de servicio. Sólo las instancias tienen un estado y pueden iniciarse y detenerse. Si una instancia falla por cualquier motivo, como un fallo de hardware o software, SMF detecta automáticamente el fallo y reinicia la instancia y cualquier instancia dependiente.

Las instancias de un servicio permiten que varias configuraciones de un servicio se ejecuten de manera simultánea. Las instancias de servicio heredan y personalizan la configuración de servicio común. Por ejemplo, puede definir un servicio de servidor web con una instancia configurada para recibir en el puerto 80 y otra instancia configurada para recibir en el puerto 1008. La mayoría de los servicios tienen una instancia `default`. Unos pocos servicios no tienen instancias, como algunos servicios que utilizan SMF para almacenar configuraciones pero no para ejecutar programas. Por ejemplo, el servicio `x11/x11-server` no tiene ninguna instancia.

Un servicio SMF se describe en un archivo denominado un *manifiesto* de servicio. El manifiesto describe las instancias de servicio, dependencias, propiedades de configuración y métodos. Los *métodos* de servicio inician, detienen y refrescan instancias de servicio. Un método puede ser un daemon, otro ejecutable binario o una secuencia de comandos ejecutable. Un archivo de *perfil* de servicio le permite personalizar un servicio existente, principalmente mediante la agregación de propiedades y la agregación y sustitución de valores de propiedad. Las nuevas propiedades y valores se colocan en capas sobre los valores asignados en el manifiesto, como se describe en [“Capas de repositorio” \[26\]](#). Consulte [“Paquetes de servicio” \[25\]](#) para obtener más información sobre manifiestos y perfiles. Un perfil también es una excelente herramienta para aplicar la misma configuración personalizada en varios sistemas, como se describe en [“Configuración de varios sistemas” \[80\]](#).

La información de servicio se almacena en el *repositorio de configuración de servicio*, que también es denominado *base de datos SMF*. El repositorio de configuración de servicio almacena el estado actual de cada instancia de servicio en el sistema y los datos de configuración para cada servicio e instancia de servicio. Los datos se almacenan en *capas* según cómo fueron modificados los valores, como se describe en [“Capas de repositorio” \[26\]](#).

SMF proporciona *acciones* que puede invocar en una instancia de servicio, incluidas acciones como activar, desactivar, refrescar y reiniciar. Cada instancia de servicio es gestionada por un *reiniciador*, que realiza estas acciones administrativas. En general, los reiniciadores realizan acciones ejecutando *métodos* para mover la instancia de servicio de un estado a otro. Para obtener más información sobre los reiniciadores, consulte [“Reiniciadores de servicio” \[22\]](#).

Un *servicio de hitos* es un tipo especial de servicio que representa un nivel de disponibilidad del sistema. Un hito es un servicio del que dependen otras instancias para iniciarse. Por ejemplo, los niveles de ejecución están representados por servicios de hitos como `svc:/milestone/multi-`

user-server. Los hitos también se pueden utilizar para indicar la disponibilidad de un grupo de servicios, como `svc:/milestone/devices`, `svc:/milestone/network` o `svc:/milestone/name-services`.

Modelos de servicio

Los servicios SMF son uno de los siguientes tres modelos:

Servicio transitorio

El servicio realiza parte del trabajo y, luego, sale sin iniciar ningún proceso de larga ejecución.

Servicio secundario o de espera

El servicio se reinicia siempre que su proceso secundario se cierra limpiamente. Un proceso secundario que se cierra limpiamente no se trata como un error.

Servicio de daemon o contrato

El servicio inicia un daemon de larga ejecución o inicia varios procesos relacionados que están unidos conjuntamente como parte de un *contrato de servicio*. El servicio de contrato gestiona los procesos que inicia y cualquier servicio dependiente y su orden de inicio. Sólo es necesario gestionar el servicio de alto nivel.

Nombres de servicio

Cada servicio e instancia de servicio es representada por un Identificador de recurso de gestión de errores (FMRI). El FMRI completo para una instancia de servicio tiene el siguiente formato:

```
svc:/service_name:instance_name
```

El *service_name* es un nombre jerárquico como `network/dns/client` o `application/pkg/server`. Los componentes de *service_name* que preceden la barra inclinada final (/) son las *categorías* del servicio. Las categorías como `application`, `device`, `milestone`, `network` y `system` ayudan a identificar el propósito del servicio.

La categoría `site` está reservada para ayudarle a evitar conflictos de nombre cuando crea sus propios servicios SMF. Por ejemplo, un servicio específico de sitio denominado `svc:/site/tool` no entrará en conflicto con un servicio de Oracle Solaris denominado `svc:/tool`.

Los nombres de instancia de servicio se anexan al nombre del servicio principal después de un carácter de dos puntos. Por ejemplo, `svc:/system/identity:node` y `svc:/system/identity:domain` son instancias del servicio `svc:/system/identity`.

En las secuencias de comandos, la práctica recomendada es utilizar el nombre de instancia de servicio completo. De manera interactiva, los nombres pueden abreviarse según las partes que están más hacia la derecha del nombre, lo que da como resultado un nombre único. Por ejemplo, `svc:/system/identity` se puede abreviar a `identity` y `svc:/system/identity:domain` se puede abreviar a `identity:domain`. Los nombres de instancia deben estar precedidos por parte del nombre de servicio, seguidos por un carácter de dos puntos.

Estados de servicio

En cualquier momento específico, una instancia de servicio SMF está en uno de los siguientes estados:

- **degraded**: la instancia se está ejecutando o está disponible para su ejecución, pero está funcionando a una capacidad limitada.
- **disabled**: la instancia no está activada y no está en ejecución o disponible para ejecutarse.
- **maintenance**: la instancia está activada, pero no puede ejecutarse. La instancia puede estar transitando por el estado **maintenance** debido a que una acción administrativa aún no se ha completado. De lo contrario, es necesario realizar una acción administrativa para resolver el problema.
- **offline**: la instancia está activada, pero no está en ejecución o disponible para ejecutarse. Por ejemplo, si no se cumplen las dependencias de un servicio activado, el servicio se mantiene en el estado **offline**.
- **online**: la instancia está activada y en ejecución o disponible para ejecutarse. El estado **online** es el estado operativo esperado para una instancia de servicio configurada correctamente con todas las dependencias cumplidas.
- **uninitialized**: éste es el estado inicial para todos los servicios.

Una instancia de servicio transita entre estados según las condiciones, como acciones administrativas o el estado de sus servicios dependientes. Por ejemplo, al activar una instancia que estaba en el estado **disabled**, la instancia recién activada primero transita al estado **offline** y luego al estado **online** cuando se cumplen todas sus dependencias. Consulte la página del comando `man smf(5)` para obtener más información sobre estos estados de servicios y cómo las instancias de servicio transitan por estos estados.

Dependencias de servicio

Un servicio puede tener una *dependencia* en un servicio, una instancia de servicio o un archivo. Las dependencias de servicio definen relaciones entre servicios.

Las relaciones de dependencia determinan cuándo un servicio se inicia y se detiene automáticamente. Cuando las dependencias de un servicio activado no se cumplen, el servicio tiene el estado `offline`. Cuando las dependencias de un servicio activado se cumplen, se inicia el servicio. Si el inicio del servicio se realiza correctamente, el servicio transita al estado `online`.

Las dependencias del servicio se vuelven a evaluar a medida que los servicios transitan por los estados. Las dependencias de servicio que se cumplen pueden no cumplirse más adelante. Las dependencias de archivos se evalúan sólo una vez.

Las dependencias pueden ser necesarias u opcionales. Es posible que las dependencias de servicio deban estar en ejecución o desactivadas. Un servicio dependiente se puede configurar para reiniciarse o no cuando una de sus dependencias de servicio se detiene o refresca.

Las relaciones de dependencia permiten las siguientes capacidades:

- Procesos de inicialización escalables y reproducibles
- Un inicio más rápido del sistema en equipos que tienen capacidades paralelas al iniciar servicios independientes en paralelo
- Contención precisa de fallos y recuperación de fallos reiniciando únicamente los servicios que son afectados directamente por un fallo y reiniciando estos servicios en orden de dependencia

Reiniciadores de servicio

Cada instancia de servicio SMF está gestionada por un *reiniciador*. El reiniciador recupera la configuración de la instancia y brinda un entorno de ejecución. Consulte [smf_restarter\(5\)](#) para obtener información común a todos los reiniciadores.

Daemon de reiniciador maestro

El daemon `svc.startd` es el daemon de *reiniciador maestro* para la SMF y el *reiniciador predeterminado* para todas las instancias de servicio. El daemon `svc.startd` gestiona los estados para todas las instancias de servicio y sus dependencias. A medida que se cumplen las dependencias cuando las instancias se desplazan al estado en línea, el reiniciador maestro invoca métodos de inicio de otras instancias o dirige el reiniciador delegado para invocar el método de inicio. El reiniciador maestro detiene una instancia de servicio cuando las dependencias de la instancia ya no se cumplen. El reiniciador intenta reiniciar una instancia si la instancia falla. Debido a que una instancia no puede estar en línea hasta que todas sus dependencias se cumplan, las dependencias de una instancia ayudan a determinar el comportamiento de reinicio de la instancia. Las propiedades establecidas en cada declaración de

dependencia definen si esa dependencia es requerida y en qué casos la instancia se reiniciará si se reinicia la dependencia.

Entre otras tareas, el daemon `svc.startd` inicia las secuencias de comandos `/etc/rc*.d` adecuadas en los niveles de ejecución adecuados, que es trabajo previamente realizado por `init`.

El siguiente ejemplo muestra que `svc.startd` es el reiniciador para la instancia de servicio `network/impd:default`. Se ha omitido otra salida de este ejemplo.

```
$ svcs -l impd:default
restarter    svc:/system/svc/restarter:default
```

Si la propiedad `restarter` está vacía o establecida en `svc:/system/svc/restarter:default`, la instancia de servicio es gestionada por `svc.startd`. Para obtener más información sobre el daemon `svc.startd`, consulte la página del comando `man svc.startd\(1M\)`.

Reiniciadores delegados

Algunos servicios tienen un conjunto de comportamientos comunes en el inicio. Un *reiniciador delegado* puede proporcionar un entorno de ejecución específico y un comportamiento de reinicio específico de la aplicación para esos servicios.

Un ejemplo de un reiniciador delegado es `inetd`, que puede iniciar servicios de Internet a petición, en lugar de tener los servicios ejecutándose en todo momento. El reiniciador `inetd` proporciona sus instancias de servicio con un entorno compuesto de una conexión de red como descriptores de archivos de entrada y salida. Para obtener más información sobre el daemon `inetd`, consulte la página del comando `man inetd\(1M\)`. El siguiente ejemplo muestra que `inetd` es el reiniciador para la instancia de servicio `cups/in-lpd:default`. Se ha omitido otra salida de este ejemplo.

```
$ svcs -l cups/in-lpd:default
restarter    svc:/network/inetd:default
```

El reiniciador delegado especificado por la propiedad `restarter` es responsable de gestionar la instancia de servicio una vez que el reiniciador está disponible.

Propiedades de servicio y grupos de propiedades

La información sobre los servicios, incluidas dependencias, métodos, estado y datos de aplicación, se almacena en el repositorio de configuración de servicio como un conjunto de *propiedades*. Las propiedades se pueden definir en el servicio o una instancia del servicio. Las propiedades que se establecen en el servicio son heredadas por todas las instancias de

dicho servicio. Las propiedades que se establecen en una instancia son utilizadas sólo por esa instancia. Las instancias de servicio pueden personalizar los valores de las propiedades heredadas y pueden definir propiedades adicionales que no se han definido para el servicio principal.

Las propiedades se organizan en *grupos de propiedades*. Algunos grupos de propiedades comunes incluyen:

- **general**: contiene información como, por ejemplo, si la instancia está activada.
- **restarter**: contiene información de tiempo de ejecución que es almacenada por el reiniciador para el servicio, incluido el estado actual de la instancia.
- **start, refresh, stop**: contiene información acerca de qué programa ejecutar para iniciar, refrescar o detener el servicio.
- **config**: utilizada por desarrolladores de servicio para mantener datos de aplicación.

Consulte la página del comando `man smf(5)` para obtener más información sobre las propiedades y los grupos de propiedades.

Repositorio de configuración de servicios

La información acerca de cada servicio se almacena en el *repositorio de configuración de servicio*, que también es denominado *base de datos SMF*. El repositorio de configuración de servicio almacena información como el estado actual de cada instancia de servicio en el sistema y las propiedades de cada servicio e instancia de servicio.

El repositorio almacena información de configuración persistente como también datos de tiempo de ejecución SMF para servicios.

- La información de configuración persistente se almacena en capas según el origen de los datos. Consulte “[Capas de repositorio](#)” [26].
- Los datos de tiempo de ejecución o la información de configuración no persistente no se conservan después del reinicio, y el repositorio no almacena la información de capas para datos no persistentes. Los datos no persistentes, por lo general, mantienen un estado de programa activo.

El repositorio también almacena los datos de plantilla de servicio, como tipos, restricciones de valor y descripciones de propiedades. Los datos de plantilla se definen en el manifiesto de servicio. Consulte la página del comando `man smf_template(5)` para obtener más información sobre los datos de plantilla.

El repositorio de configuración de servicios sólo se puede manipular o consultar mediante interfaces de la SMF. Utilice los comandos `svcs`, `svccprop`, `svcadm` y `svccfg` o las funciones de biblioteca de Utilidad de gestión de servicios enumeradas en la página del comando `man libscf(3LIB)`. Puede leer y escribir los valores de propiedad y mostrar los valores

de propiedad en las instantáneas y las capas especificadas. Para obtener información sobre las capas, consulte [“Capas de repositorio” \[26\]](#). Para obtener información sobre las instantáneas, consulte [“Instantáneas del repositorio” \[27\]](#). Puede mostrar sólo las propiedades de la instancia de servicio seleccionada o servicio principal, o puede mostrar una vista *compuesta* de propiedades. En una vista compuesta, se muestran tanto las propiedades establecidas en el servicio principal como las propiedades establecidas en la instancia de servicio; los valores mostrados son los valores definidos en la instancia de servicio.

Paquetes de servicio

Un paquete de servicio es un archivo XML que contiene la información que se almacena en el repositorio de configuración de servicio para un servicio o instancia de servicio. La información proporcionada en los paquetes de servicio se almacena en el repositorio de configuración de servicio y se pueden exportar desde el repositorio. Los paquetes de servicio en ubicaciones estándar se importan al repositorio durante el inicio del sistema.

Los dos tipos de paquetes de servicio son manifiestos y perfiles.

Manifiestos	Los manifiestos contienen el conjunto completo de propiedades asociadas con un conjunto de servicios o instancias de servicio específicos.
Perfiles	Normalmente los perfiles proporcionan la personalización de un servicio o instancia de servicio que aumenta o sustituye la información proporcionada en el manifiesto. Los ejemplos de las personalizaciones incluyen propiedades adicionales y valores de propiedad modificados.

La *ubicación estándar* para los manifiestos es `/lib/svc/manifest`. La ubicación estándar para los perfiles es `/etc/svc/profile`.

Cuando se inicia el sistema o se reinicia el servicio de importación de manifiestos, se importan los manifiestos y se aplican los perfiles si son nuevos o están modificados. Un paquete IPS que ofrece un paquete de servicio puede especificar que el servicio de importación de manifiesto debe reiniciarse cuando se instala el paquete.

Las personalizaciones locales pueden proporcionarse en archivos de perfil con un sufijo `.xml` en el directorio `/etc/svc/profile/site`. Si la misma propiedad en la misma capa de repositorio para el mismo servicio o instancia es definida por varios manifiestos, la SMF no puede determinar el valor que se debe utilizar. Cuando se detecta este tipo de conflicto, la instancia se coloca en el estado de mantenimiento. Consulte [“Capas de repositorio” \[26\]](#) para obtener más información sobre las capas.

Además de la entrega de servicios en Oracle Solaris, los paquetes de servicio también pueden brindar configuración personalizada a una gran variedad de sistemas.

Un perfil de sistema, `/etc/svc/profile/generic.xml`, se aplica durante la instalación. No cambie este perfil. Cualquier cambio realizado a este perfil de sistema se sobrescribirá en la

actualización. Consulte la página del comando `man smf_bootstrap(5)` para obtener más información.

Capas de repositorio

El repositorio de configuración de servicio puede almacenar distintos valores para una sola propiedad. El repositorio almacena datos en *capas* según el origen de los datos. El origen puede ser manifiestos, perfiles del sistema, perfiles de sitio y personalizaciones realizadas mediante comandos SMF e interfaces de biblioteca. Puede ver los valores en las diferentes capas para comprender el origen del valor en la configuración en ejecución: si se ha asignado un valor en el manifiesto, en un perfil, o ha sido modificado por un administrador.

Los cambios de configuración realizados mediante comandos SMF e interfaces de biblioteca sólo aparecen en la capa `admin`. La configuración en otras capas se define en archivos de manifiesto y perfil en ubicaciones estándar. Cuando una propiedad se agrega al repositorio desde un archivo, la información acerca de dicha propiedad incluye el nombre de dicho archivo.

Capa	Contenido
<code>admin</code>	Cualquier cambio realizado usando los comandos SMF o interfaces de biblioteca, por un administrador o por una aplicación. La capa <code>admin</code> también incluye cualquier cambio realizado mediante la importación de un manifiesto o aplicación de un perfil desde una ubicación no estándar. Consulte “ Importación y la aplicación de manifiestos y perfiles ” [79] para advertencias acerca del uso de ubicaciones no estándar.
<code>site-profile</code>	Cualquier valor de los archivos de perfil en el directorio <code>/etc/svc/profile/site</code> o los perfiles antiguos <code>/etc/svc/profile/site.xml</code> y <code>/var/svc/profile/site.xml</code> . Tenga en cuenta que <code>/var/svc/profile</code> se descarta como una ubicación estándar y no debería utilizarse para nuevos perfiles.
<code>system-profile</code>	Cualquier valor de los perfiles del sistema <code>/etc/svc/profile/generic.xml</code> y <code>/etc/svc/profile/platform.xml</code> .
<code>manifest</code>	Valores de los manifiestos en los directorios <code>/lib/svc/manifest</code> y <code>/var/svc/manifest</code> . Tenga en cuenta que <code>/var/svc/manifest</code> se descarta como una ubicación estándar y no debería utilizarse para nuevos manifiestos.

No se permiten conflictos de propiedad dentro de cualquier capa. Una propiedad en conflicto en la capa `admin` sobrescribe la propiedad anterior. Si la misma propiedad es entregada por varios archivos en cualquier otra capa, y no está definida en una capa superior, la instancia se etiqueta como en conflicto y no se inicia hasta que la definición conflictiva se elimine o la propiedad se defina en una capa superior.

Puede especificar el nivel de datos de configuración para ver y, por lo tanto, identificar qué datos son personalizaciones administrativas y qué datos se entregaron con el software. Cuando un cliente no especifica la capa desde la que se recuperan datos de configuración, se proporcionan los datos de la capa superior. La capa superior está determinada por el siguiente orden de prioridad de arriba a abajo: capa `admin`, capa `site-profile`, capa `system-profile`,

capa `manifest`. Si una propiedad tiene un valor en la capa `admin`, ese es el valor que entrega el repositorio. De esta forma, las personalizaciones locales se prefieren sobre los valores que se proporcionan durante la instalación del sistema.

Instantáneas del repositorio

El repositorio captura una *instantánea* de sólo lectura de cada servicio cada vez que el servicio se inicia correctamente. Estas instantáneas le permiten volver fácilmente a un estado de trabajo anterior si es necesario. Las siguientes instantáneas pueden estar disponibles para cualquier instancia determinada:

<code>initial</code>	La configuración inicial cuando el servicio y sus instancias se han importado por primera vez. Una instantánea <code>initial</code> no se crea si un perfil inicia el servicio o la instancia antes de la importación de manifiesto.
<code>previous</code>	La configuración actual que se captura cuando se realiza una importación de manifiesto para un servicio que ya se ha entregado. Puede que el servicio ya haya sido entregado por el manifiesto que se está importando o por otro manifiesto.
<code>running</code>	La configuración en ejecución de la instancia de servicio. Al cambiar los datos de configuración, utilice el comando <code>svcadm refresh</code> o el comando <code>svccfg refresh</code> para ascender los nuevos valores a la instantánea en ejecución.
<code>start</code>	Configuración capturada durante una transición correcta al estado <code>online</code> .

Copias de seguridad del repositorio

SMF automáticamente realiza las siguientes *copias de seguridad* del repositorio de configuración de servicio:

- La copia de seguridad `boot` se realiza inmediatamente antes de realizar el primer cambio en el repositorio durante cada inicio del sistema.
- Las copias de seguridad `manifest_import` se producen antes de que se complete `svc:/system/early-manifest-import:default` o `svc:/system/manifest-import:default`, si el servicio importó nuevos manifiestos o ejecutó secuencias de comandos de actualización.

El sistema conserva cuatro copias de seguridad de cada tipo y suprime las copias de seguridad más antiguas, según sea necesario.

Puede restaurar el repositorio a partir de una de estas copias de seguridad. Consulte [Cómo restaurar un repositorio desde una copia de seguridad \[112\]](#).

Archivos de configuración y servicios SMF

SMF es el mecanismo recomendado a utilizar para iniciar las aplicaciones. En la mayoría de los casos, SMF reemplaza el uso de archivos de configuración para gestión de servicios. En esta sección se describe cómo se manejan algunos archivos y secuencias de comandos de configuración comunes.

secuencias de comandos `/etc/rc?.d`

Los directorios `/etc/rc?.d`, donde `?` representa un nivel de ejecución, contienen secuencias de comandos de inicialización y terminación antiguas para la gestión de servicios que se ejecutan en transiciones de nivel de ejecución. La mayoría de los servicios que fueron implementados anteriormente por secuencias de comandos `/etc/rc?.d` son gestionados por SMF. Algunas secuencias de comandos `/etc/rc?.d` se mantienen para permitirle el uso de aplicaciones de terceros que esperan estos servicios como secuencias de comandos `/etc/rc*.d`. Estas secuencias de comandos están fuertemente enlazadas a archivos en el directorio `/etc/init.d`. Para obtener información sobre secuencias de comandos `/etc/rc?.d` y sobre niveles de ejecución, consulte el archivo `/etc/init.d/README`, los archivos `README` en los directorios `/etc/rc?.d` y la página del comando `man inittab(4)`. Para obtener instrucciones para convertir una secuencia de comandos de control de ejecución, consulte [Cómo convertir una secuencia de comandos de control en ejecución en un servicio SMF \[97\]](#). Después de convertir una secuencia de comandos `rc?.d`, cambie el nombre de la secuencia de comandos de *Sscript* a *sscript* para eliminar la secuencia de comandos de manera eficaz.

secuencias de comandos `/etc/init.d`

El directorio `/etc/init.d` contiene secuencias de comandos de inicialización y terminación para cambiar estados `init`. Algunas de estas secuencias de comandos están fuertemente enlazadas a secuencias de comandos en los directorios `/etc/init.d`. Para obtener información sobre las secuencias de comandos `/etc/init.d`, consulte `/etc/init.d/README` y la página del comando `man init.d(4)`.

Las secuencias de comandos de control de ejecución `init.d` antiguas están representadas por FMRI de SMF que comienzan con `lrc` en lugar de `svc`. Por ejemplo, la secuencia de comandos de configuración `PPP /etc/rc2.d/S47pppd` está representada por el servicio `lrc:/etc/rc2_d/S47pppd`. El estado de estos servicios `lrc` es `legacy_run`. Como se muestra en el siguiente ejemplo, puede enumerar los nombres y las horas de inicio de servicios antiguos, pero no puede administrar estos servicios mediante SMF.

```
$ svcs lrc:\*
STATE          STIME          FMRI
```

```
legacy_run      9:34:54  lrc:/etc/rc2_d/S47pppd
legacy_run      9:34:54  lrc:/etc/rc2_d/S89PRESERVE
$ svcs -l lrc:/etc/rc2_d/S47pppd
svcs: Operation not supported for legacy service 'lrc:/etc/rc2_d/S47pppd'
$ svccfg -s lrc:/etc/rc2_d/S47pppd listprop
svccfg: Operation not supported for legacy service 'lrc:/etc/rc2_d/S47pppd'
```

entradas /etc/inittab

Las entradas en el archivo `/etc/inittab` controlan la distribución de procesos según `init`. No edite el archivo `/etc/inittab` directamente. En su lugar, modifique los servicios SMF. Consulte [Cómo modificar un valor de propiedad `ttymon` \[70\]](#) para obtener un ejemplo de cómo modificar un parámetro trasferido a `ttymon`.

Para obtener información sobre el formato de las entradas de archivo `/etc/inittab`, consulte la página del comando `man inittab(4)`. Para obtener información sobre los niveles de ejecución, consulte la página del comando `man inittab(4)` y `/etc/init.d/README`.

archivo `/etc/inetd.conf`

Los servicios que se configuraron anteriormente mediante el archivo `inetd.conf` ahora son configurados mediante SMF. Las configuraciones en el archivo `inetd.conf` deben convertirse a servicios SMF para que estén disponible para su uso. Consulte [“Conversión de servicios `inetd` a los servicios SMF” \[119\]](#). Para los servicios `inetd` que ya se convirtieron a servicios SMF, consulte [“Modificación de los servicios controlados por `inetd`” \[82\]](#).

archivo `/etc/nscd.conf`

archivo `/etc/nsswitch.conf`

archivo `/etc/resolv.conf`

No edite estos archivos. Se perderán las ediciones. Estos archivos se generan automáticamente a partir de datos SMF para compatibilidad con versiones anteriores de aplicaciones que posiblemente analicen el archivo. Utilice el comando `svccfg setprop` para modificar valores de propiedad como se muestra en [“Configuración de valores de propiedad” \[68\]](#).

La función del archivo `nscd.conf` es reemplazada por el servicio SMF `svc:/system/name-service-cache`. Consulte la página del comando `man nscd.conf(4)` para ver qué propiedades `name-service-cache` configurar en lugar de editar el archivo `nscd.conf`.

La función del archivo `nsswitch.conf` es reemplazada por el servicio SMF `svc:/system/name-service/switch`. Consulte la página del comando `man nsswitch.conf(4)` para ver qué propiedades `name-service/switch` configurar en lugar de editar el archivo `nsswitch.conf`.

La función del archivo `resolv.conf` es reemplazada por el servicio SMF `svc:/network/dns/client`. Consulte la página del comando `man resolv.conf(4)` para ver qué propiedades `dns/client` configurar en lugar de editar el archivo `resolv.conf`.

Estos archivos son ejemplos de los archivos de configuración que no debe editar. Existen otros archivos de este tipo. En algunos casos, la edición de un archivo de configuración es la forma correcta de modificar la configuración, como se describe en [“Modificación de servicios que están configurados por un archivo” \[84\]](#). Antes de editar cualquier archivo de configuración, lea los comentarios en el archivo y cualquier página del comando `man` asociada para asegurarse de que la edición del archivo es la manera correcta de modificar la configuración para el servicio relacionado.

Privilegios de gestión de servicios

La modificación de estado de servicio y configuración requiere más privilegios. Utilice uno de los siguientes métodos para obtener los privilegios que necesita. Consulte [“Protección de los usuarios y los procesos en Oracle Solaris 11.2”](#) para obtener más información sobre las autorizaciones, los perfiles y los roles, incluido cómo determinar qué perfil o rol necesita, y cómo asignar privilegios.

Autorizaciones

Consulte la página del comando `man smf_security(5)` para obtener información detallada sobre las autorizaciones requeridas para operaciones SMF. También puede inspeccionar un servicio concreto para propiedades como `action_authorization`, `modify_authorization`, `read_authorization` y `value_authorization`.

Perfiles de derechos

Utilice el comando `profiles` para obtener una lista de los perfiles de derechos que se le han asignado. El perfil de derechos Gestión de servicios le otorga las autoridades `solaris.smf.manage` y `solaris.smf.modify` y le permite usar los comandos `svcadm` y `svccfg`. El perfil de derechos Operador de servicios le otorga las autoridades `solaris.smf.manage` y `solaris.smf.modify.framework`.

Roles

Utilice el comando `roles` para obtener una lista de los roles que se le hayan asignado. Si tiene el rol `root`, puede utilizar el comando `su` para asumir el rol `root`.

Comando `sudo`

En función de la política de seguridad de su sitio, es posible que pueda utilizar el comando `sudo` con su contraseña de usuario para ejecutar un comando con privilegios.

♦ ♦ ♦ 2 CAPÍTULO 2

Obtención de información sobre los servicios

En este capítulo se muestra cómo obtener información sobre servicios como los siguientes:

- Estado de servicio, dependencias y otros valores de propiedad
- Procesos iniciados por un servicio de contrato
- Ubicación del archivo log para resolución de problemas
- Configuración de notificaciones de evento de transición de servicio y evento FMA

Enumeración de servicios en el sistema

El comando `svcs` es el comando principal para enumerar estados y estados de instancia de servicios.

Visualización de estado de servicio

Consulte [“Estados de servicio” \[21\]](#) para obtener descripciones de los estados que se muestran en estos ejemplos.

EJEMPLO 2-1 Enumeración de todos los servicios activados

Sin opciones u argumentos, el comando `svcs` muestra todas las instancias de servicio activadas en este sistema, así como las instancias que están desactivadas temporalmente.

Las instancias de servicio en el estado `disabled` en esta lista se activarán en el siguiente inicio del sistema. Las instancias en el estado `legacy_run` no están gestionadas por SMF. Consulte [“Archivos de configuración y servicios SMF” \[28\]](#) para obtener más información sobre estos servicios antiguos. Consulte [“Obtención de más información sobre estados de servicio” \[39\]](#) si tiene servicios en los estados `maintenance`, `degraded` u `offline`.

La columna `STIME` muestra la hora en que la instancia ingresó al estado mencionado. Si la instancia ingresó a este estado más de 24 horas atrás, la columna `STIME` muestra la fecha.

\$ `svcs`

STATE	STIME	FMRI
legacy_run	Sep_09	lrc:/etc/rc2_d/S47pppd
legacy_run	Sep_09	lrc:/etc/rc2_d/S81dodatadm_udaplt
legacy_run	Sep_09	lrc:/etc/rc2_d/S89PRESERVE
disabled	Sep_09	svc:/system/vbiosd:default
online	Sep_09	svc:/system/early-manifest-import:default
online	Sep_09	svc:/system/svc/restarter:default
...		

EJEMPLO 2-2 Enumeración de todos los servicios instalados

Para enumerar todas las instancias de servicio instaladas en este sistema, incluidas las instancias disabled que no se activarán automáticamente en el siguiente inicio, utilice el comando `svcs -a`.

```
$ svcs -a
```

Un asterisco (*) se agrega al estado para las instancias de servicio que realizan una transición del estado enumerado a otro estado. Por ejemplo, `offline*` probablemente significa que la instancia aún está ejecutando su método de inicio.

Un signo de interrogación (?) se muestra si el estado está ausente o no se reconoce.

EJEMPLO 2-3 Enumeración de todas las instancias de un servicio

Con un nombre de servicio especificado, el comando `svcs` muestra todas las instancias de un servicio. Consulte [“Visualización de información de servicio seleccionado” \[34\]](#) para obtener información sobre la opción `-o`.

```
$ svcs -Ho inst identity
node
domain
```

Visualización de más información sobre los servicios

El comando `svcs -l` muestra una lista larga para cada instancia de servicio especificada que incluye información más detallada sobre el estado de la instancia, las rutas al archivo log y los archivos de configuración para la instancia, los tipos de dependencias, los valores de atributos de reinicio de dependencia y el estado de dependencia. El siguiente ejemplo muestra que todas las dependencias necesarias de esta instancia de servicio están en línea. La dependencia que está desactivada es una dependencia opcional. Para obtener información sobre los tipos de dependencias y reiniciar los valores de atributo, consulte [“Visualización de dependencias de servicio” \[34\]](#). En la salida `svcs -l`, son posibles otros estados además de los descritos en [“Estados de servicio” \[21\]](#) para dependencias. Consulte la página del comando `man svcs(1)`

para obtener descripciones. El siguiente ejemplo muestra también que la instancia de servicio especificada está temporalmente activada, está en línea y el servicio es un servicio de tipo de contrato. Consulte [“Modelos de servicio” \[20\]](#) para obtener definiciones de tipos de servicio. Si el valor de estado tiene un asterisco final, por ejemplo `offline*`, la instancia está en transición y el siguiente campo de estado muestra un valor de estado en lugar de `none`. `state_time` es el tiempo en que la instancia ingresó al estado mencionado.

```
$ svcs -l net-snmp
fmri          svc:/application/management/net-snmp:default
name          net-snmp SNMP daemon
enabled       true (temporary)
state         online
next_state    none
state_time    September 17, 2013 05:57:26 PM PDT
logfile       /var/svc/log/application-management-net-snmp:default.log
restarter     svc:/system/svc/restarter:default
contract_id   160
manifest      /etc/svc/profile/generic.xml
manifest      /lib/svc/manifest/application/management/net-snmp.xml
dependency    require_all/none svc:/system/filesystem/local (online)
dependency    optional_all/none svc:/milestone/name-services (online)
dependency    optional_all/none svc:/system/system-log (online)
dependency    optional_all/none svc:/network/rpc/rstat (disabled)
dependency    require_all/restart svc:/system/cryptosvc (online)
dependency    require_all/restart svc:/milestone/network (online)
dependency    require_all/refresh file://localhost/etc/net-snmp/snmp/snmpd.conf (online)
dependency    require_all/none svc:/milestone/multi-user (online)
```

EJEMPLO 2-4 Visualización de procesos iniciados por un servicio de contrato

Utilice el comando `svcs -p` para mostrar los identificadores de proceso y los nombres de comandos de procesos iniciados por una instancia de servicio de contrato. El servicio `net-snmp` gestiona el agente SNMP `/usr/sbin/snmpd` que recopila información sobre un sistema mediante un conjunto de bases de información de gestión (MIBs).

```
$ svcs -p net-snmp
STATE      STATE      FMRI
online     17:57:26   svc:/application/management/net-snmp:default
           17:57:26     5022 snmpd
```

EJEMPLO 2-5 Visualización de un reinicio de servicio de contrato automáticamente después de la detención de un proceso

Las instancias de servicio de contrato se reinician automáticamente si el contrato se vacía. SMF también intenta reiniciar los procesos asociados con una instancia de servicio de contrato como parte de la recuperación automática de eventos de fallo de hardware o software. El siguiente ejemplo muestra que después de que se elimina el proceso `/usr/sbin/snmpd`, se reinicia automáticamente con un nuevo ID de proceso. La instancia `net-snmp:default` aún está en línea y tiene una nueva hora de inicio.

```
$ kill 5022
$ svcs -p net-snmp
STATE      STIME      FMRI
online     17:57:59   svc:/application/management/net-snmp:default
           17:57:59   5037 snmpd
```

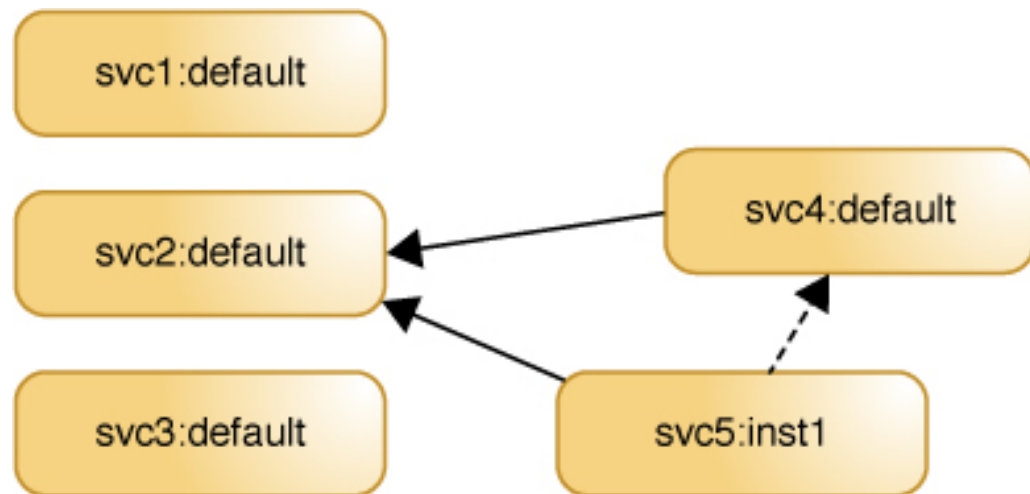
Visualización de información de servicio seleccionado

La salida del comando `svcs` puede ser muy útil para conducir a otros comandos o usar en secuencias de comandos. La opción `-o` del comando `svcs` le permite especificar las columnas de información que desea y el orden de las columnas. Puede colocar en nombre de servicio y el nombre de instancia en columnas distintas, el estado actual y próximo estado del servicio, y el ID de contrato, por ejemplo. Con las opciones `-s` y `-S`, puede especificar el orden de clasificación de la salida para una o más columnas.

Visualización de dependencias de servicio

Las relaciones de dependencia rigen transiciones de estado de instancia de servicio. Consulte [“Dependencias de servicio” \[21\]](#) si desea ver una descripción de alto nivel de las dependencias. Consulte [Capítulo 5, Uso de SMF para controlar la aplicación](#) para obtener descripciones detalladas y cómo especificar diferentes tipos de dependencias.

En la siguiente figura, las instancias de servicio `svc1:default`, `svc2:default` y `svc3:default` no requieren otros servicios o archivos ni otros recursos para iniciarse. Estas instancias pueden iniciarse en paralelo, ejecutar sus métodos de inicio y pasar al estado en línea sin necesidad de esperar cualquier otro recurso. La instancia `svc4:default` no puede ejecutar su método de inicio hasta que la instancia `svc2:default` esté en línea. La instancia `svc5:inst1` necesita recursos `svc2:default` y recursos `svc4:default`. La dependencia que `svc5:inst1` tiene en `svc4:default` es una dependencia opcional y se cumple si `svc4:default` está en uno de los siguientes estados: activado, en línea, desactivado o no está presente. La instancia `svc5:inst1` debe esperar hasta que `svc2:default` esté en línea, y si `svc4:default` está presente y activada, `svc5:inst1` también debe esperar hasta que `svc4:default` esté en línea. Si `svc4:default` está presente y activada, o no está presente, `svc5:inst1` no necesita esperar a `svc4:default`.

FIGURA 2-1 Relaciones de dependencia de servicio

Agrupaciones de dependencia

A cada dependencia se asigna una de las siguientes agrupaciones. La agrupación define cómo se cumplen las dependencias en dicha agrupación.

<code>require_all</code>	Esta dependencia se cumple cuando se cumplen las dos condiciones siguientes: ■ Todas las dependencias de servicio en esta agrupación están en ejecución, ya sea online o degraded. ■ Todas las dependencias de archivos de esta agrupación están presentes.
<code>require_any</code>	Esta dependencia se cumple cuando se cumple una de las condiciones siguientes: ■ Al menos una de las dependencias de servicio en esta agrupación está en ejecución, ya sea online o degraded. ■ Al menos una de las dependencias de archivos en esta agrupación está presente.
<code>optional_all</code>	Esta dependencia se cumple cuando todas las dependencias de servicio de esta agrupación cumplen cualquiera de las siguientes condiciones:

- El servicio está en ejecución, ya sea `online` o `degraded`.
- El servicio requiere una acción administrativa para su ejecución. El servicio no está presente, está incompleto, está `disabled`, está en `maintenance`, o está `offline` esperando dependencias que requieren de una acción administrativa para iniciarse.

Las dependencias de archivos en este grupo pueden estar presentes o no. Esta dependencia no se cumple si la instancia de servicio está en transición y no requiere intervención administrativa para iniciarse. En este caso, el servicio dependiente espera a que esta dependencia se inicie o espera la determinación de que la dependencia no se puede iniciar sin ninguna acción administrativa.

`exclude_all`

Esta dependencia se cumple cuando se cumplen las dos condiciones siguientes:

- Todas las dependencias de servicio en esta agrupación están `disabled`, en `maintenance` o no están presentes.
- Todas las dependencias de archivos de esta agrupación no están presentes.

Enumeración de instancias de las que depende un servicio

El comando `svcs -d` enumera las instancias de servicio de las que depende un servicio determinado.

En este ejemplo se muestran las instancias de servicio de las que depende el servicio `system-repository`:

```
$ svcs -d system-repository
STATE      STIME      FMRI
online     Sep_09     svc:/milestone/network:default
online     Sep_09     svc:/system/filesystem/local:default
online     Sep_09     svc:/system/filesystem/autofs:default
```

El comando `svcs -l` además muestra los servicios de los que depende un servicio determinado. Además del nombre y el estado de la dependencia, la salida de la opción `-l` muestra el tipo, o agrupación, de la dependencia y el valor de la propiedad `restart_on` de la dependencia. En este ejemplo, dos de las dependencias son obligatorias y una es opcional. Consulte [“Agrupaciones de dependencia” \[35\]](#) para descripciones sobre cómo las dependencias en estas agrupaciones afectan el servicio dependiente. Consulte [“Agrupaciones de dependencia” \[35\]](#) para obtener descripciones de cómo diferentes valores de la propiedad `restart_on` de la dependencia afectan el servicio dependiente.

```
$ svcs -l system-repository
fmri          svc:/application/pkg/system-repository:default
name          IPS System Repository
enabled       false
state         disabled
next_state    none
state_time    Mon Sep 09 18:42:28 2013
restarter     svc:/system/svc/restarter:default
manifest      /lib/svc/manifest/application/pkg/pkg-system-repository.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)
```

También puede utilizar el comando `svcpprop` para enumerar estas dependencias. Este formulario muestra las agrupaciones y los valores `restart_on` de la dependencia en líneas separadas, y no muestra el estado de la dependencia.

```
$ svcpprop -g dependency system-repository:default
network/entities fmri svc:/milestone/network:default
network/grouping astring require_all
network/restart_on astring error
network/type astring service
filesystem-local/entities fmri svc:/system/filesystem/local:default
filesystem-local/grouping astring require_all
filesystem-local/restart_on astring none
filesystem-local/type astring service
autofs/entities fmri svc:/system/filesystem/autofs:default
autofs/grouping astring optional_all
autofs/restart_on astring error
autofs/type astring service
```

Enumeración de instancias que dependen de un servicio

El comando `svcs -D` enumera las instancias de servicio que dependen de un servicio determinado.

En este ejemplo se muestran las instancias de servicio que dependen del servicio `system-repository`:

```
$ svcs -D system-repository
STATE      STIME      FMRI
online     16:39:30  svc:/application/pkg/zones-proxyd:default
```

El siguiente comando confirma que `zones-proxyd` depende de `system-repository`.

```
$ svcs -do svc,desc zones-proxyd
SVC          DESC
application/pkg/system-repository IPS System Repository
```

```
system/filesystem/minimal      minimal file system mounts
milestone/network              Network milestone
```

El siguiente comando muestra más información sobre cómo `zones-proxyd` depende de `system-repository`. La última línea de esta salida muestra que el servicio `zones-proxyd` requiere que el servicio `system-repository` esté en ejecución y muestra que `system-repository` está actualmente en ejecución. Esta salida muestra también que el servicio `zones-proxyd` se reiniciará si se refresca el servicio `system-repository`.

```
$ svcs -l zones-proxyd
fmri          svc:/application/pkg/zones-proxyd:default
name          Zones Proxy Daemon
enabled       true
state         online
next_state    none
state_time    January 6, 2014 04:39:30 PM PST
restarter     svc:/system/svc/restarter:default
manifest      /lib/svc/manifest/application/pkg/zoneproxyd.xml
dependency    require_any/none svc:/system/filesystem/minimal (online)
dependency    require_any/error svc:/milestone/network (online)
dependency    require_all/restart svc:/application/pkg/system-repository (online)
```

Visualización de si un servicio se reiniciará automáticamente

Un servicio en ejecución se puede configurar para que se reinicie cuando una de sus dependencias se detenga o refresque. Si las dependencias de un servicio en ejecución (estado `online` o `degraded`) no se cumplen, el servicio pasa al estado `offline`. Si un servicio se reinicia después de que una dependencia se detiene o se refresca, es posible que las dependencias se cumplan nuevamente y que el servicio dependiente pase a un estado en ejecución.

Los siguientes factores determinan si un servicio se reinicia después de que una dependencia `require_all`, `require_any` o `optional_all` se detiene o se refresca:

- Si la dependencia se detuvo o se refrescó. Si se detuvo, ya sea que la dependencia se haya detenido debido a un error como un error de hardware o un volcado de núcleo central, o por algún otro motivo como una acción administrativa.
- El valor del atributo `restart_on` de la dependencia. Los valores posibles son `none`, `error`, `restart` y `refresh`.

Como se muestra en la siguiente tabla, si el valor del atributo `restart_on` de la dependencia es `none`, el servicio dependiente no se reinicia cuando la dependencia se detiene o refresca. Si el valor del atributo `restart_on` de la dependencia es `refresh`, el servicio dependiente siempre se reinicia cuando la dependencia se detiene o refresca. Si el valor de `restart_on` es `error`, el servicio dependiente sólo se reinicia si la dependencia se detuvo debido a un error. Si el valor de `restart_on` es `restart`, el servicio dependiente sólo se reinicia si se refrescó la dependencia.

TABLA 2-1 Reinicio automático de un servicio después de que se detiene una dependencia

Dependencia <code>require_all</code> , <code>require_any</code> o <code>optional_all</code>	Valor de atributo <code>restart_on</code> de dependencia			
Detener o refrescar evento	<code>none</code>	<code>error</code>	<code>restart</code>	<code>refresh</code>
Detener debido a un error	No reiniciar	Reiniciar	No reiniciar	Reiniciar
Otra detención	No reiniciar	No reiniciar	No reiniciar	Reiniciar
Refrescar	No reiniciar	No reiniciar	Reiniciar	Reiniciar

“Enumeración de instancias de las que depende un servicio” [36] muestra que el servicio `system-repository` tiene dos dependencias `require_all` y una dependencia `optional_all`. El siguiente comando muestra que el servicio `system-repository` se reiniciará si el servicio `milestone/network` o el servicio `system/filesystem/autofs` se detiene debido a un error, pero no si ellos se detienen o refrescan por cualquier otro motivo. El servicio `system-repository` no se reiniciará si el servicio `system/filesystem/local` se refrescó o se detuvo por cualquier otro motivo.

```
$ svccfg -s system-repository:default listprop -o propname,propval '*restart_on'
network/restart_on      astring      error
filesystem-local/restart_on astring      none
autofs/restart_on       astring      error
```

Obtención de más información sobre estados de servicio

Sin argumentos, el comando `svcs -x` proporciona información explicativa acerca de las siguientes instancias de servicio:

- Instancias que están activadas, pero no están en ejecución.
- Instancias que evitan que otros servicios activados se ejecuten.

Si todos los servicios activados están en ejecución, el comando `svcs -x` no produce ninguna salida.

En el ejemplo siguiente, el servicio `pkg/depot` está en el estado `maintenance` porque su método de inicio falló.

```
$ svcs -x
svc:/application/pkg/depot:default (IPS Depot)
State: maintenance since September 11, 2013 01:30:42 PM PDT
Reason: Start method exited with $SMF_EXIT_ERR_FATAL.
See: http://support.oracle.com/msg/SMF-8000-KS
See: pkg.depot-config(1M)
See: /var/svc/log/application-pkg-depot:default.log
Impact: This service is not running.
```

La salida sugiere un artículo de conocimientos de reparación automática predictiva de My Oracle Support, una página del comando `man` y un archivo `log` a los que remitirse para

determinar por qué falló el método de inicio. Consulte [“Visualización de archivos log de servicio” \[40\]](#) para obtener información sobre diferentes maneras de ver archivos log. Consulte [“Reparación de una instancia degradada, fuera de línea o en mantenimiento” \[107\]](#) para obtener información sobre cómo reparar un servicio que está en el estado maintenance.

En el siguiente ejemplo, el servicio print/server tiene servicios dependientes que no se están ejecutando. Los servicios dependientes no se puede ejecutar porque el servicio print/server se ha desactivado.

```
$ svcs -x
svc:/application/print/server:default (LP print server)
  State: disabled since Fri Mar 08 14:42:32 2013
Reason: Disabled by an administrator.
  See: http://sun.com/msg/SMF-8000-05
  See: lpsched(1M)
Impact: 2 dependent services are not running. (Use -v for list.)
$ svcs -xv
svc:/application/print/server:default (LP print server)
  State: disabled since Fri Mar 08 14:42:32 2013
Reason: Disabled by an administrator.
  See: http://sun.com/msg/SMF-8000-05
  See: man -M /usr/share/man -s 1M lpsched
Impact: 2 dependent services are not running:
  svc:/application/print/rfc1179:default
  svc:/application/print/ipp-listener:default
$ svcs -D print/server
STATE      STIME      FMRI
online     Mar_08     svc:/milestone/multi-user:default
offline    Mar_08     svc:/application/print/ipp-listener:default
offline    Mar_08     svc:/application/print/rfc1179:default
```

Si un argumento especificado para el comando `svcs -x` no cumple los criterios indicados al principio de esta sección, la salida no muestra ningún motivo para el estado de la instancia, pero aun así muestra los recursos para obtener más información.

```
$ svcs -x smb
svc:/network/smb:default (SMB properties)
  State: online since Thu Sep 12 19:16:56 2013
  See: smb(4)
  See: /var/svc/log/network-smb:default.log
Impact: None.
```

Visualización de archivos log de servicio

SMF registra información sobre acciones de reiniciador significativas, salida estándar de método y salida de errores estándar para `/var/svc/log/service:instance.log` para cada instancia de servicio. Los guiones se reemplazan por barras diagonales en el nombre *service* en el nombre del archivo log. El comando `svcs` con la opción `-L`, `-l` o `-x` muestra el nombre

de ruta completo del archivo log para la instancia de servicio especificada. El comando `svcs -xL` muestra las últimas líneas del archivo log y le indica utilizar el comando `svcs -Lv` para ver todo el archivo log. El comando `svcs -Lv` muestra el archivo completo, lo que puede ser bastante largo. Si prefiere ver el archivo log en un editor o ver sólo las últimas entradas *n*, por ejemplo, realice operaciones en la salida del comando `svcs -L`.

El siguiente ejemplo muestra cómo utilizar el archivo log para investigar por qué el servicio que se muestra en la lista `svcs` está desactivado temporalmente.

```
$ svcs
STATE          STIME      FMRI
legacy_run     Sep_09     lrc:/etc/rc2_d/S47pppd
legacy_run     Sep_09     lrc:/etc/rc2_d/S81dodatadm_udaplt
legacy_run     Sep_09     lrc:/etc/rc2_d/S89PRESERVE
disabled       Sep_09     svc:/system/vbiosd:default
online         Sep_09     svc:/system/early-manifest-import:default
online         Sep_09     svc:/system/svc/restarter:default
...
$ svcs -x vbiosd
svc:/system/vbiosd:default (BIOS call emulation)
  State: disabled since Mon Sep  9 18:42:37 2013
Reason: Temporarily disabled by service method: "vbiosd is not supported on UEFI systems."
  See: http://support.oracle.com/msg/SMF-8000-1S
  See: vbiosd(1M)
  See: /var/svc/log/system-vbiosd:default.log
Impact: This service is not running.
$ svcs -xL vbiosd
svc:/system/vbiosd:default (BIOS call emulation)
  State: disabled since Mon Sep  9 18:42:37 2013
Reason: Temporarily disabled by service method: "vbiosd is not supported on UEFI systems."
  See: http://support.oracle.com/msg/SMF-8000-1S
  See: vbiosd(1M)
  See: /var/svc/log/system-vbiosd:default.log
Impact: This service is not running.
  Log:
[ Sep  9 18:42:27 Enabled. ]
[ Sep  9 18:42:37 Executing start method ("/lib/svc/method/svc-vbiosd start"). ]
[ Sep  9 18:42:37 Method "start" exited with status 101. ]
[ Sep  9 18:42:37 "start" method requested temporary disable: "vbiosd is not supported on UEFI systems"
]

  Use: 'svcs -Lv svc:/system/vbiosd:default' to view the complete log.
$ svcs -L vbiosd
/var/svc/log/system-vbiosd:default.log
$ view `svcs -L vbiosd`
```

Otros archivos log que pueden ser de utilidad incluyen el log para el daemon de reiniciador maestro y el log del sistema. Para ver el nombre del archivo log y ver el archivo log para el daemon de reiniciador `svc.startd`, utilice el nombre de servicio `restarter` con el comando `svcs`. Para ver el archivo log para el daemon log del sistema `syslogd`, utilice el nombre de servicio `system-log`.

Consulte [“Especificación de la cantidad de mensajes de inicio” \[114\]](#) para obtener instrucciones para cambiar la cantidad de mensajes que puede ver en el inicio del sistema. Consulte [“Configuración de notificación de transición de estado y eventos FMA” \[62\]](#) para obtener instrucciones para configurar servicios para notificarle cuándo ellos pasan a un estado de servicio o salen del mismo, o cuándo se produce un evento FMA.

Inspección de configuración de servicio

La configuración de servicio se expresa en las propiedades que se establecen en los servicios y las instancias de servicio y se almacenan en capas en el repositorio de configuración de servicio. Las propiedades que se establecen en un servicio son heredadas por todas las instancias de dicho servicio. Las propiedades que se establecen en una instancia son utilizadas sólo por esa instancia. Las instancias de servicio pueden personalizar los valores de las propiedades heredadas y pueden definir propiedades adicionales que no se han definido para el servicio principal.

Esta sección muestra cómo recuperar los valores de propiedad y cómo identificar si el valor es global para el servicio, es específico para una instancia, se ha entregado con el software, o es una personalización administrativa.

Visualización de descripciones de propiedades y grupos de propiedades

El comando `svccfg describe` muestra una descripción de los grupos de propiedades y propiedades de un servicio, incluidos los valores actuales de propiedades. Sin operandos, `describe` muestra descripciones de todos los grupos de propiedades y las propiedades del servicio o instancia de servicio seleccionados. Utilice la opción `-v` para mostrar más información, incluida una descripción del valor actual y una lista de posibles valores. Utilice la opción `-t` para mostrar información de plantillas.

```
$ svccfg -s pkg/server describe network/restart_on
network/restart_on astring      none
    Determines whether to restart the service due to a dependency refresh, restart, or
    failure.
$ svccfg -s pkg/server describe -v network/restart_on
network/restart_on astring      none
    type: astring
    required: true
    Determines whether to restart the service due to a dependency refresh, restart, or
    failure.
    visibility: readwrite
```

```

    minimum number of values: 1
    maximum number of values: 1
value: none
    value description: Never restart due to dependency refresh, restart, or failure.
value constraints:
    value name: none
    value name: error
    value name: restart
    value name: refresh

```

Visualización de valores de propiedad de instancia y servicio

Los ejemplos de esta sección describen cómo ver propiedades de instancia y servicio, y grupos de propiedades en distintas vistas, capas e instantáneas.

EJEMPLO 2-6 Enumeración de instancias y propiedades heredadas actualmente en uso

De manera predeterminada, el comando `svcprop` muestra los valores asignados a las propiedades en la instantánea en ejecución, que son los valores que se están utilizando actualmente. De manera predeterminada, el comando `svcprop` muestra propiedades en la *vista compuesta* de la instantánea en ejecución, lo que significa que se muestran las propiedades específicas de la instancia y las propiedades heredadas. Si el valor de una propiedad heredada está personalizado en la instancia, se muestra el valor definido en la instancia. La salida muestra una línea para cada propiedad, que muestra el grupo de propiedad y el nombre de propiedad separados por una barra diagonal, el tipo de datos del valor de la propiedad y el valor de la propiedad. Si no se especifica ninguna propiedad o nombre de grupo, se muestran todos los valores de propiedad en la instantánea en ejecución.

Si el operando de patrón o FMRI no especifica una instancia, se muestran sólo las propiedades establecidas en el servicio. No se muestran sólo las propiedades establecidas en una instancia. El siguiente comando muestra propiedades como dependencias de servicio, el tipo de servicio y las rutas de archivos de manifiesto y perfil.

```
$ svcprop svc:/system/identity
```

Al especificar una instancia, consulte la vista compuesta de propiedades personalizadas para esa instancia y las propiedades heredadas del servicio principal. El siguiente comando muestra todas las propiedades de la instantánea en ejecución para la instancia especificada, incluidas las propiedades heredadas del servicio principal y las propiedades específicas para esta instancia. Para propiedades heredadas cuyo valor está personalizado para esta instancia, se muestra el valor personalizado. En este ejemplo se muestran propiedades como dependencias adicionales, la ruta al ejecutable que inicia esta instancia, la ruta al archivo log para esta instancia e información sobre el estado de esta instancia.

```
$ svcprop svc:/system/identity:domain
```

EJEMPLO 2-7 Enumeración de propiedades especificadas o grupos de propiedades actualmente en uso

Utilice la opción `-p` para mostrar propiedades específicas o todas las propiedades en un grupo de propiedades determinado.

```
$ svcprop -p pkg/port pkg/server
svc:/application/pkg/server:oss:properties/pkg/port count 82
svc:/application/pkg/server:s11:properties/pkg/port count 81
svc:/application/pkg/server:default:properties/pkg/port count 80
$ svcprop -p pkg pkg/server:s11
pkg/inst_root astring /export/ipsrepos/Solaris11
pkg/port count 81
...
pkg/ssl_cert_file astring ""
pkg/ssl_key_file astring ""
...
```

EJEMPLO 2-8 Enumeración de valores de instancia y servicios en la vista de edición

Con las opciones, el comando `svcprop` muestra la vista de *edición* de la instantánea en ejecución. La vista de edición muestra los cambios más recientes. Los cambios en la vista de edición pueden o no haberse confirmado en la instantánea en ejecución al refrescar o reiniciar la instancia. Los siguientes comandos muestran la diferencia entre la instantánea en ejecución y la vista de edición. Las instancias `oss` y `s11` se han creado y los valores de propiedad se han cambiado, pero aún no se han refrescado las instancias. El primer comando muestra la vista compuesta de la instantánea en ejecución. Debido a que estas instancias no se han refrescado desde que se personalizaron, los valores que se muestran son los valores del servicio `pkg/server`. La opción `-c` muestra la vista compuesta de los valores de edición. La opción `-C` muestra los valores de edición sin composición. Debido a que esta no es una vista compuesta, no se encuentra ningún valor para la instancia que no se haya personalizado.

```
$ svcprop -p pkg/port pkg/server
svc:/application/pkg/server:oss:properties/pkg/port count 80
svc:/application/pkg/server:s11:properties/pkg/port count 80
svc:/application/pkg/server:default:properties/pkg/port count 80
$ svcprop -c -p pkg/port pkg/server
svc:/application/pkg/server:oss:properties/pkg/port count 82
svc:/application/pkg/server:s11:properties/pkg/port count 81
svc:/application/pkg/server:properties/pkg/port count 80
$ svcprop -C -p pkg/port pkg/server
svc:/application/pkg/server:oss:properties/pkg/port count 82
svc:/application/pkg/server:s11:properties/pkg/port count 81
svcprop: Couldn't find property 'pkg/port' for instance 'svc:/application/pkg/server:default'.
```

El comando `svccfg` muestra los valores de propiedad predeterminados, no los valores en la instantánea en ejecución. Puede forzar `svccfg` para visualizar valores en la instantánea en

ejecución utilizando el subcomando `selectsnap` como se muestra en [“Visualización de valores en una instantánea especificada” \[48\]](#).

El comando `svccfg` sólo muestra valores para el servicio principal cuando especifica un servicio principal y sólo muestra valores para una instancia cuando especifica una instancia. Si no recibe ninguna salida del comando `svccfg listprop`, es posible que la propiedad especificada no esté establecida en el servicio principal o la instancia especificada. Si se suprimió la propiedad, utilice `listcust -M` para ver el valor enmascarado, como se muestra en [“Visualización de personalizaciones de configuración” \[48\]](#).

El siguiente comando enumera todos los valores de propiedad de edición para el servicio especificado porque no se especificó un grupo de propiedades o un nombre de propiedad. Además de la salida mostrada por el comando `svccfg svc:/system/identity`, esta salida incluye nombres de grupo de propiedades y datos de plantilla y tipos.

```
$ svccfg -s svc:/system/identity listprop
```

El siguiente comando enumera todos los valores de propiedad de edición para la instancia de servicio especificada. Debido a que este comando no muestra la vista compuesta, esta salida no muestra las rutas al perfil y los archivos de manifiesto, por ejemplo.

```
$ svccfg -s svc:/system/identity:domain listprop
```

EJEMPLO 2-9 Enumeración de propiedades especificadas o grupos de propiedades en la vista de edición

El siguiente comando muestra todos los valores de propiedad de edición en el grupo de propiedades especificado para la instancia de servicio especificada. La opción `-o` le permite seleccionar las columnas a mostrar. Consulte la página del comando `man svccfg(1M)` para obtener la lista de nombres de columna válidos.

```
$ svccfg -s pkg/server:s11 listprop pkg
pkg                application
pkg/inst_root      astring           /export/ipsrepos/Solaris11
pkg/port           count             81
$ svccfg -s pkg/server:s11 listprop -o propname,value pkg
inst_root          /export/ipsrepos/Solaris11
port               81
```

Visualización de propiedades en un tipo de grupo de propiedades

Además de mostrar los valores de propiedad por nombre de propiedad o nombre de grupo de propiedad, también puede mostrar valores de propiedad por tipo de grupo de propiedades.

EJEMPLO 2-10 Visualización de grupos de propiedades por sus tipos

El subcomando `listpg` del comando `svccfg` muestra el nombre y el tipo de cada grupo de propiedades.

```
$ svccfg -s pkg/server listpg
pkg          application
pkg_bui      application
pkg_secure   application
fs           dependency
autofs       dependency
ntp          dependency
network      dependency
general      framework
manifestfiles framework
start        method
stop         method
tm_common_name template
$ svccfg -s pkg/server:s11 listpg
pkg          application
general      framework
restarter    framework          NONPERSISTENT
```

Los grupos de propiedades no persistentes generalmente conservan un estado de programa activo. Los valores de propiedades en grupos de propiedades no persistentes se borran durante el inicio del sistema.

Especifique un nombre de grupo de propiedades para mostrar el tipo de sólo ese grupo de propiedades.

```
$ svccfg -s pkg/mirror listpg config
config application
```

EJEMPLO 2-11 Enumeración de propiedades de un tipo de grupo de propiedades

Utilice la opción `-g` del comando `svccprop` para mostrar propiedades en un tipo de grupo de propiedades específico. Los tipos de grupos de propiedades incluyen `application`, `dependency`, `method`, `framework` y `template`.

```
$ svccprop -g method pkg/server:s11
start/exec astring %{pkg/pkg_root}/lib/svc/method/svc-pkg-server\ %m
start/timeout_seconds count 0
start/type astring method
stop/exec astring %{pkg/pkg_root}/lib/svc/method/svc-pkg-server\ %m %{restarter/contract}
stop/timeout_seconds count 30
stop/type astring method
$ svccprop -g method -p exec pkg/server:s11
start/exec astring %{pkg/pkg_root}/lib/svc/method/svc-pkg-server\ %m
stop/exec astring %{pkg/pkg_root}/lib/svc/method/svc-pkg-server\ %m %{restarter/contract}
```

Visualización de la capa donde se define un valor

El repositorio de configuración de servicio almacena datos de propiedades en capas según el origen de los datos. Los comandos `svccprop` y `svccfg` pueden mostrar la capa que es el origen de un valor de propiedad. La opción `-l` de los comandos `svccprop` y `svccfg` requiere un argumento para especificar la capa para la que desea información. Los valores de argumento son `manifest`, `system-profile`, `site-profile` y `admin`. La salida indica si un valor de la propiedad determinado fue definido en el manifiesto de servicio, un perfil o por un administrador. Consulte [“Capas de repositorio” \[26\]](#) para obtener descripciones de las capas. La palabra clave `all` es un alias para todas las capas. Si la capa que especifica no es el origen de los valores de propiedad que solicita, no se muestra ninguna salida.

El siguiente comando muestra que algunos valores de propiedad proceden del manifiesto de servicio, algunos fueron definidos por un administrador y algunas propiedades tienen valores en más de una capa. La propiedad `pkg/readonly` tiene un valor definido en el manifiesto de servicio y un administrador también estableció el mismo valor. Los valores de diferentes capas pueden ser diferentes.

```
$ svccprop -l all -p pkg pkg/server:s11
pkg/port count admin 81
pkg/inst_root astring admin /export/ipsrepos/Solaris11
pkg/address net_address manifest
pkg/cfg_file astring manifest ""
...
pkg/readonly boolean manifest true
pkg/readonly boolean admin true
...
```

La opción `-l` del comando `svccfg listprop` también toma el argumento `current`. Al usar `current` como el argumento `-l` se muestran los mismos valores de propiedad que se muestran cuando no utiliza la opción `-l`. La única diferencia en la salida es que el nombre de la capa también se muestra. Los datos no persistentes no muestran un nombre de capa (la tercera columna muestra `<none>`) porque el repositorio de configuración de servicio no almacena información de capa para datos no persistentes. Los grupos de propiedades no persistentes generalmente mantienen un estado de programa activo, y los valores de propiedades en grupos de propiedades no persistentes se borran durante el inicio del sistema.

```
$ svccfg -s pkg/server:s11 listprop -l current
pkg                                application admin
pkg/inst_root                     astring      admin      /export/ipsrepos/Solaris11
pkg/port                          count        admin      81
general                           framework    admin
general/complete                 astring      manifest
general/enabled                  boolean      admin      true
restarter                        framework    <none>      NONPERSISTENT
restarter/logfile                 astring      <none>      /var/svc/log/application-pkg-
server:default.log
restarter/contract                count        <none>      121
restarter/start_pid              count        <none>      1055
```

restarter/start_method_timestamp	time	<none>	1379605275.329096000
restarter/start_method_waitstatus	integer	<none>	0
restarter/auxiliary_state	astring	<none>	dependencies_satisfied
restarter/next_state	astring	<none>	none
restarter/state	astring	<none>	online
restarter/state_timestamp	time	<none>	1379605275.332259000

Visualización de valores en una instantánea especificada

El siguiente comando enumera las instantáneas disponibles para esta instancia de servicio. Utilice estos nombres de instantánea con `svccfg` o `svccfg` para mostrar valores de propiedades que fueron establecidos en esa instantánea. Sólo las instancias tienen instantáneas. Los servicios no tienen instantáneas. Consulte [“Instantáneas del repositorio” \[27\]](#) para obtener más información sobre instantáneas del repositorio de configuración de servicio.

```
$ svccfg -s pkg/server:default listsnap
initial
previous
running
start
$ svccfg -s pkg/server:s11 listsnap
previous
running
start
```

Los comandos siguientes muestran que el valor de la propiedad `pkg/inst_root` era diferente en la instantánea anterior.

```
$ svccfg -s previous -p pkg/inst_root pkg/server:s11
/var/share/pkg/repositories/solaris
$ svccfg -s pkg/server:s11
svc:/application/pkg/server:s11> selectsnap previous
[previous]svc:/application/pkg/server:s11> listprop pkg/inst_root
pkg/inst_root astring /var/share/pkg/repositories/solaris
[previous]svc:/application/pkg/server:s11> exit
```

Visualización de personalizaciones de configuración

El comando `svccfg listcust` muestra las personalizaciones en la capa `admin` para el servicio especificado. Utilice la opción `-L` para mostrar también personalizaciones en la capa `site-profile`. El siguiente comando muestra que las propiedades `pkg/port` y `pkg/inst_root` del servicio `pkg/server` han sido personalizadas por un administrador.


```
$ svccfg -s pkg/server:s11 listcust
pkg                application admin
pkg/port           count      admin      81
pkg/inst_root      astring   admin      /export/ipsrepos/Solaris11
general            framework admin
general/complete  astring   manifest
general/enabled    boolean   admin      true
```

El siguiente comando `svccprop` muestra que la definición de la propiedad `config/nodename` es proporcionada en la capa `manifest`, pero el valor `solaris` está definido en la capa `admin`.

```
$ svccfg -s identity:node describe config/nodename
config/nodename astring      solaris
    Network name of the computer
$ svccprop -p config/nodename -l all svc:/system/identity:node
config/nodename astring manifest ""
config/nodename astring admin solaris
```

El comando `svccfg listcust` también muestra todas las entidades *enmascaradas*. Utilice la opción `-M` para mostrar sólo entidades enmascaradas. Antes de utilizar el comando `svccfg delcust`, utilice el comando `svccfg listcust` para verificar los elementos que se van a suprimir. Consulte “[Supresión de grupos de propiedades, propiedades y valores de propiedad](#)” [74] y la página del comando `man smf(5)` para obtener una descripción de las entidades enmascaradas.

Visualización de parámetros de notificación de eventos

El comando `svcs -n` muestra los parámetros de notificación de eventos FMA, los parámetros de notificación de transición de estado SMF y los parámetros de notificación de transición de estado de instancia de servicios. Consulte “Notification Parameters” (Parámetros de notificación) en la página del comando `man smf(5)` para obtener más información sobre estos parámetros.

```
$ svcs -n
Notification parameters for FMA Events
  Event: problem-diagnosed
    Notification Type: smtp
      Active: true
      reply-to: root@localhost
      to: root@localhost

    Notification Type: snmp
      Active: true

    Notification Type: syslog
      Active: true
```

```
Event: problem-repaired
Notification Type: snmp
Active: true

Event: problem-resolved
Notification Type: snmp
Active: true

System wide notification parameters:
svc:/system/svc/global:default:
Event: to-maintenance
Notification Type: smtp
Active: true
to: sysadmins@example.com

svc:/application/pkg/mirror:default:
Event: to-maintenance
Notification Type: smtp
Active: true
to: installteam@example.com
```

Se muestran tres eventos FMA: problem-diagnosed, problem-repaired y problem-resolved. Los parámetros de notificación también pueden configurarse para un cuarto evento: problem-updated.

Para la configuración de notificación de transición de estado por todo el sistema, también se menciona el servicio que almacena estos valores de configuración globales. Esta configuración en todo el sistema es una configuración personalizada. Los valores de todo el sistema, o globales, se aplican a todas las instancias de servicio que no tienen valores personalizados establecidos.

La última configuración mostrada es una configuración personalizada para una instancia de servicio en particular.

Utilice el comando `svccfg listnotify` para mostrar los parámetros de notificación sólo para el evento especificado. Para eventos de transición de estado, utilice la opción `-g` para mostrar valores de configuración globales. La salida muestra también el origen de los valores de los parámetros de notificación.

```
$ svccfg listnotify problem-resolved
Event: problem-resolved (source: svc:/system/fm/notify-params:default)
Notification Type: snmp
Active: true
$ svccfg listnotify -g to-maintenance
Event: to-maintenance (source: svc:/system/svc/global:default)
Notification Type: smtp
Active: true
to: sysadmins@example.com
$ svccfg -s pkg/mirror listnotify to-maintenance
Event: to-maintenance (source: svc:/application/pkg/mirror)
Notification Type: smtp
Active: true
```

to: installteam@example.com

Consulte [“Configuración de notificación de transición de estado y eventos FMA” \[62\]](#) para obtener más información sobre la configuración de notificación de eventos.

3

♦ ♦ ♦ C A P Í T U L O 3

Administración de servicios

En este capítulo se describe cómo iniciar, detener y reiniciar un servicio y cómo volver a leer la configuración de servicio. En este capítulo también se describe cómo configurar el sistema para recibir notificación de eventos FMA o transiciones de estado de servicio. Estos cambios son personalizaciones de nivel admin como se describe en [“Capas de repositorio” \[26\]](#).

El comando que cambia el estado del servicio es `svcadm`. El comando `svcadm` opera en una instancia de servicio. Si proporciona un nombre de servicio sin ninguna instancia especificada y el servicio tiene sólo una única instancia, `svcadm` opera en esa instancia. Si proporciona un nombre de servicio sin ninguna instancia especificada y el servicio tiene varias instancias, o si especifica otro patrón que coincide con varias instancias, `svcadm` emite un mensaje de error.

Gestión de instancias de servicio SMF

Una instancia de servicio está siempre en uno de los estados descritos en [“Estados de servicio” \[21\]](#). En esta sección se discute cómo hacer que una instancia pase a un estado diferente, cómo confirmar valores de propiedad actualizados para la instantánea en ejecución y cómo suprimir instancias de la vista normal.

Inicio de un servicio

Una instancia de servicio que está en cualquiera de los siguientes estados ya está activada y no necesita iniciarse: `degraded`, `maintenance`, `offline`, `online`. Si la instancia que desea iniciar está en el estado `degraded`, `maintenance` o `offline`, consulte [“Reparación de una instancia degradada, fuera de línea o en mantenimiento” \[107\]](#). Si la instancia que desea iniciar está en el estado `disabled`, active la instancia como se muestra en el siguiente procedimiento. Cuando activa una instancia, el reiniciador para esa instancia intenta realizar la transición de la instancia al estado `online`.

▼ Cómo activar una instancia de servicio

1. Compruebe el estado de la instancia y las dependencias.

Compruebe que la instancia esté desactivada y que todas sus dependencias necesarias estén en ejecución (en el estado online o degraded).

```
$ svcs -l FMRI
```

2. Active la instancia.

El reiniciador para el servicio intenta llevar la instancia especificada al estado online.

Una instancia puede activarse de manera temporal o permanente. La activación permanente persiste durante el reinicio del sistema y es el valor predeterminado. La activación temporal dura sólo hasta el reinicio.

■ Active permanentemente la instancia.

```
$ svcadm enable FMRI
```

■ Active temporalmente la instancia.

Utilice la opción -t para especificar una activación temporal.

```
$ svcadm enable -t FMRI
```

Si desea que una instancia se ejecute ahora pero no se ejecute en el siguiente reinicio, asegúrese de que la instancia esté desactivada y, a continuación, active temporalmente la instancia. Para verificar que la instancia esté activada temporalmente, utilice el comando `svcs -l` y compruebe la fila enabled:

```
enabled      true (temporary)
```

■ Active la instancia de manera síncrona.

Si especifica la opción -s, `svcadm` activa la instancia y espera que la instancia ingrese al estado online o degraded antes de volver. El comando `svcadm` vuelve cuando la instancia alcanza un estado en línea o cuando determina que la instancia requiere que intervenga el administrador para alcanzar un estado en línea.

Utilice la opción -T con la opción -s para especificar un límite superior en segundos para realizar la transición o determinar que no se puede realizar la transición.

```
$ svcadm enable -sT 10 FMRI
```

3. Verifique que la instancia esté en línea.

```
$ svcs FMRI
```

Si la instancia está en el estado degraded, maintenance o offline, consulte [“Reparación de una instancia degradada, fuera de línea o en mantenimiento” \[107\]](#).

ejemplo 3-1 Activación de una instancia de servicio permanentemente

El siguiente comando muestra que la instancia de servicio pkg/mirror:default está actualmente desactivada y todas sus dependencias necesarias están en línea.

```
$ svcs -l pkg/mirror
fmri          svc:/application/pkg/mirror:default
name          IPS Repository Mirror
enabled       false
state         disabled
next_state    none
state_time    September 17, 2013 07:16:52 AM PDT
restarter     svc:/system/svc/restarter:default
manifest      /lib/svc/manifest/application/pkg/pkg-mirror.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)
dependency    require_all/none svc:/application/pkg/repositories-setup (online)
```

El siguiente comando activa la instancia pkg/mirror:default. En este caso, el comando svcadm vuelve porque la instancia pkg/mirror:default se ha activado correctamente.

```
$ svcadm enable -sT 10 pkg/mirror:default
$ svcs pkg/mirror
STATE      STIME      FMRI
online     22:03:53   svc:/application/pkg/mirror:default
```

ejemplo 3-2 Activación de una instancia de servicio temporalmente

El siguiente comando muestra que la instancia de servicio net-snmp:default está actualmente desactivada y todas sus dependencias necesarias están en línea. La dependencia que está desactivada es una dependencia opcional.

```
$ svcs -l net-snmp
fmri          svc:/application/management/net-snmp:default
name          net-snmp SNMP daemon
enabled       false
state         disabled
next_state    none
state_time    September 17, 2013 05:56:39 PM PDT
logfile       /var/svc/log/application-management-net-snmp:default.log
restarter     svc:/system/svc/restarter:default
contract_id
manifest      /etc/svc/profile/generic.xml
manifest      /lib/svc/manifest/application/management/net-snmp.xml
dependency    require_all/none svc:/system/filesystem/local (online)
dependency    optional_all/none svc:/milestone/name-services (online)
```

```

dependency optional_all/none svc:/system/system-log (online)
dependency optional_all/none svc:/network/rpc/rstat (disabled)
dependency require_all/restart svc:/system/cryptosvc (online)
dependency require_all/restart svc:/milestone/network (online)
dependency require_all/refresh file://localhost/etc/net-snmp/snmp/snmpd.conf (online)
dependency require_all/none svc:/milestone/multi-user (online)

```

Después de activar la instancia con la opción `-t` como se muestra en el siguiente ejemplo, la instancia está activada temporalmente, está en línea y tiene un ID de contrato porque ha iniciado el proceso `snmpd`, como muestra el comando `svcs -p`.

```

$ svcadm enable -t net-snmp:default
$ svcs -l net-snmp
fmri          svc:/application/management/net-snmp:default
name          net-snmp SNMP daemon
enabled       true (temporary)
state         online
next_state    none
state_time    September 17, 2013 05:57:26 PM PDT
logfile       /var/svc/log/application-management-net-snmp:default.log
restarter     svc:/system/svc/restarter:default
contract_id   160
manifest      /etc/svc/profile/generic.xml
manifest      /lib/svc/manifest/application/management/net-snmp.xml
dependency    require_all/none svc:/system/filesystem/local (online)
dependency    optional_all/none svc:/milestone/name-services (online)
dependency    optional_all/none svc:/system/system-log (online)
dependency    optional_all/none svc:/network/rpc/rstat (disabled)
dependency    require_all/restart svc:/system/cryptosvc (online)
dependency    require_all/restart svc:/milestone/network (online)
dependency    require_all/refresh file://localhost/etc/net-snmp/snmp/snmpd.conf (online)
dependency    require_all/none svc:/milestone/multi-user (online)
$ svcs -p net-snmp
STATE      STIME      FMRI
online     17:57:26  svc:/application/management/net-snmp:default
           17:57:26      5022 snmpd

```

Detención de un servicio

Utilice el comando `svcadm disable` para desactivar una instancia activada o temporalmente desactivada. Una instancia desactivada no se puede reiniciar. Primero debe activar la instancia.

▼ Cómo desactivar una instancia de servicio

1. Compruebe si otros servicios dependen de esta instancia.
 - a. Enumere los servicios que dependen de esta instancia.


```
$ svcs -D FMRI
```

b. Compruebe si el servicio dependiente requiere esta instancia.

Para cada resultado del comando `svcs -D`, utilice el comando `svcs -l` para comprobar si la dependencia es una dependencia necesaria.

No debe desactivar esta instancia si esta instancia es una dependencia necesaria de otro servicio.

2. Desactive la instancia.

El reiniciador para el servicio intenta llevar la instancia especificada al estado `disabled`.

Una instancia puede desactivarse de manera temporal o permanente. La desactivación permanente persiste durante el reinicio del sistema y es el valor predeterminado. La desactivación temporal dura sólo hasta el reinicio.

■ **Desactive permanentemente la instancia.**

```
$ svcadm disable FMRI
```

■ **Desactive temporalmente la instancia.**

Utilice la opción `-t` para especificar una desactivación temporal.

```
$ svcadm disable -t FMRI
```

Si desea que una instancia se desactive ahora pero se ejecute en el siguiente reinicio, asegúrese de que la instancia esté en ejecución (en el estado `online` o `degraded`) y, a continuación, desactive temporalmente la instancia. Para verificar que la instancia esté desactivada temporalmente, utilice el comando `svcs -l` y compruebe la fila `enabled`:

```
enabled      false (temporary)
```

■ **Desactive la instancia de manera síncrona.**

Si especifica la opción `-s`, `svcadm` desactiva la instancia y espera que la instancia ingrese al estado `disabled` antes de volver. El comando `svcadm` vuelve cuando la instancia alcanza el estado `disabled` o cuando determina que la instancia requiere que intervenga el administrador para alcanzar el estado `disabled`.

Utilice la opción `-T` con la opción `-s` para especificar un límite superior en segundos para realizar la transición o determinar que no se puede realizar la transición.

```
$ svcadm disable -sT 10 FMRI
```

3. Verifique que la instancia esté desactivada.

```
$ svcs FMRI
```

ejemplo 3-3 Desactivación de una instancia de servicio

En este ejemplo se muestra que la instancia de servicio `pkg/update:default` está inicialmente en línea y no hay otros servicios que dependan de esta instancia. El comando `svcadm disable` se ejecuta correctamente, la instancia está actualmente en el estado `disabled` y el intento de reinicio falla.

```
$ svcs pkg/update
STATE      STIME      FMRI
online      7:18:17    svc:/application/pkg/update:default
$ svcs -D pkg/update:default
STATE      STIME      FMRI
$ svcadm disable pkg/update
$ svcs pkg/update
STATE      STIME      FMRI
disabled    22:51:12    svc:/application/pkg/update:default
$ svcadm restart pkg/update:default
$ svcs pkg/update
STATE      STIME      FMRI
disabled    22:51:12    svc:/application/pkg/update:default
```

Reinicio de un servicio

La operación de reinicio sólo reinicia las instancias que están actualmente en ejecución (en el estado `online` o `degraded`). Puede que deba reiniciar una instancia en ejecución porque ha realizado un cambio de configuración que no se puede hacer efectivo mientras se esté ejecutando la instancia, por ejemplo.

Reiniciar una instancia de servicio no refresca la configuración. El comando `svcadm restart` ejecuta el método de detención de la instancia y, a continuación, ejecuta el método de inicio de la instancia. El comando `svcadm restart` no confirma los cambios de propiedad en la instantánea en ejecución y no ejecuta el método de refrescamiento de la instancia. Consulte [“Relectura de configuración de servicio” \[59\]](#) para obtener información sobre la confirmación de cambios de configuración en la instantánea en ejecución.

Reiniciar el servicio `manifest-import` es un caso especial. Reiniciar el servicio `manifest-import` importa cualquier manifiesto o perfil modificado en ubicaciones estándar, confirma los cambios en el repositorio de configuración de servicio, toma una nueva instantánea en ejecución y ejecuta el método de refrescamiento de instantáneas modificadas si existe un método de refrescamiento.

▼ Cómo reiniciar una instancia de servicio

1. Compruebe el estado de la instancia.

La instancia debe estar en el estado `online` o `degraded`.

```
$ svcs FMRI
```

2. Reinicie la instancia.

El reiniciador para el servicio intenta llevar la instancia especificada al estado online. La mayoría de los reiniciadores implementan la operación de reinicio como una operación de detención seguida de una operación de inicio.

■ Reinicie la instancia.

```
$ svcadm restart FMRI
```

■ Reinicie la instancia de manera síncrona.

Si especifica la opción `-s`, `svcadm` reinicia la instancia y espera que la instancia ingrese al estado `online`, `degraded` o `maintenance` antes de volver. El comando `svcadm` vuelve cuando la instancia alcanza uno de estos estados o cuando determina que la instancia requiere que intervenga el administrador para alcanzar uno de estos estados.

Utilice la opción `-T` con la opción `-s` para especificar un límite superior en segundos para realizar la transición o determinar que no se puede realizar la transición.

```
$ svcadm restart -sT 10 FMRI
```

3. Verifique que la instancia se haya iniciado.

Si el reinicio se realizó correctamente, la instancia está en estado `online`, `degraded` o `maintenance`. Si la instancia está en el estado `degraded` o `maintenance`, consulte [“Reparación de una instancia degradada, fuera de línea o en mantenimiento” \[107\]](#).

```
$ svcs FMRI
```

Relectura de configuración de servicio

Al cambiar la configuración del servicio, los cambios no aparecen inmediatamente en la instantánea en ejecución. Esos cambios se almacenan en el repositorio de configuración de servicio como valores de propiedad de edición o actuales. La operación de refrescamiento actualiza la instantánea en ejecución de la instancia de servicio especificada con los valores de la configuración de edición.

Los comandos `svcadm refresh` y `svccfg refresh` realizan los siguientes pasos:

1. Crear una nueva instantánea en ejecución para confirmar las propiedades de edición en la instantánea en ejecución.
2. Ejecutar el método de refrescamiento de la instancia, si existe un método de refrescamiento y la instancia está en estado `online` o `degraded`. El método de refrescamiento debe notificar

a la aplicación que se han realizado cambios. El método de refrescamiento puede volver a leer los valores de propiedad de la instantánea en ejecución. Incluso si no existe un método de refrescamiento, se actualiza la configuración en la instantánea en ejecución.

El comando `svcadm refresh` opera en una instancia de servicio. El comando `svccfg refresh` opera en una instancia de servicio o en un servicio principal. Si se especifica un servicio, el comando `svccfg refresh` refresca todas las instancias de ese servicio. Si bien se toman instantáneas sólo para las instancias de servicio y no para los servicios principales, las propiedades de servicio principal son heredadas por las instancias de servicio. Las propiedades de servicio principal modificadas aparecen en una instantánea de instancia de servicio si la instancia no sobrescribe esos cambios.

Algunos cambios, como cambios de dependencia, se aplican inmediatamente. Otros cambios no se hacen efectivos hasta que el servicio se reinicia como se describe en [“Reinicio de un servicio” \[58\]](#). Los cambios que no se pueden realizar mientras se está ejecutando la aplicación necesitan un refrescamiento seguido de un reinicio. Algunos ejemplos de los cambios que no se pueden realizar mientras se está ejecutando la aplicación incluyen: cerrar o abrir un socket o restablecer una variable de entorno.

Si especifica la opción `-s` con el comando `svcadm refresh`, `svcadm` refresca la instancia y espera que la instancia ingrese al estado `online`, `degraded` o `maintenance` antes de volver. El comando `svcadm` vuelve cuando la instancia alcanza uno de estos estados o cuando determina que la instancia requiere que intervenga el administrador para alcanzar uno de estos estados. Utilice la opción `-T` con la opción `-s` para especificar un límite superior en segundos para realizar la transición o determinar que no se puede realizar la transición.

Supresión de un servicio

El comando `svccfg delete` no elimina una instancia de servicio del sistema. En su lugar, el comando `svccfg delete` enmascara la instancia. Después de ejecutar el comando `svccfg delete`, el manifiesto de servicio sigue existiendo en `/lib/svc/manifest`. SMF mantiene el repositorio de configuración de servicio en sincronización con el contenido del sistema de archivos. Ya que el manifiesto aún existe en el sistema de archivos en una ubicación estándar, esa información de servicio aún se almacena en el repositorio y sólo es enmascarada de la vista normal. Las personalizaciones administrativas se suprimen de una instancia enmascarada. Consulte la página del comando `man smf(5)` para obtener una descripción de entidades enmascaradas.

Los archivos que admiten una instancia de servicio se actualizan cuando se utilizan comandos `pkg`, incluso si esa instancia de servicio está enmascarada. Cuando los archivos que admiten una instancia de servicio se actualizan mediante comandos `pkg`, el almacén de datos SMF se actualiza, aunque el servicio sigue enmascarado de la vista. Si la instancia de servicio no está enmascarada, la instancia de servicio ya está actualizada desde los archivos enviados por `pkg`.

sin necesidad de ninguna intervención. Para desenmascarar una instancia de servicio, consulte [Cómo deshacer la supresión de una instancia de servicio \[61\]](#).

▼ Cómo suprimir una instancia de servicio

1. Compruebe los dependientes de la instancia que desea suprimir.

Utilice el comando `svcs -D` para mostrar las instancias que dependen de esta instancia. Después de suprimir esta instancia, es posible que no pueda ejecutar instancias dependientes. Utilice el comando `svcs -l` para comprobar si esta instancia es una dependencia necesaria de la instancia dependiente.

2. Enmascare la instancia.

Utilice el comando `svccfg delete` para enmascarar la instancia de la vista normal. Utilice el comando `svcs` para mostrar el estado de la instancia. Si la instancia está en ejecución (está en el estado `online` o `degraded`), utilice el comando `svccfg delete -f` para enmascarar la instancia de la vista normal.

```
$ svcs -H my-svc
disabled          7:25:37 svc:/site/my-svc:default
$ svccfg delete svc:/site/my-svc:default
```

3. Verifique que la instancia esté enmascarada.

Utilice el comando `svccfg listcust -M` para confirmar que la instancia está enmascarada. Los comandos como `svcs` deben mostrar un mensaje de error de que no se ha encontrado una instancia coincidente.

```
$ svccfg listcust -M
svc:/site/my-svc:default manifest MASKED
  general          admin    MASKED
  general/complete astring  admin    MASKED
  general/enabled  boolean  admin    MASKED true
$ svcs -H my-svc
svcs: Pattern 'my-svc' doesn't match any instances
```

▼ Cómo deshacer la supresión de una instancia de servicio

1. Confirme que la instancia esté enmascarada.

Utilice el comando `svccfg listcust -M` como se muestra en el procedimiento anterior.

2. Desenmascare la instancia.

```
$ svccfg -s svc:/site/my-svc:default delcust
Deleting customizations for instance: default
```

Volver a importar el manifiesto no elimina una máscara.

3. Verifique que la instancia esté desenmascarada.

Utilice el comando `svccfg listcust -M` para confirmar que la instancia no está enmascarada. El comando `svcs` debe mostrar el estado de la instancia.

Configuración de notificación de transición de estado y eventos FMA

Puede configurar el sistema para que le notifique cuando un servicio cambia de estado o cuando se produce un evento FMA. Puede especificar notificación SMTP (protocolo simple de transferencia de correo) o SNMP (protocolo simple de administración de redes).

De manera predeterminada, las capturas SNMP se envían en transiciones de mantenimiento. Si utiliza SNMP para la notificación de transición, puede configurar más capturas para otras transiciones de estado.

Los siguientes ejemplos muestran cómo configurar parámetros de notificación para eventos SMF y FMA, y cómo suprimir parámetros de notificación.

EJEMPLO 3-4 Configuración de una notificación global para un evento de estado de servicio

El siguiente comando crea una notificación que envía un correo electrónico cuando los servicios pasan al estado `maintenance`.

```
$ svccfg setnotify -g to-maintenance mailto:sysadmins@example.com
```

-g La opción `-g` establece este parámetro de notificación para todas las instancias de servicio que no tienen valores personalizados establecidos. Todas las instancias de servicio modificadas se refrescan. La opción `-g` sólo puede ser utilizada cuando configura una notificación para transiciones de estado de servicio, no con eventos FMA.

to-maintenance El argumento `to-maintenance` es un evento de transición de estado descrito en “Parámetros de notificación” en la página del comando `man smf(5)`. La especificación del nombre de estado incluye transiciones `to-state` y `from-state`. Este evento también podría ser una lista separada por comas de transiciones.

mailto: El argumento `mailto` especifica la notificación que desea recibir para el evento especificado. Este argumento puede también especificar `snmp`. Un valor de notificación `snmp` debe ser `snmp:active` o `snmp:inactive`. Un valor de notificación `mailto` puede ser `mailto:active` o `mailto:inactive`, además de la forma mostrada en este ejemplo.

Establecer un parámetro de notificación sobrescribe cualquier valor existente para ese evento. La configuración `active` e `inactive` no sobrescribe valores existentes pero se alterna si la notificación existente está activa para el evento especificado.

EJEMPLO 3-5 Configuración de una notificación para una instancia de servicio especificada

El siguiente comando crea una notificación que envía correo electrónico cuando el servicio `pkg/mirror` pasa al estado `maintenance`.

```
$ svccfg -s pkg/mirror setnotify to-maintenance mailto:installteam@example.com
```

El siguiente comando crea una notificación que envía correo electrónico cuando el servicio `http:apache22` sale del estado `online`.

```
$ svccfg -s http:apache22 setnotify from-online mailto:webservices@example.com
```

EJEMPLO 3-6 Configuración de una notificación para un evento FMA

El argumento `problem-diagnosed` es un evento FMA. Este argumento puede ser una lista separada por comas de eventos FMA. Consulte la lista de eventos FMA en “Parámetros de notificación” en la página del comando `man smf(5)`.

```
$ svccfg setnotify problem-diagnosed mailto:IT@example.com
```

EJEMPLO 3-7 Supresión de valores de notificación

Los siguientes comandos suprimen la configuración de notificación definida en los ejemplos anteriores.

```
$ svccfg delnotify -g to-maintenance
$ svccfg -s pkg/mirror delnotify to-maintenance
$ svccfg setnotify problem-diagnosed mailto:root@localhost
```


Configuración de servicios

SMF almacena datos de configuración en el repositorio de configuración de servicio. Configurar servicios SMF significa modificar los datos en el repositorio de configuración y, luego, confirmar las modificaciones en la instantánea en ejecución. En este capítulo se describe cómo modificar los datos en el repositorio de configuración. Para ver los datos en el repositorio de configuración, consulte [“Inspección de configuración de servicio” \[42\]](#). Para confirmar modificaciones de configuración en instantáneas en ejecución, consulte [“Relectura de configuración de servicio” \[59\]](#).

Cada servicio e instancia de servicio almacena datos de configuración en propiedades, que se organizan en grupos de propiedades. La modificación de los datos en el repositorio de configuración incluye la modificación de valores de propiedad de servicio, la creación de propiedades y grupos de propiedades personalizados, la creación de instancias de un servicio y la aplicación de un perfil. La modificación de la configuración también incluye la supresión de las personalizaciones y reversión de instantáneas de repositorio.

En este capítulo, también se muestra cómo modificar un servicio `inetd`.

Los cambios de configuración de SMF se pueden registrar mediante la estructura de auditoría de Oracle Solaris. Consulte [“Configuring the Audit Service \(Task Map\)”](#) (Configuración de servicio de auditoría [Mapa de tarea]) en *Gestión de auditorías en Oracle Solaris 11.2* para obtener más información.

Uso del comando de configuración de servicio

El comando `svccfg` manipula datos en el repositorio de configuración de servicio. Los cambios realizados con el comando `svccfg` se registran en la capa `admin`. Consulte [“Capas de repositorio” \[26\]](#) para obtener información sobre capas. Los cambios realizados con el comando `svccfg` se almacenan en el repositorio de configuración de servicio como valores de propiedad de edición o actuales y no aparecen inmediatamente en la instantánea en ejecución. Al cambiar los datos de configuración, utilice el comando `svcadm refresh` o el comando `svccfg refresh` para confirmar los nuevos valores para la instantánea en ejecución.

Mantener datos modificados recientemente separados de la instantánea en ejecución le permite hacer varios cambios y, luego, confirmar todos los cambios en la instantánea en ejecución.

Mientras está en el proceso de realizar varios cambios, algunos valores de propiedad pueden ser incompatibles o inconsistentes, pero no se modifica la instantánea en ejecución. Cuando termine de realizar cambios, realice un refrescamiento.

Puede usar el comando `svccfg` en cualquiera de las siguientes maneras:

- Utilice el comando `svccfg editprop` para llamar un editor en los grupos de propiedades y las propiedades de la entidad seleccionada actualmente.
- Introduzca un comando `svccfg` completo en la línea de comandos, especificando subcomandos como `setprop`.
- Introduzca sólo `svccfg` o `svccfg -s FMRI` en la línea de comandos para iniciar una sesión interactiva.
- Especifique la opción `-f` para leer los comandos `svccfg` de un archivo.

Invocación de un editor de propiedades

Invocar el comando `svccfg` como se muestra en el siguiente ejemplo abre un editor en las propiedades de la entidad seleccionada. Esta forma del comando `svccfg` puede ser muy rápida y cómoda para modificar varios valores de propiedad. Para el subcomando `editprop`, debe especificar una entidad con la opción `-s`.

```
$ svccfg -s pkg/server:s11 editprop
```

Un archivo de comandos `setprop` para los valores actuales de cada propiedad de la entidad especificada se abre en el editor especificado por la variable de entorno `VISUAL`. Si `VISUAL` no está definido, se abre el editor especificado por `EDITOR`. Si `VISUAL` o `EDITOR` no están definidos, el archivo de propiedad se abre en `vi`.

Cada línea del archivo está precedido por un carácter de comentario. Para cambiar el valor de una propiedad en la configuración de edición `svccfg`, elimine el carácter de comentario, cambie el valor y guarde el archivo. Para cambiar el valor de una propiedad en la instantánea en ejecución, elimine el carácter de comentario de la última línea del archivo, que es el subcomando `refresh`.

La lista siguiente muestra un ejemplo parcial de un archivo creado por el subcomando `editprop`:

```
##
## Change property values by removing the leading '#' from the
## appropriate lines and editing the values. svccfg subcommands
## such as delprop can also be added to the script.
##
## Property group "pkg"
## The following properties are defined in the selected instance
```

```
## (svc:/application/pkg/server:s11)

# setprop pkg/port = count: 81
# setprop pkg/inst_root = astring: /export/ipsrepos/Solaris11

## The following properties inherit from the parent service
## (svc:/application/pkg/server)

# ...

## Property group "pkg_bui"

# ...

## Property group "pkg_secure"

# ...

## Uncomment to apply these changes to this instance.
# refresh
```

Como las instrucciones en el estado de archivo, puede agregar subcomandos distintos de `setprop`. Por ejemplo, puede agregar un comando `delprop`. Algunos grupos de propiedades, como `framework` y `dependency`, no se muestran de manera predeterminada. Especifique `editprop -a` para mostrar todas las propiedades.

Los comandos sin comentarios en este archivo temporal son ejecutados cuando guarda y sale de la sesión de edición.

Invocación de `svccfg` de manera interactiva o con un archivo

Invocar el comando `svccfg` de manera interactiva como se muestra en el siguiente ejemplo puede ser conveniente cuando desea realizar varias operaciones de configuración.

```
$ svccfg
svc:> select pkg/server
svc:/application/pkg/server> list
:properties
default
svc:/application/pkg/server> add s11
svc:/application/pkg/server> select s11
svc:/application/pkg/server:s11> setprop pkg/inst_root=/export/ipsrepos/Solaris11
svc:/application/pkg/server:s11> setprop pkg/port=81
svc:/application/pkg/server:s11> unselect
svc:/application/pkg/server> add oss
svc:/application/pkg/server> select oss
svc:/application/pkg/server:oss> setprop pkg/inst_root=/export/ipsrepos/SolarisStudio
```

```
svc:/application/pkg/server:oss> setprop pkg/port=82
svc:/application/pkg/server:oss> unselect
svc:/application/pkg/server> list
:properties
default
s11
oss
svc:/application/pkg/server> refresh
svc:/application/pkg/server> select pkg/mirror:default
svc:/application/pkg/mirror:default> listprop config/crontab_period
config/crontab_period astring      "30 2 25 * *"
svc:/application/pkg/mirror:default> setprop config/crontab_period="00 3 25 * *"
svc:/application/pkg/mirror:default> refresh
svc:/application/pkg/mirror:default> exit
$
```

Los mismos comandos proporcionados en las solicitudes interactivas en el ejemplo anterior también pueden proporcionarse en un archivo y ejecutarse con un comando como el siguiente.

```
$ svccfg -f cfgpkgrepos
```

Configuración de valores de propiedad

Los siguientes comandos establecen valores de propiedad:

```
svccfg setprop
```

Cambia el valor de una propiedad.

```
svccfg addpropvalue
```

Agrega un valor a una propiedad de varios valores.

```
svccfg setenv
```

Cambia el valor de una variable de entorno para un entorno de ejecución de proceso de servicio.

No olvide utilizar el comando `svccfg refresh` o `svcadm refresh` para confirmar cambios de configuración en la instantánea en ejecución.

EJEMPLO 4-1 Configuración de un valor simple

En el uso más simple de `setprop`, especifique un *pg/name* para el servicio o instantánea seleccionados, donde *pg* es el nombre del grupo de propiedades y *name* es el nombre de la propiedad, y especifique el nuevo valor después de un símbolo de igual. Si la propiedad ya existe o está en una plantilla, no necesita especificar el tipo de propiedad.

```
$ svccfg -s pkg/server:s11 setprop pkg/port=81
```

EJEMPLO 4-2 Configuración de un valor que contiene un carácter de dos puntos

Si el valor de la propiedad contiene un carácter dos puntos (:), entonces especifique el tipo de propiedad como se muestra en el siguiente ejemplo donde el tipo es `astring`:

```
$ svccfg -s system-repository:default setprop config/http_proxy = astring:
https://proxyURI
```

Utilice el subcomando `listprop` para buscar el tipo de la propiedad que desea establecer.

```
$ svccfg -s system-repository:default listprop config/http_proxy
config/http_proxy astring
```

EJEMPLO 4-3 Configuración de un valor que contiene espacios

Utilice comillas dobles para definir un valor que contiene espacios. En función de su shell, es posible que tenga que poner la cadena entre comillas dobles en comillas simples.

```
$ svccfg -s pkg/mirror setprop config/crontab_period = "00 3 25 * *"
$ svccfg -s pkg/mirror setprop config/crontab_period = '"00 3 25 * *"'
```

Utilice comillas para definir un valor que contiene comillas dobles o caracteres de barra invertida y utilice un carácter de barra diagonal inversa como escape de comillas dobles o caracteres de barra invertida.

EJEMPLO 4-4 Configuración de un valor que es un conjunto de valores

Utilice paréntesis para especificar un conjunto de valores como un valor único. En función de su shell, es posible que tenga que poner el valor definido entre comillas simples.

```
$ svccfg -s dns/client setprop config/nameserver = (10.0.0.1 192.168.0.1)
$ svccfg -s dns/client setprop config/nameserver = '(10.0.0.1 192.168.0.1)'
$ svccfg -s dns/client listprop config/nameserver
config/nameserver net_address 10.0.0.1 192.168.0.1
```

Utilice el subcomando `describe` para buscar el número de valores permitidos en el conjunto de valores.

```
$ svccfg -s dns/client describe -v config/nameserver
config/nameserver net_address 10.0.0.1 192.168.0.1
  type: net_address
  required: false
  The IP address of a DNS nameserver to be used by the resolver.
  visibility: readwrite
  minimum number of values: 1
  maximum number of values: 3
  value: 10.0.0.1
  value: 192.168.0.1
```

EJEMPLO 4-5 Agregación de un valor

Utilice el subcomando `addpropvalue` para agregar el valor especificado a la propiedad especificada de servicio o instancia de servicio seleccionados. El nuevo valor se anexa al final de la lista de valores de propiedad existente para la propiedad.

```
$ svcprop -p keymap/layout keymap:default
US-English
$ svccfg -s keymap:default addpropvalue keymap:layout UK-English
$ svccfg -s keymap:default listprop keymap:layout
keymap/layout astring      "US-English" "UK-English"
```

En el ejemplo `setprop` anterior, todos los valores del conjunto de valores se deben especificar a la vez. Si sólo se especifica un valor, dicho valor se convierte en el nuevo conjunto de un valor. En este ejemplo `addpropvalue`, los valores agregados son distintos. Para acceder a estos valores, debe utilizar la función `libscf_scf_iter_property_values()` para iterar sobre los valores. Si bien `listprop` muestra ambos valores, describe muestra sólo el primer valor e informa que el número máximo permitido de valores para esta propiedad es uno.

```
$ svccfg -s keymap:default describe -v keymap/layout
keymap/layout astring      US-English
    type: astring
    required: true
    The keyboard layout
    visibility: readwrite
    minimum number of values: 1
    maximum number of values: 1
    value: US-English
```

▼ Cómo modificar un valor de propiedad `ttymon`

Este procedimiento muestra cómo modificar parámetros transferidos a `ttymon`.

1. Identifique el servicio que desea modificar.

La página del comando `man ttymon(1M)` establece que el servicio a modificar es `svc:/system/console-login`. La página del comando `man ttymon(1M)` también contiene descripciones de las propiedades en el grupo de propiedades `ttymon`.

El siguiente comando muestra varias instancias del servicio `console-login` en esta imagen y muestra que la instancia `default` es la única instancia actualmente en línea:

```
$ svcs console-login
STATE      STIME      FMRI
disabled   10:49:43   svc:/system/console-login:term
disabled   10:49:43   svc:/system/console-login:termb
online     10:50:54   svc:/system/console-login:default
```

2. Identifique la propiedad que desea modificar.

El siguiente comando muestra el nombre, el tipo de datos, el valor y una breve descripción de cada propiedad en el grupo de propiedades `ttymon` en la instancia `default`:

```
$ svccfg -s console-login:default describe ttymon
ttymon                application
ttymon/device         astring      /dev/console
    The terminal device to be used for the console login prompt.
ttymon/terminal_type  astring
    Sets the initial value of the TERM environment variable
```

La salida anterior muestra que no hay ningún valor para la propiedad `terminal_type`. El siguiente comando confirma que el valor de la propiedad `ttymon/terminal_type` de la instancia `console-login:default` es actualmente nulo:

```
$ svcprop -p ttymon/terminal_type console-login:default
""
```

3. Modifique el valor de la propiedad.

Introduzca el siguiente comando para cambiar el valor de la propiedad `ttymon/terminal_type` de la instancia `console-login:default` a `xterm`:

```
$ svccfg -s system/console-login:default setprop ttymon/terminal_type=xterm
```

4. Confirme el nuevo valor en la instantánea en ejecución.

La siguiente salida muestra que el valor de la propiedad `terminal_type` se modificó en la configuración de edición pero no en la instantánea en ejecución:

```
$ svccfg -s console-login:default listprop ttymon/terminal_type
ttymon/terminal_type astring      xterm
$ svcprop -p ttymon/terminal_type console-login:default
""
```

Tras actualizar la instancia de servicio, el valor de la propiedad se cambia en la instantánea en ejecución:

```
$ svcadm refresh console-login:default
$ svcprop -p ttymon/terminal_type console-login:default
xterm
```

▼ Cómo modificar una variable de entorno para un entorno de proceso de servicio

Este procedimiento muestra cómo definir un valor para una variable de entorno en el entorno donde se ejecutarán los procesos iniciados por el servicio.

El ejemplo en este procedimiento muestra cómo modificar variables de entorno cron para ayudar con la depuración.

1. Verifique que el servicio se esté ejecutando.

La siguiente salida muestra que el servicio cron está en línea y un proceso cron está en ejecución.

```
$ svcs -p cron
STATE      STIME      FMRI
online     10:24:05   svc:/system/cron:default
           10:24:05      1089 cron
```

2. Configure variables de entorno.

El subcomando setenv define una variable de entorno para el entorno donde se ejecutará un proceso iniciado por un servicio o instancia de servicio.

Utilice el siguiente comando para comprobar los valores actuales de las variables de entorno que desea definir:

```
$ pargs -e `pgrep -f /usr/sbin/cron`
```

Las variables de entorno que se establecen en este ejemplo no tienen valores actuales.

Los siguientes comandos establecen las variables de entorno UMEM_DEBUG y LD_PRELOAD para el proceso /usr/sbin/cron iniciado por la instancia de servicio svc:/system/cron:default:

```
$ svccfg -s system/cron:default setenv UMEM_DEBUG default
$ svccfg -s system/cron:default setenv LD_PRELOAD libumem.so
```

3. Actualice y reinicie el servicio.

Al cambiar un valor de variable de entorno se requiere de un reinicio para que se lleve a cabo un refrescamiento.

```
$ svcadm refresh system/cron:default
$ svcadm restart system/cron:default
```

4. Verifique que el cambio se haya realizado.

La siguiente salida muestra que el servicio se ha reiniciado, el proceso tiene un nuevo ID de proceso, y las dos variables de entorno están establecidas para ese entorno de proceso.

```
$ svcs -p cron
STATE      STIME      FMRI
online     9:24:39   svc:/system/cron:default
           9:24:39      5601 cron

$ svcprop -g method -p environment system/cron:default
start/environment astring LD_PRELOAD=libumem.so UMEM_DEBUG=default
$ pargs -e `pgrep -f /usr/sbin/cron`
5601: /usr/sbin/cron
envp[0]: LOGNAME=root
```



```
envp[1]: LD_PRELOAD=libumem.so
envp[2]: PATH=/usr/sbin:/usr/bin
envp[3]: SMF_FMRI=svc:/system/cron:default
envp[4]: SMF_METHOD=start
envp[5]: SMF_RESTARTER=svc:/system/svc/restarter:default
envp[6]: SMF_ZONENAME=global
envp[7]: UMEM_DEBUG=default
```

Véase también El subcomando `unsetenv` anula una variable de entorno para un proceso iniciado por un servicio o instancia de servicio.

Agregación de grupos de propiedades, propiedades y valores de propiedad

Los siguientes comandos agregan propiedades y grupos de propiedades:

```
svccfg setprop
svccfg addpropvalue
```

Agrega la propiedad cuyo valor se establece si la propiedad aún no existe.

```
svccfg addpg
```

Agrega un nuevo grupo de propiedades a un servicio o instancia de servicio.

No olvide utilizar el comando `svccfg refresh` o `svcadm refresh` para confirmar cambios de configuración en la instantánea en ejecución.

EJEMPLO 4-6 Uso de `addpg` para crear un nuevo grupo de propiedades

Use el subcomando `addpg` para agregar un grupo de propiedades al servicio o instancia de servicio seleccionados.

```
svccfg -s FMRI addpg name type [flags]
```

type Por convención, el valor de *type* suele ser *application*. Consulte [Capítulo 5, Uso de SMF para controlar la aplicación](#) para obtener más información sobre tipos de grupos de propiedades.

flags Especifique *P* para el valor de *flags* para almacenar el grupo de propiedades y cualquier propiedad agregada como no persistente. Si se especifica *P*, este grupo de propiedades y las propiedades contenidas se eliminarán automáticamente en el reinicio. El valor *P* es un alias para `SCF_PG_FLAG_NONPERSISTENT`. Consulte la página del comando `man scf_service_add_pg(3SCF)`.

```
$ svccfg -s svc:/site/my-svc addpg config application
$ svccfg -s my-svc listprop config
config application
$ svccfg -s my-svc:default listprop config
$
```

En este ejemplo, el administrador agregó el grupo de propiedades config al servicio principal, my-svc, pero no a la instancia, my-svc:default. El comando listprop muestra que el grupo de propiedades config no existe en la instancia de servicio.

EJEMPLO 4-7 Uso de setprop para crear una nueva propiedad

Utilice el subcomando setprop para establecer un valor de propiedad como se describe en [“Configuración de valores de propiedad” \[68\]](#). Si la propiedad aún no existe en la instancia o servicio seleccionados, el grupo de propiedades se crea si el tipo y los indicadores se encuentran en las definiciones de plantilla. Si la propiedad aún no existe en la instancia o servicio seleccionados, debe especificar la propiedad *type*.

```
$ svccfg -s my-svc:default setprop config/vendor = astring: vendora
$ svccfg -s my-svc:default listprop config/vendor
config/vendor astring vendora
```

EJEMPLO 4-8 Uso de addpropvalue para crear una nueva propiedad

Utilice el subcomando addpropvalue para agregar un valor de propiedad como se describe en [“Configuración de valores de propiedad” \[68\]](#). Si la propiedad aún no existe en la instancia o servicio seleccionados, el grupo de propiedades se crea si el tipo y los indicadores se encuentran en las definiciones de plantilla. Si la propiedad aún no existe en la instancia o servicio seleccionados, debe especificar la propiedad *type*.

```
$ svccfg -s my-svc:default addpropvalue config/vendor astring: vendorb
$ -s my-svc:default addpropvalue config/customer astring: acustomer
$ svccfg -s my-svc:default listprop config
config application
config/vendor astring "vendora" "vendorb"
config/customer astring acustomer
```

Supresión de grupos de propiedades, propiedades y valores de propiedad

Los siguientes comandos suprimen valores de propiedad, propiedades y grupos de propiedades:

```
svccfg setprop
```

Suprimir todos los valores de una propiedad.

`svccfg delpropvalue`

Suprimir todos los valores de la propiedad especificada que coinciden con el patrón especificado.

`svccfg delprop`

Suprimir una propiedad.

`svccfg delpg`

Suprimir un grupo de propiedades.

`svccfg delcust`

Suprimir personalizaciones administrativas.

No olvide utilizar el comando `svccfg refresh` o `svcadm refresh` para confirmar cambios de configuración en la instantánea en ejecución.

Supresión de configuración administrativa

Las modificaciones de configuración realizadas mediante los comandos `svccfg` o las llamadas `libscf` modifican sólo la capa `admin` del repositorio de configuración de servicio. Consulte [“Capas de repositorio” \[26\]](#) para obtener información sobre capas. Al suprimir una configuración que sólo está definida en la capa `admin` y no existe en cualquier otra capa, esa configuración desaparece. Los comandos que muestran la configuración ya no muestran la configuración suprimida, incluso cuando utiliza la opción `-l` para mostrar todas las capas del repositorio de configuración de servicio. Consulte [“Supresión de configuración no administrativa” \[77\]](#) para obtener información sobre la supresión de una configuración que existe en otras capas.

EJEMPLO 4-9 Supresión de todos los valores de una propiedad

Utilice el subcomando `setprop` como se describe en [“Configuración de valores de propiedad” \[68\]](#). Para suprimir todos los valores de una propiedad, no especifique cualquier tipo o valor. Los valores se suprimen, pero la propiedad todavía existe.

```
$ svccfg -s my-svc:default setprop config/vendor =
$ svccfg -s my-svc:default listprop config/vendor
config/vendor    astring
```

EJEMPLO 4-10 Supresión de todos los valores coincidentes de una propiedad

Utilice el subcomando `delpropvalue` para suprimir todos los valores de la propiedad con nombre que coincide con el patrón especificado.

```
$ svccfg -s my-svc:default setprop config/tool = astring: '(hammer tongs wrench)'
```

```
$ svccfg -s my-svc:default listprop config
config          application
config/customer astring  acustomer
config/vendor   astring  "vendra" "vendorb"
config/tool     astring  "hammer tongs wrench"
$ svccfg -s my-svc:default delpropvalue config/vendor '*b'
$ svccfg -s my-svc:default delpropvalue config/tool '*tong*'
$ svccfg -s my-svc:default listprop config
config          application
config/customer astring  acustomer
config/vendor   astring  vendora
config/tool     astring  "hammer tongs wrench"
$ # config/tool is a single value that is a value set
$ svccfg -s my-svc:default delpropvalue config/tool '*tong*'
$ svccfg -s my-svc:default listprop config
config          application
config/customer astring  acustomer
config/vendor   astring  vendora
config/tool     astring
```

EJEMPLO 4-11 Supresión de una propiedad

Utilice el subcomando `delprop` para suprimir la propiedad con nombre del servicio o instancia de servicio seleccionados.

```
$ svccfg -s my-svc:default delprop config/tool
$ svccfg -s my-svc:default listprop config
config          application
config/customer astring  acustomer
config/vendor   astring  vendora
```

EJEMPLO 4-12 Supresión de un grupo de propiedades

Los subcomandos `delpg` y `delprop` pueden suprimir un grupo de propiedades. El subcomando `delpgs` suprime el grupo de propiedades con nombre del servicio o instancia de servicio seleccionados. El subcomando `delprop` suprime el grupo de propiedades con nombre si no hay ninguna propiedad con nombre.

```
$ svccfg -s my-svc:default delpg config
$ svccfg -s my-svc:default listprop config
$
```

EJEMPLO 4-13 Supresión de personalizaciones

El subcomando `delcust` suprime personalizaciones administrativas en el servicio o instancia de servicio seleccionados. Antes de utilizar el subcomando `delcust`, utilice el subcomando `listcust` con el mismo patrón u opción para ver lo que se va a suprimir. Si se especifica un patrón, el patrón debe coincidir con una propiedad o grupo de propiedades.

```
$ svccfg -s my-svc:default listcust
config                               application admin
config/customer                     astring      admin      acustomer
config/vendor                       astring      admin      "vendora" "vendorb"
config/tool                         astring      admin      "hammer tongs wrench"
$ svccfg -s my-svc:default listcust '*tool'
config/tool                         astring      admin      "hammer tongs wrench"
$ svccfg -s my-svc:default delcust '*tool'
Deleting customizations for property: config/tool
$ svccfg -s my-svc:default listcust '*tool'
$ svccfg -s my-svc:default listcust
config                               application admin
config/customer                     astring      admin      acustomer
config/vendor                       astring      admin      "vendora" "vendorb"
```

Supresión de configuración no administrativa

La configuración que existe en las capas `site-profile`, `system-profile` y `manifest` del repositorio de configuración de servicio está definida en archivos de perfil y manifiestos de servicio. Consulte [“Capas de repositorio” \[26\]](#) para obtener información sobre capas. SMF mantiene el repositorio de configuración de servicio en sincronización con el contenido del sistema de archivos. Cualquier configuración definida en un archivo de perfil o manifiesto en una ubicación estándar aún existe en el sistema de archivos después de la personalización administrativa, incluso una vez suprimida, y se mantiene almacenada en el repositorio de configuración de servicio. Se dice que la configuración definida en un manifiesto o perfil admite paquetes. Al suprimir la configuración que admite paquetes, la información no se suprime del sistema de archivos pero se *enmascara* de modo que no se vea en la vista normal. Consulte la página del comando `man smf(5)` para obtener una descripción de entidades enmascaradas.

Suprimir la configuración que admite paquetes es una personalización administrativa. En este caso, el subcomando `delcust` *desenmascara* la configuración, en lugar de suprimir algo. Utilice el subcomando `listcust -M` para ver la configuración enmascarada. Utilice el subcomando `delcust -M` para desenmascarar una configuración o deshacer la supresión o enmascaramiento de la configuración.

EJEMPLO 4-14 Supresión de configuración que admite paquetes

En [“Supresión de configuración administrativa” \[75\]](#), el grupo de propiedades `config` del servicio `my-svc` sólo existía en la capa `admin`. El grupo de propiedades `config` no existía en cualquier manifiesto o perfil. Cuando estas propiedades se suprimieron, se eliminaron del sistema. En este ejemplo se muestra el resultado diferente cuando suprime la configuración que admite paquetes.

La propiedad está definida en el manifiesto de servicio:

```
$ svccfg -s pkg/server listprop -l all pkg/inst_root
pkg/inst_root astring    admin          /export/ipsrepos/Solaris11
pkg/inst_root astring    manifest       /var/pkgrepo
$ svccfg -s pkg/server delprop pkg/inst_root
```

Después de la supresión, no se muestra la propiedad mediante `listprop` sin ninguna opción. Debido a que la propiedad admite paquetes, la propiedad aún existe en el repositorio de configuración de servicio y puede visualizarse utilizando las opciones `-l` o `-M` con el subcomando `listprop`.

```
$ svccfg -s pkg/server listprop pkg/inst_root
$ svccfg -s pkg/server listprop -l all pkg/inst_root
pkg/inst_root astring    admin          MASKED /export/ipsrepos/Solaris11
pkg/inst_root astring    manifest       MASKED /var/pkgrepo
$ svccfg -s pkg/server listcust -M
pkg/inst_root astring    admin          MASKED /export/ipsrepos/Solaris11
```

EJEMPLO 4-15 Desenmascaramiento de configuración

Cuando desenmascara la propiedad, ambas personalizaciones desaparecen:

- La propiedad ya no está más enmascarada u oculta.
- La propiedad ya no tiene sus valores personalizados.

```
$ svccfg -s pkg/server delcust -M
Deleting customizations for property: pkg/inst_root
$ svccfg -s pkg/server listprop -l all pkg/inst_root
pkg/inst_root astring    manifest       /var/pkgrepo
$ svccfg -s pkg/server listprop pkg/inst_root
pkg/inst_root astring    /var/pkgrepo
```

Agregación de instancias de servicio

Las instancias de un servicio permiten que varias configuraciones de un servicio se ejecuten de manera simultánea. Las instancias de servicio heredan y personalizan la configuración de servicio común.

Utilice el subcomando `add` para crear una nueva entidad con el nombre especificado como elemento secundario del servicio seleccionado.

```
$ svcs -Ho inst pkg/server
default
$ svccfg -s pkg/server add s11
$ svcs -Ho inst pkg/server
default
s11
```

Reversión de instantáneas

Cada una de las siguientes operaciones crea una nueva instantánea en ejecución:

- `svcadm restart manifest-import`
- `svcadm refresh`
- `svccfg refresh`

El subcomando `revert` revierte las personalizaciones administrativas (capa `admin`) de la instancia especificada por la opción `-s` y su servicio a los valores registrados en la instantánea con nombre o la instancia actualmente seleccionada. Utilice el subcomando `listsnap` para ver una lista de posibles instantáneas para esta instancia de servicio. Utilice el subcomando `selectsnap` para seleccionar una instantánea en modo interactivo.

```
$ svcprop -p pkg/inst_root pkg/server:default
pkg/inst_root astring /export/ipsrepos/Solaris11
$ svccfg -s pkg/server:default listsnap
initial
previous
running
start
$ svcprop -s previous -p pkg/inst_root pkg/server:default
pkg/inst_root astring /var/pkgrepo
```

Debido a que el subcomando `revert` revierte todas las personalizaciones administrativas, enumere todas las personalizaciones administrativas y examine sus valores antes de revertirlas.

```
$ svcprop -s previous -l admin pkg/server:default
pkg/inst_root astring /var/pkgrepo
$ svccfg -s pkg/server:default revert previous
$ svcadm refresh pkg/server:default
$ svcprop -p pkg/inst_root pkg/server:default
pkg/inst_root astring /var/pkgrepo
```

Importación y la aplicación de manifiestos y perfiles

Al reiniciar el servicio `manifest-import`, los manifiestos en ubicaciones estándar se importan y los perfiles en ubicaciones estándar se aplican si son nuevos o fueron modificados. Consulte [“Paquetes de servicio” \[25\]](#) para ubicaciones estándar de perfiles y manifiestos. Si la importación de un manifiesto o la aplicación de un perfil resulta en el inicio o detención del servicio, se ejecuta el método apropiado, si es que existe.

Si especifica un archivo en una ubicación estándar para el comando `svccfg import` se reinicia el servicio `manifest-import`.

La práctica recomendada es colocar los archivos de perfil y manifiesto en ubicaciones estándar y reiniciar el servicio `manifest-import` en lugar de utilizar los comandos `svccfg import` o `svccfg apply`.

```
$ svcadm restart manifest-import
```

Al reiniciar el servicio `manifest-import`, la configuración de los perfiles y manifiestos en ubicaciones estándar se aplica a la capa `manifest`, `system-profile` o `site-profile` de las instancias afectadas. Las instancias afectadas se refrescan y validan, y se crea una nueva instantánea.

Al importar manifiestos o aplicar perfiles y manifiestos en ubicaciones no estándar, la configuración se aplica a la capa `admin` de las instancias afectadas. No se recomienda utilizar ubicaciones no estándar para la entrega de configuración inicial o predeterminada. Para realizar un gran número de cambios de configuración, es posible que la importación o aplicación desde una ubicación no estándar sea más fácil que la emisión de muchos comandos, pero pierde el beneficio de los mecanismos de gestión automatizados del servicio `admin`. Para gestionar la entrega de servicio, el servicio `manifest-import` requiere ubicaciones conocidas y esperadas.

Configuración de varios sistemas

Para implementar la misma configuración en varios sistemas, cree un perfil SMF que especifique los servicios que desea activar y los valores de las propiedades, y coloque dicho perfil en el directorio `site` en cada sistema. Como se indica en [“Capas de repositorio” \[26\]](#), se prefieren las personalizaciones locales por sobre los valores proporcionados cuando se instaló el sistema, y las personalizaciones en la `site-profile` están en segundo lugar sólo respecto de las personalizaciones realizadas por un administrador o aplicación.

Los perfiles agregan y establecen propiedades para los servicios e instancias existentes, y especifican nuevas instancias de servicio. Los perfiles pueden especificar casi todo lo que puede especificar un manifiesto.

Utilice uno de los siguientes métodos para crear un perfil de sitio:

- Utilice el comando `svcbundle` con `bundle-type=profile` para crear un nuevo archivo de perfil.
- Utilice el comando `svccfg extract` para capturar información de perfil de un servicio existente.

Personalice valores de propiedad en el archivo de perfil e incluya comentarios sobre el motivo de cada personalización. Copie el archivo a `/etc/svc/profile/site` y reinicie el servicio `manifest-import`.

▼ Cómo crear un perfil mediante `svcbundle`

El servicio `svc:/system/rmtmpfiles` es responsable de depurar el directorio `/tmp` durante el inicio. De manera predeterminada, el servicio `rmtmpfiles` no depura `/var/tmp`. Para limpiar `/var/tmp` durante el proceso de inicio, cambie el comportamiento del servicio `svc:/system/rmtmpfiles` estableciendo la propiedad `options/clean_vartmp` en `true`. La forma más sencilla de lograr este comportamiento en varios sistemas es crear un perfil y colocarlo en `/etc/svc/profile/site` en cada sistema.

1. Cree el perfil.

El siguiente comando crea un nuevo perfil en `/tmp/rmtmpfiles.xml`.

```
$ svcbundle -o /tmp/rmtmpfiles.xml -s service-name=system/rmtmpfiles \
-s bundle-type=profile -s service-property=options:clean_vartmp:boolean:true
```

2. Realice los cambios necesarios en el perfil.

3. Copie el perfil en el directorio correcto.

```
$ cp /tmp/rmtmpfiles.xml /etc/svc/profile/site/rmtmpfiles.xml
```

4. Reinicie el servicio de importación de manifiesto para aplicar el perfil al sistema.

```
$ svcadm restart manifest-import
```

ejemplo 4-16 Instalación automática de un perfil mediante `svcbundle`

Si no necesita realizar cambios en el nuevo perfil, puede utilizar la opción `-i` para instalar el perfil en cuanto se crea. El comando `svcbundle` escribirá el perfil en `/etc/svc/profile/site/rmtmpfiles.xml` y reiniciará el servicio `manifest-import`. Cualquier archivo existente con el mismo nombre en el directorio `/etc/svc/profile/site` se sobrescribirá.

```
# svcbundle -i -s service-name=system/rmtmpfiles \
-s bundle-type=profile -s service-property=options:clean_vartmp:boolean:true
```

▼ Cómo crear un perfil mediante `svccfg`

1. Cree un perfil.

El comando `svccfg extract` muestra un perfil de servicio para el servicio o instancia especificados para las capas `admin` y `site-profile`. Para extraer valores de otras capas, utilice la opción `-l`.

```
$ svccfg extract -l current network/dns/client > dnsclientprofile.xml
```

2. Realice los cambios necesarios en el perfil.

Cambie el nombre del perfil a un nombre significativo. De manera predeterminada, el nombre está establecido en `extract`, como se muestra en el siguiente ejemplo:

```
<service_bundle type='profile' name='extract'>
```

3. Copie el perfil en el directorio correcto.

```
$ cp dnsclientprofile.xml /etc/svc/profile/site/dnsclientprofile.xml
```

4. Reinicie el servicio de importación de manifiesto para aplicar el perfil al sistema.

```
$ svcadm restart manifest-import
```

Modificación de los servicios controlados por inetd

Un servicio controlado mediante `inetd` es un servicio SMF que fue convertido a partir de una configuración en el archivo `inetd.conf`. El comando `inetd` es el reiniciador delegado para estos servicios.

El siguiente procedimiento muestra cómo cambiar los valores de propiedades de servicios controlados por `inetd`.

Para confirmar que el servicio que desea modificar es controlado por `inetd`, invoque el comando `inetadm` sin opciones ni argumentos para enumerar todos los servicios controlados por `inetd`. El siguiente ejemplo sólo muestra una lista parcial.

```
$ inetadm
ENABLED  STATE      FMRI
enabled  online     svc:/application/cups/in-lpd:default
...
disabled disabled  svc:/application/x11/xfs:default
```

La opción `-l` del comando `inetadm` enumera todas las propiedades del servicio controlado por `inetd`. En el siguiente ejemplo, el mensaje de error indica que el servicio especificado no es un servicio controlado por `inetd`. “No restarter property” (Sin propiedad de reiniciador) significa que el reiniciador principal, `svc.startd`, gestiona la instancia de servicio.

```
$ inetadm -l ssh
Error: Specified service instance "svc:/network/ssh:default" has no
restarter property. inetd is not the delegated restarter of this instance.
```

Del mismo modo, en el ejemplo siguiente, el mensaje “Couldn't find property 'general/restarter'” (No se ha podido encontrar la propiedad 'general/reiniciador') indica que el reiniciador predeterminado, `svc.startd`, gestiona la instancia de servicio.

```
$ svcprop -p general/restarter ssh
svcprop: Couldn't find property 'general/restarter' for instance
'svc:/network/ssh:default'.
```

Si un servicio es controlado por `inetd`, su reiniciador es `inetd`, como se muestra en el siguiente ejemplo.

```
$ svcprop -p general/restarter xfs
svc:/network/inetd:default
```

▼ Cómo cambiar un valor de propiedad para un servicio controlado por `inetd`

1. Enumere las propiedades para el servicio.

Utilice la opción `-l` del comando `inetadm` para enumerar todas las propiedades del servicio especificado. Inspeccione los valores actuales de las propiedades.

```
# inetadm -l FMRI
```

2. Cambie un valor de propiedad.

Utilice la opción `-m` del comando `inetadm` para cambiar el valor de una propiedad especificada. En la página del comando `man` asociada con el servicio, debe encontrar información específica sobre las propiedades de un servicio.

```
# inetadm -m FMRI property-name=value
```

Para suprimir un valor de propiedad, especifique un valor vacío.

```
$ inetadm -m svc property=""
```

3. Verifique que el valor de propiedad se haya modificado.

Enumere las propiedades de nuevo para asegurarse de que los cambios apropiados se hayan producido.

```
# inetadm -l FMRI
```

4. Confirme que el cambio se haya aplicado.

Confirme que el cambio de propiedad se haya hecho efectivo.

ejemplo 4-17 Modificación del comando para ejecutarse cuando se inicia un servicio controlado por `inetd`

En este ejemplo se muestra cómo agregar o eliminar una opción o argumento de la línea de comandos de un servicio controlado por `inetd`. El comando que se ejecuta cuando se inicia el servicio es el valor de la propiedad `exec`.

Utilice la opción `-l` del comando `inetadm` para enumerar todas las propiedades del servicio especificado de modo que pueda inspeccionar el valor actual de la propiedad `exec`. En este ejemplo se muestra el servicio `svc:/application/x11/xfs`, que el servidor de fuentes del sistema de ventanas X. Consulte la página del comando `man xfs(1)` para obtener más información.

```
$ inetadm -l xfs | grep exec
exec="/usr/bin/xfs -inetd"
```

Utilice la opción `-m` del comando `inetadm` para cambiar el valor de la propiedad `exec` del servicio especificado.

```
$ inetadm -m xfs exec="/usr/bin/xfs -inetd -config /opt/site/fs/config"
```

Verifique que el valor de propiedad se haya modificado.

```
$ inetadm -l xfs | grep exec
exec="/usr/bin/xfs -inetd -config /opt/site/fs/config"
```

Modificación de servicios que están configurados por un archivo

Unos pocos servicios SMF que no están gestionados por `inetd` obtienen su configuración a partir de un archivo en lugar de las propiedades del servicio. Para modificar esta configuración, edite el archivo de configuración y utilice los comandos SMF para reiniciar el servicio. Estos archivos de configuración se pueden cambiar mientras el servicio está en ejecución, pero el contenido de los archivos es de sólo lectura cuando el servicio está activo.

Antes de editar un archivo de configuración directamente, compruebe las siguientes condiciones:

- Asegúrese de que el archivo de configuración no contiene un mensaje advirtiéndole que no edite directamente.
- Asegúrese de que el servicio no tiene un grupo de propiedades de tipo `configfile`.

```
$ svcprop -g configfile network/ntp
```

Si el servicio tiene un grupo de propiedades de tipo `configfile`, modifique las propiedades en esos grupos de propiedades y no el archivo de configuración. Consulte [“Uso de un cliché para crear un archivo de configuración” \[98\]](#).

Por ejemplo, para agregar un nuevo servidor NTP que admita sus clientes NTP, agregue una nueva entrada para el servidor en el archivo `/etc/inet/ntp.conf` y, luego, reinicie el servicio NTP como se muestra en el siguiente comando:

```
$ svcadm restart svc:/network/ntp:default
```

Para activar IKEv2, modifique el archivo `/etc/inet/ike/ikev2.config` para configurar el daemon IKEv2 y, luego, active el daemon IKEv2 como se muestra en el siguiente comando. Para editar el archivo `ikev2.config`, utilice el comando `pfedit` como usuario que tiene asignado el perfil de gestión IPsec de red. Al editar el archivo de esta manera se conserva la correcta propiedad del archivo. Consulte la página del comando `man pfedit(1M)` para obtener más información sobre el uso de `pfedit`.

```
$ svcadm enable svc:/network/ipsec/ike:ikev2
```


♦ ♦ ♦ 5 C A P Í T U L O 5

Uso de SMF para controlar la aplicación

En este capítulo se describe cómo proporcionar servicios de soporte para la aplicación, incluido cómo utilizar un cliché para crear un archivo de configuración si la aplicación no puede utilizar interfaces de biblioteca `libscf` para leer las propiedades.

Para obtener ejemplos adicionales de la creación y la entrega de servicios para realizar tareas como la configuración de aplicaciones, consulte [Capítulo 7, “Automating System Change as Part of Package Installation”](#) de “Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2 ” y [Capítulo 8, “Advanced Topics For Package Updating”](#) de “Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2 ”.

Creación de un servicio SMF

Un manifiesto de servicio contiene el conjunto completo de propiedades asociadas a un servicio específico, incluidas instancias, dependencias, secuencias de comandos a ejecutar cuando el servicio se inicia y se detiene, y valores de propiedades de aplicaciones predeterminados. Los manifiestos también proporcionan información de plantilla como una descripción del servicio. Consulte la página del comando `man service_bundle(4)` y el paquete de servicio `/usr/share/lib/xml/dtd/service_bundle.dtd.1` DTD para obtener una descripción completa del contenido y el formato de un manifiesto SMF. Consulte también “[Servicios de nombres, instancias, propiedades y grupos de propiedades](#)” [96] para reglas de denominación y asignación de tipos de grupos de propiedades.

Utilizar la herramienta de generador de paquete de servicio `svcbundle` es una buena forma de crear un servicio simple o de iniciar un servicio más complejo. Para obtener más información, consulte la página del comando `man svcbundle(1M)`. Puede utilizar el paquete de servicio DTD y otros manifiestos de servicio para completar un servicio más complejo.

La ubicación estándar para manifiestos personalizados es `/lib/svc/manifest/site`. Los manifiestos almacenados en esta ubicación se importan al repositorio de configuración de servicio mediante el servicio `svc:/system/early-manifest-import:default` durante el proceso de inicio antes de que comience cualquier servicio. La ejecución temprana del proceso de importación garantiza que el repositorio contendrá información de los manifiestos más

actuales antes de que los servicios se inicien. Los manifiestos almacenados en esta ubicación estándar también se importan cuando se reinicia el servicio `svc:/system/manifest-import`.

Se pueden utilizar varios manifiestos para describir un único servicio. Esto puede ser útil, por ejemplo, para definir una nueva instancia de un servicio sin modificar el manifiesto existente para el servicio. Sin embargo, si la misma propiedad en la misma capa para el mismo servicio o instancia es definida por varios manifiestos, la SMF no puede determinar el valor que se debe utilizar. Cuando se detecta este tipo de conflicto, la instancia se coloca en el estado de mantenimiento. Consulte [“Capas de repositorio” \[26\]](#) para obtener más información sobre las capas.

Agregue un nombre y metadatos de descripción a los manifiestos de modo que los usuarios puedan obtener información sobre este servicio a partir de los comandos `svcs` y `svccfg describe`. También puede agregar descripciones de valores de propiedad. Consulte los elementos `value`, `values` y `template` en la DTD.

Utilice el comando `svccfg validate` para validar el archivo de manifiesto de servicio o FMRI de instancia de servicio. Con el manifiesto, el método y los archivos de perfil en ubicaciones estándar, reinicie el servicio `manifest-import` para instalar y configurar instancias de servicio. Utilice el comando `svcs` para comprobar el estado de las instancias de servicio.

Esta sección muestra cómo crear servicios SMF personalizados. El prefijo `site` está reservado para personalizaciones específicas del sitio. Un servicio denominado `svc:/site/service-name` no entrará en conflicto con los servicios proporcionados en una versión de Oracle Solaris.

▼ Cómo crear un servicio SMF mediante la herramienta de generador de paquete de servicio

Este procedimiento muestra cómo crear un servicio que utiliza una secuencia de comandos personalizada existente como método de inicio.

1. Determine el modelo de servicio.

De manera predeterminada, `svcbundle` crea un servicio `transient`. Determine si la secuencia de comandos de método de inicio para este servicio inicia cualquier daemon de larga ejecución y es un servicio `contract`. Consulte [“Modelos de servicio” \[20\]](#) y la página del comando `man svc.startd(1M)` para obtener información sobre los modelos de servicio.

2. Copie la secuencia de comandos en la ubicación estándar.

En este ejemplo, la secuencia de comandos que será la secuencia de comandos de método de inicio para este servicio se denomina `ex_svc`. Copie esta secuencia de comandos en `/lib/svc/method/ex_svc`.

3. Cree un manifiesto inicial.

En este ejemplo, el nombre del servicio es `site/ex_svc`. Este es un servicio transitorio y no es necesario un método de detención.

```
$ svcbundle -o /tmp/ex_svc.xml -s service-name=site/ex_svc \
-s start-method=/lib/svc/method/ex_svc
```

4. Realice todos los cambios necesarios en el manifiesto.

Compruebe que el contenido del manifiesto `/tmp/ex_svc.xml` sea lo que necesita. Agregue comentarios según sea necesario.

5. Verifique que la ruta de manifiesto sea válida.

Utilice el comando `svccfg validate` para asegurarse de que el manifiesto de servicio sea válido.

6. Copie el manifiesto en el directorio estándar.

```
$ cp /tmp/ex_svc.xml /lib/svc/manifest/site/ex_svc.xml
```

7. Importe el manifiesto e inicie el servicio.

```
$ svcadm restart manifest-import
```

ejemplo 5-1 Instalación automática de un manifiesto generado

Si no necesita realizar cambios en el nuevo manifiesto de servicio, puede utilizar la opción `-i` para instalar el manifiesto en cuanto se crea. El comando `svcbundle` escribirá el manifiesto en `/lib/svc/manifest/site` y reiniciará el servicio `manifest-import`. Cualquier archivo existente con el mismo nombre en el directorio `/lib/svc/manifest/site` se sobrescribirá.

```
# svcbundle -i -s service-name=site/ex_svc \
-s start-method=/lib/svc/method/ex_svc
```

Creación de un servicio para iniciar o detener una instancia de Oracle Database

En este ejemplo se presentan los siguientes servicios que ayudan a gestionar Oracle Database:

- Un servicio de base de datos que inicia o detiene una instancia de Oracle Database.
- Un servicio de listener que inicia el listener, que es un proceso que gestiona el tráfico entrante de las solicitudes de conexión cliente a la instancia de base de datos.

Este ejemplo usa almacenamiento de copia de seguridad en archivo. Una alternativa al uso de almacenamiento con copia de seguridad en archivo es utilizar la función Automatic Storage

Management (ASM). ASM es un gestor de volumen y un sistema de archivo para archivos Oracle Database.

Las siguientes variables de entorno se deben definir para cada instalación de Oracle Database:

ORACLE_HOME	La ubicación donde está instalada la base de datos. En el ejemplo de esta sección, la ubicación de la instalación de la base de datos es <code>/opt/oracle/product/home</code> .
ORACLE_SID	El ID de sistemas para identificar de manera única una base de datos concreta en un sistema.

En este ejemplo, estas variables de entorno se establecen en los manifiestos de servicio y, a continuación, se utilizan en las secuencias de comandos de método.

Servicio de inicio y detención de base de datos

Esta sección muestra el manifiesto de servicio de control de instancia de Oracle Database, `/lib/svc/manifest/site/oracle.xml`. Las siguientes son algunas funciones a tener en cuenta acerca de este manifiesto de servicio:

- Se define una instancia de servicio, denominada `svc:/site/application/database/oracle:default`. Esta instancia se encuentra activada de manera predeterminada.
En este ejemplo se muestran dos maneras de definir la instancia predeterminada. En este ejemplo, la instancia predeterminada está definida en el elemento `create_default_instance` en la parte superior del manifiesto. El elemento `instance` en la parte inferior del manifiesto muestra la otra manera de hacerlo.
- Este servicio requiere que se monten todos los sistemas de archivos locales y se inicialicen todas las interfaces de red.
Si está utilizando una base de datos con copia de seguridad en archivo, el servicio de base de datos debe depender del sistema de archivos local. Si utiliza ASM, el servicio de base de datos debe depender del servicio que gestiona ASM. El servicio de base de datos debe depender de la red para permitir conexiones de clientes remotos.
- El elemento `method_environment` en el elemento `method_context` define las variables de entorno `ORACLE_HOME` y `ORACLE_SID`, que identifican la instancia de base de datos a iniciar o detener. Estos valores están disponibles para la secuencia de comandos de método a utilizar.
Si crea varias instancias de este servicio (consulte el elemento `instance` en la parte inferior del manifiesto), entonces, es posible que cada instancia necesite su propio elemento `method_context` para definir los valores `ORACLE_HOME` y `ORACLE_SID` únicos para esa base de datos en particular.
- Los atributos del elemento `method_context` pueden definir una agrupación de recursos además del proyecto y directorio de trabajo mostrados en este ejemplo. También

puede definir un elemento `method_profile` o `method_credential` en el elemento `method_context`. El elemento `method_credential` puede especificar los valores `supp_groups` y `limit_privileges` además de los valores `user`, `group` y `privileges` mostrados en este ejemplo. Consulte la DTD para obtener más información.

- La secuencia de comandos de método de inicio/detención es `/lib/svc/method/oracle`. Se incrementa el número de segundos predeterminado antes de que se agote el tiempo de espera.
- Un usuario debe tener asignada la autorización `solaris.smf.manage.oracle` para activar o desactivar esta instancia de servicio. En este ejemplo, el usuario `oracle` tiene asignada la autorización `solaris.smf.manage.oracle`.

```
<?xml version="1.0"?>

<!--
  Define a service to control the startup and shutdown of a database instance.
-->

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">

<service_bundle type="manifest" name="oracle">
<service name="site/application/database/oracle" type="service" version="1">
  <create_default_instance enabled="true" />

  <!--
    Wait for all local file systems to be mounted.
    Wait for all network interfaces to be initialized.
  -->

  <dependency type="service"
    name="fs-local"
    grouping="require_all"
    restart_on="none">
    <service_fmri value="svc:/system/filesystem/local" />
  </dependency>

  <dependency type="service"
    name="network"
    grouping="require_all"
    restart_on="none">
    <service_fmri value="svc:/milestone/network:default" />
  </dependency>

  <!-- Define the methods. -->

  <method_context project=":default" working_directory=":default">
    <method_credential user="oracle" group="dba" privileges=":default" />
    <method_environment>
      <envvar name="ORACLE_HOME" value="/opt/oracle/product/home" />
      <envvar name="ORACLE_SID" value="oracle" />
    </method_environment>
  </method_context>
```

```

<exec_method type="method"
  name="start"
  exec="/lib/svc/method/oracle start"
  timeout_seconds="120"/>

<exec_method type="method"
  name="stop"
  exec="/lib/svc/method/oracle stop"
  timeout_seconds="120" />

<!--
  What authorization is needed to allow the framework
  general/enabled property to be changed when performing the
  action (enable, disable, etc) on the service.
-->

<property_group name="general" type="framework">
  <propval type="astring"
    name="action_authorization"
    value="solaris.smf.manage.oracle" />
</property_group>

<!-- Define an instance of the database. -->

<!--<instance name="default" enabled="true" />-->

<stability value="Evolving" />
</service>
</service_bundle>

```

Agregue un nombre y metadatos de descripción al manifiesto de modo que los usuarios puedan obtener información sobre este servicio a partir de los comandos `svcs` y `svccfg describe`. Consulte el elemento `template` en la DTD.

Utilice el comando `svccfg validate` para asegurarse de que el manifiesto de servicio sea válido.

La siguiente secuencia de comandos de método de inicio/detención es `/lib/svc/method/oracle`, para el servicio de control de instancia de Oracle Database. Este método llama a los comandos `dbstart` y `dbshut` de la base de datos.

```

#!/bin/ksh -p

. /lib/svc/share/smf_include.sh

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
export PATH=$PATH:$ORACLE_HOME/bin

function startup
{
    dbstart $ORACLE_HOME
}

```

```
function shutdown
{
    dbshut $ORACLE_HOME
}

case $1 in
    start) startup ;;
    stop) shutdown ;;

    *) echo "Usage: $0 { start | stop }" >&2
       exit $SMF_EXIT_ERR_FATAL
       ;;
esac

exit $SMF_EXIT_OK
```

Servicio de listener de base de datos

El listener es un proceso que gestiona el tráfico entrante de las solicitudes de conexión cliente a la instancia de base de datos. El servicio de listener depende de la instancia de servicio de base de datos cuyas conexiones cliente está gestionando.

Esta sección muestra el manifiesto de servicio de listener de instancia de Oracle Database, `/lib/svc/manifest/site/listener.xml`. Las siguientes son algunas funciones a tener en cuenta acerca de este manifiesto de servicio:

- Se define una instancia de servicio, denominada `svc:/site/application/database/listener:default`. Esta instancia se encuentra activada de manera predeterminada.
- Este servicio requiere que se inicie el servicio de control de instancia de Oracle Database, `svc:/site/application/database/oracle`. Si la instancia de base de datos se reinicia por cualquier motivo, el listener también se reiniciará.
- El elemento `method_environment` en el elemento `method_context` define las variables de entorno `ORACLE_HOME` y `ORACLE_SID`, que identifican la instancia de base de datos a iniciar o detener. Estos valores están disponibles para la secuencia de comandos de método a utilizar.

Si crea varias instancias de este servicio (consulte el elemento `instance` en la parte inferior del manifiesto), entonces, es posible que cada instancia necesite su propio elemento `method_context` para definir los valores `ORACLE_HOME` y `ORACLE_SID` únicos para esa base de datos en particular.

- Los atributos del elemento `method_context` pueden definir una agrupación de recursos además del proyecto y directorio de trabajo mostrados en este ejemplo. También puede definir un elemento `method_profile` o `method_credential` en el elemento `method_context`. El elemento `method_credential` puede especificar los valores `supp_groups` y `limit_privileges` además de los valores `user`, `group` y `privileges` mostrados en este ejemplo. Consulte la DTD para obtener más información.

- La secuencia de comandos de método de inicio/detención es `/lib/svc/method/listener`. Se incrementa el número de segundos predeterminado antes de que se agote el tiempo de espera.
- Un usuario debe tener asignada la autorización `solaris.smf.manage.oracle` para activar o desactivar esta instancia de servicio.
- El servicio es `transient`. Consulte [“Modelos de servicio” \[20\]](#).

```
<?xml version="1.0"?>

<!--
  Define a service to control the startup and shutdown of a database listener.
-->

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">

<service_bundle type="manifest" name="listener">
<service name="site/application/database/listener" type="service" version="1">
  <create_default_instance enabled="true" />
  <!--<single_instance />-->

  <!-- Wait for the database to be started. -->

  <dependency type="service"
    name="oracle"
    grouping="require_all"
    restart_on="refresh">
    <service_fmri value="svc:/site/application/database/oracle" />
  </dependency>

  <!-- Define the methods. -->

  <method_context project=":default" working_directory=":default">
    <method_credential user="oracle" group="dba" privileges=":default" />
    <method_environment>
      <envvar name="ORACLE_HOME" value="/opt/oracle/product/home" />
      <envvar name="ORACLE_SID" value="oracle" />
    </method_environment>
  </method_context>

  <exec_method type="method"
    name="start"
    exec="/lib/svc/method/listener start"
    timeout_seconds="150"/>

  <exec_method type="method"
    name="stop"
    exec="/lib/svc/method/listener stop"
    timeout_seconds="30" />

  <!--
    What authorization is needed to allow the framework
    general/enabled property to be changed when performing the
    action (enable, disable, etc) on the service.
```

```
-->

<property_group name="general" type="framework">
  <propval type="astring"
    name="action_authorization"
    value="solaris.smf.manage.oracle" />
</property_group>

<!-- Make the instance transient (since it backgrounds itself). -->

<property_group name="startd" type="framework">
  <propval name="duration" type="astring" value="transient" />
</property_group>

<!-- Define an instance of the listener. -->

<!--<instance name="default" enabled="true" />-->

  <stability value="Evolving" />
</service>
</service_bundle>
```

Agregue un nombre y metadatos de descripción al manifiesto de modo que los usuarios puedan obtener información sobre este servicio a partir de los comandos `svcs` y `svccfg describe`. Consulte el elemento `template` en la DTD.

La siguiente secuencia de comandos de método de inicio/detención es `/lib/svc/method/listener`, para el servicio de listener de instancia de Oracle Database. Este método inicia o detiene un proceso del listener, `lsnrctl`. Cuando `lsnrctl` se inicia, comprueba el estado del servicio de base de datos.

```
#!/bin/ksh -p

. /lib/svc/share/smf_include.sh

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
export PATH=$PATH:$ORACLE_HOME/bin

function startup
{
    lsnrctl start

    # Wait for the listener to report ready.

    i=0
    while ! lsnrctl status | grep -i ready ; do
        ((i = i+1))
        if (( $i == 120 )) ; then
            # It's been *at least* 2 minutes, time to give up.
            echo "The listener failed to report ready." >&2
            exit $SMF_EXIT_ERR_FATAL
        fi
    fi
}
```

```

        sleep 1
    done

    # Ping the database once to prove it is now available.

    if ! tnsping $ORACLE_SID ; then
        exit $SMF_EXIT_ERR_FATAL
    fi
}

function shutdown
{
    lsnrctl stop
}

case $1 in
    start) startup ;;
    stop) shutdown ;;

    *) echo "Usage: $0 { start | stop }" >&2
       exit $SMF_EXIT_ERR_FATAL
       ;;
esac

exit $SMF_EXIT_OK

```

Servicios de nombres, instancias, propiedades y grupos de propiedades

Los nombres de instancia y servicio deben coincidir con la siguiente expresión:

```
[A-Za-z0-9][_A-Za-z0-9.-]*,[A-Za-z0-9][_A-Za-z0-9.-]*
```

El nombre de instancia o servicio distingue mayúsculas y minúsculas, debe comenzar con un carácter alfanumérico y puede contener caracteres alfanuméricos, guiones de subrayado (_), guiones (-) y puntos (.). Para mantener la compatibilidad con versiones anteriores, se permite una única coma (,) entre el primer y último carácter.

Los nombres de propiedad y grupos de propiedades deben coincidir con la siguiente expresión:

```
[A-Za-z0-9-._~:/?#\[\]@!$&'\"()*+;= %]+
```

Un nombre de propiedad o grupo de propiedades distingue mayúsculas y minúsculas y puede contener caracteres alfanuméricos, guion (-), punto (.), guion de subrayado (_), tilde (~), dos puntos (:), barra inclinada (/), signo de interrogación (?), signo de almohadilla (#), corchetes ([y]), arroba (@), signo de exclamación (!), símbolo de dólar (\$), símbolo de "y" comercial (&), comillas simples ('), paréntesis ((y)), asterisco (*), signo más (+), coma (,), punto y coma (;), signo igual (=), espacio y signo de porcentaje (%).

En un FMRI, los nombres de propiedad o grupo de propiedades están codificados según [Uniform Resource Identifier \(URI\) Generic Syntax RFC 3986](#) con excepción de que el carácter de coma no está codificado.

Un tipo de grupo de propiedades es una categoría para este grupo de propiedades. Los tipos de grupos de propiedades incluyen `application`, `dependency`, `method`, `framework`, `implementation` y `template`. Se pueden agregar tipos de grupo de propiedades adicionales, siempre que cumplan con la convención de denominación extendida en `smf(5)`. No especifique `framework`, `implementation` o `template` como el tipo de su grupo de propiedades. Los grupos de propiedades de tipo `framework`, `implementation` o `template` tienen un uso especial en SMF. Se espera que los grupos de propiedades de tipo `application` sean sólo de interés para el servicio al que el grupo está adjunto.

▼ Cómo convertir una secuencia de comandos de control en ejecución en un servicio SMF

Este procedimiento muestra cómo reemplazar una secuencia de comandos de control de ejecución con un manifiesto de servicio SMF de modo que el servicio de control de ejecución pueda ser gestionado por SMF. Para convertir una secuencia de comandos de control de ejecución, utilice el nombre `rc-script` con la opción `-s` del comando `svcbundle`. Consulte la página del comando `man svcbundle(1M)` para obtener más información o escriba `svcbundle help rc-script`.

1. Determine el modelo de servicio.

De manera predeterminada, `svcbundle` crea un servicio `transient`. Determine si esta secuencia de comandos de control de ejecución inicia cualquier `daemon` de larga ejecución y es un servicio `contract`. Consulte “[Modelos de servicio](#)” [20] y la página del comando `man svc.startd(1M)` para obtener información sobre los modelos de servicio.

2. Cree un manifiesto inicial.

En este ejemplo, el nombre del servicio es `ex_con` y es un servicio de contrato que se ejecuta en el nivel 2.

```
$ svcbundle -o /tmp/ex_con.xml -s service-name=ex_con
-s rc-script=/etc/init.d/ex_con:2 -s model=contract
```

3. Realice todos los cambios necesarios en el manifiesto.

Compruebe que el contenido del manifiesto `/tmp/ex_con.xml` sea lo que necesita. Agregue comentarios según sea necesario.

4. Copie el manifiesto en el directorio estándar.

```
$ cp /tmp/ex_con.xml /lib/svc/manifest/site/ex_con.xml
```

5. Detenga el servicio existente.

```
$ /etc/init.d/ex_con stop
```

6. Desactive la secuencia de comandos de control de ejecución.

Elimine cualquier enlace a la secuencia de comandos de control de ejecución de los directorios `rcn.d` pertinentes.

7. Importe el manifiesto e inicie el servicio.

```
$ svcadm restart manifest-import
```

8. Mencione el nuevo servicio.

Verifique que el nuevo servicio exista y esté en el estado esperado.

```
$ svcs ex-con
```

Uso de un cliché para crear un archivo de configuración

Si la aplicación no se puede utilizar para interfaces de biblioteca `libscf` para leer propiedades, puede utilizar un cliché para crear un archivo de configuración. Un *servicio de cliché* crea archivos de configuración mediante un archivo de cliché y valores de propiedad definidos en el servicio de cliché. Un *archivo de cliché* contiene una definición estructural de un archivo de configuración que es requerido por un servicio aunque ese servicio esté ahora gestionado por SMF. Los servicios de cliché le permiten aprovechar las ventajas de la gestión de configuración SMF sin cambios en la aplicación existente.

El cliché se utiliza para generar un archivo de configuración inmediatamente antes de que se ejecute un método `start` de instancia de servicio. Si actualiza los valores de propiedad de cliché, reinicie el servicio para incorporar los cambios en el archivo de configuración antes de que la aplicación se inicie y lea el archivo de configuración.

▼ Cómo crear un servicio de cliché

Un archivo de cliché contiene una definición estructural de un archivo de configuración que sigue siendo requerido por un servicio aunque ese servicio esté ahora gestionado por SMF. La utilidad `svcio` genera el archivo de configuración a partir de las definiciones en el archivo de cliché y las propiedades del servicio SMF. Consulte la página del comando `man svcio(1)` para obtener más información sobre la utilidad `svcio` y la página del comando `man smf_stencil(4)` para obtener información sobre el formato de archivo de cliché.

1. Cree un archivo de cliché.

El archivo de cliché le indica a la utilidad `svcio` el formato a utilizar para crear el archivo de configuración. La utilidad `svcio` convierte las propiedades SMF en archivos de configuración específicos de la aplicación basados en una plantilla denominada cliché.

2. Agregue un grupo de propiedades al servicio.

El grupo de propiedades de servicio de cliché le indica a la utilidad `svcio` la ruta y el propietario a utilizar para crear el archivo de configuración. SMF vuelve a generar la configuración para todos los servicios de cliché antes de iniciar o refrescar métodos. Los grupos de propiedades `configfile` le indican a SMF cómo generar los archivos de configuración. Cada grupo de propiedades `configfile` describe un único archivo de configuración para el servicio y le indica a `svcio` cómo generar estos archivos a partir de otras propiedades almacenadas en el repositorio SMF.

Para configurar un servicio para que tome en cuenta el cliché, agregue un grupo de propiedades para cada archivo de configuración gestionado que contenga las rutas del archivo de cliché a utilizar como plantilla y del archivo de configuración resultante. El grupo de propiedades tiene las siguientes propiedades:

<code>path</code>	La ruta en la que escribir el archivo de configuración, por ejemplo <code>/etc/svc.conf</code> .
<code>stencil</code>	La ruta del archivo de cliché a utilizar, en relación con <code>/lib/svc/stencils</code> . Por ejemplo, si el valor de la propiedad <code>stencil</code> es <code>svc.stencil</code> , se utilizará el archivo <code>/lib/svc/stencils/svc.stencil</code> .
<code>mode</code>	El modo que se va a utilizar para el archivo de configuración (<code>path</code>), por ejemplo <code>644</code> .
<code>owner</code>	El propietario que se debe definir para el archivo de configuración (<code>path</code>). Si esta propiedad no está definida, el propietario del archivo es el usuario que invoca <code>svcio</code> .
<code>group</code>	El grupo que se debe definir para el archivo de configuración (<code>path</code>). Si esta propiedad no está definida, el grupo será el grupo predeterminado para <code>path</code> .

Ejemplos de servicio de cliché en Oracle Solaris

Servicios para Puppet y Kerberos para utilizar clichés para proporcionar archivos de configuración.

Servicio de cliché Puppet

Puppet es un kit de herramientas para gestionar la configuración de varios sistemas. En Oracle Solaris, la aplicación Puppet es gestionada por SMF.

Vista de alto nivel de servicios Puppet

Al instalar el paquete `system/management/puppet`, obtiene dos instantáneas de servicio SMF: `puppet:master` y `puppet:agent`. De manera predeterminada, estas instancias están desactivadas.

Después de activar estas instancias, el siguiente comando muestra los servicios de contrato `puppet:master` y `puppet:agent`.

```
$ svcs -p puppet
STATE      STIME      FMRI
online     17:19:32   svc:/application/puppet:agent
           17:19:32   2565 puppet
online     17:19:32   svc:/application/puppet:master
           17:19:32   2567 puppet
```

El siguiente comando muestra un poco más de información sobre los procesos iniciados por los servicios de contrato:

```
$ ps -o pid,args -p 2565,2567
PID COMMAND
2565 /usr/ruby/1.9/bin/ruby /usr/sbin/puppet agent --logdest /var/log/puppet/puppet-
2567 /usr/ruby/1.9/bin/ruby /usr/sbin/puppet master --logdest /var/log/puppet/puppet
```

Como sugirió la salida `ps`, puppet escribe los archivos log en `/var/log/puppet`:

```
$ ls /var/log/puppet
puppet-agent.log  puppet-master.log
```

Archivo de configuración Puppet inicial

Puppet espera utilizar un archivo de configuración denominado `/etc/puppet/puppet.conf`. La aplicación `/usr/sbin/puppet` lee información de configuración de `/etc/puppet/puppet.conf` y no de propiedades definidas en las instancias de servicio `application/puppet`. Para proporcionar el archivo de configuración requerido, cada instancia puppet proporciona un archivo de cliché y un grupo de propiedades `configfile`. El grupo de propiedades `configfile` le indica a la utilidad `svcio` que ejecute y cree el archivo de configuración especificado. El archivo de cliché se utiliza para escribir datos de valores de propiedad de servicio en el archivo de configuración en el formato correcto.

El siguiente comando muestra todas las propiedades de servicio puppet que están en un grupo de propiedades de tipo `configfile`. Esta salida muestra que ambas instancias del servicio

puppet tienen las mismas propiedades configfile con los mismos valores. Cada instancia de servicio puppet proporciona la ruta al archivo de configuración, el modo del archivo de configuración y la ruta al archivo de cliché.

```
$ svcprop -g configfile puppet
svc:/application/puppet:master/:properties/puppet_stencil/mode astring 0444
svc:/application/puppet:master/:properties/puppet_stencil/path astring /etc/puppet/puppet.conf
svc:/application/puppet:master/:properties/puppet_stencil/stencil astring puppet.stencil
svc:/application/puppet:agent/:properties/puppet_stencil/mode astring 0444
svc:/application/puppet:agent/:properties/puppet_stencil/path astring /etc/puppet/puppet.conf
svc:/application/puppet:agent/:properties/puppet_stencil/stencil astring puppet.stencil
```

Los siguientes comandos confirman que estas propiedades de instancia se heredan del servicio principal.

```
$ svccfg -s puppet listprop -l all puppet_stencil
puppet_stencil      configfile manifest
puppet_stencil/mode  astring  manifest      0444
puppet_stencil/path  astring  manifest      /etc/puppet/puppet.conf
puppet_stencil/stencil astring  manifest      puppet.stencil
$ svccfg -s puppet:agent listprop -l all puppet_stencil
$ svccfg -s puppet:master listprop -l all puppet_stencil
```

Para la infraestructura, es posible que necesite instancias puppet:agent1 y puppet:agent2, por ejemplo. En ese caso, debe personalizar los valores de propiedad y agregar propiedades para cada instancia como se muestra en [“Modificación del archivo de configuración Puppet” \[102\]](#).

A continuación, se muestra el contenido inicial del archivo de configuración, /etc/puppet/puppet.conf:

```
# WARNING: THIS FILE GENERATED FROM SMF DATA.
# DO NOT EDIT THIS FILE. EDITS WILL BE LOST.
#
# See puppet.conf(5) and http://docs.puppetlabs.com/guides/configuring.html
# for details.
```

Archivo de cliché Puppet

El contenido del archivo de cliché le indica qué propiedades y qué otra información se escriben en el archivo de configuración. La ruta puppet.stencil que es el valor de la propiedad puppet_stencil/stencil se relaciona con /lib/svc/stencils. A continuación se muestra el contenido del archivo de cliché, /lib/svc/stencils/puppet.stencil:

```
# WARNING: THIS FILE GENERATED FROM SMF DATA.
# DO NOT EDIT THIS FILE. EDITS WILL BE LOST.
#
# See puppet.conf(5) and http://docs.puppetlabs.com/guides/configuring.html
# for details.
```

```
; walk each instance and extract all properties from the config PG
$%/(svc:/$%s:(.*)/:properties)/ {
$%{$%1/general/enabled:?
[$%2]
$%/$%1/config/(.*)/ {
$%3 = $%{$%1/config/$%3} }
}
}
```

En el archivo de cliché, `svc:/$%s:(.*)/:properties (o %1)` se expande a `svc:/application/puppet:agent/:properties` y `svc:/application/puppet:master/:properties`, donde `.*` (o %2) coincide con cada instancia. El nombre de la instancia es luego utilizada para etiquetar el bloque en el archivo de configuración. La siguiente incidencia de `.*` (o %3) coincide con cada propiedad en el grupo de propiedades `config` para la instancia de servicio %1. El cliché le indica a `svcio` escribir el nombre propiedad y el valor de esa propiedad de la instancia de servicio al archivo de configuración.

Modificación del archivo de configuración Puppet

Como puede ver en “[Archivo de configuración Puppet inicial](#)” [100], inicialmente sólo las líneas de comentario literal se escriben en el archivo de configuración. La escritura de valores de propiedad en el archivo de configuración es evitada por la evaluación del valor de la propiedad `general/enabled` en el archivo de cliché. El siguiente comando muestra que, de manera predeterminada, el valor de la propiedad `general/enabled` es falso:

```
$ svcprop -p general/enabled puppet
svc:/application/puppet:master/:properties/general/enabled boolean false
svc:/application/puppet:agent/:properties/general/enabled boolean false
```

El uso del comando `svcadm enable` para activar la instancia no cambia el valor de la propiedad `general/enabled`. Al cambiar el valor de la propiedad `general/enabled` a `true` y reiniciar la instancia, todas las propiedades en el grupo de propiedades `config` para esa instancia se escriben en el archivo de configuración.

```
$ svccfg -s puppet:agent setprop general/enabled=true
$ svcprop -p general/enabled puppet:agent
false
$ svcadm refresh puppet:agent
$ svcprop -p general/enabled puppet:agent
true
$ svcadm restart puppet:agent
```

El siguiente comando muestra que inicialmente la única propiedad en el grupo de propiedades `config` es la ruta al archivo log para cada instancia:

```
$ svcprop -p config puppet
svc:/application/puppet:master/:properties/config/logdest astring /var/log/puppet/puppet-
master.log
```

```
svc:/application/puppet:agent/:properties/config/logdest astring /var/log/puppet/puppet-agent.log
```

La propiedad `config` para la instancia activada ha sido agregada al archivo de configuración en un bloque etiquetado con el mismo nombre de instancia:

```
# WARNING: THIS FILE GENERATED FROM SMF DATA.
# DO NOT EDIT THIS FILE. EDITS WILL BE LOST.
#
# See puppet.conf(5) and http://docs.puppetlabs.com/guides/configuring.html
# for details.
```

```
[agent]
```

```
logdest = /var/log/puppet/puppet-agent.log
```

La documentación de configuración Puppet indica que el archivo de configuración puede tener los bloques `[main]`, `[agent]` y `[master]`. La configuración en el bloque `[main]` se aplica al agente y al elemento principal. Para el agente Puppet, la configuración en el bloque `[agent]` sobrescribe la misma configuración en el bloque `[main]`. Para el elemento principal Puppet, la configuración en el bloque `[master]` sobrescribe la misma configuración en el bloque `[main]`. Si desea proporcionar un bloque `[main]` para una configuración que sea común tanto al agente y al elemento principal, cree una instancia `puppet:main` y las propiedades `config` apropiadas para esa instancia.

Los comandos siguientes muestran cómo agregar la configuración al archivo de configuración Puppet.

```
$ svccfg -s puppet:agent
svc:/application/puppet:agent> setprop config/report=true
svc:/application/puppet:agent> setprop config/pluginsync=true
svc:/application/puppet:agent> refresh
svc:/application/puppet:agent> exit
$ svcadm restart puppet:agent
$ cat /etc/puppet/puppet.conf
# WARNING: THIS FILE GENERATED FROM SMF DATA.
# DO NOT EDIT THIS FILE. EDITS WILL BE LOST.
#
# See puppet.conf(5) and http://docs.puppetlabs.com/guides/configuring.html
# for details.
```

```
[agent]
```

```
logdest = /var/log/puppet/puppet-agent.log
pluginsync = true
report = true
```

Se pueden utilizar comandos similares para eliminar propiedades y cambiar los valores de propiedad. Consulte [Capítulo 4, Configuración de servicios](#). Para agregar una instancia `main`, utilice el comando `svccfg add` como se muestra en [“Agregación de instancias de servicio” \[78\]](#).

Servicio de cliché Kerberos

Otro ejemplo de un servicio Oracle Solaris que utiliza un cliché es Kerberos. El siguiente comando muestra que el grupo de propiedades configfile es krb5_conf, el archivo de cliché es /lib/svc/stencils/krb5.conf.stencil y el archivo de configuración es /etc/krb5/krb5.conf.

```
$ svcprop -g configfile svc:/system/kerberos/install:default
krb5_conf/disabled boolean true
krb5_conf/group astring sys
krb5_conf/mode integer 644
krb5_conf/owner astring root
krb5_conf/path astring /etc/krb5/krb5.conf
krb5_conf/stencil astring krb5.conf.stencil
```


Prácticas recomendadas y solución de problemas de SMF

En este apéndice se proporcionan algunas de las mejores prácticas recomendadas y se muestra cómo solucionar algunos problemas de servicio de SMF.

Prácticas recomendadas de SMF

La mayoría de los servicios describen la política de configuración. Si la configuración que desea no está implementada, modifique la descripción de la política modificando el servicio. Modifique los valores de las propiedades de servicio o cree nuevas instancias de servicio con distintos valores de propiedad. No desactive las instancias de servicio y edite los archivos de configuración que serán gestionados por un servicio SMF.

No modifique los manifiestos y perfiles de sistema entregados por Oracle o proveedores de software de terceros. Estos manifiestos y perfiles se pueden reemplazar al actualizar el sistema y, entonces, sus cambios en estos archivos se perderán. En su lugar, cree un perfil de sitio para personalizar el servicio o utilice el comando `svccfg` o `inetadm` para manipular las propiedades directamente. Los directorios `/lib/svc/manifest/site` y `/var/svc/manifest/site` también se reservan para uso específico del sitio. Oracle Solaris no entrega manifiestos a estos directorios.

Para aplicar la misma configuración personalizada en varios sistemas, utilice el comando `svcbundle` o el comando `svccfg extract` para crear un archivo de perfil. Personalice valores de propiedad en ese archivo e incluya comentarios sobre el motivo de cada personalización. Copie el archivo a `/etc/svc/profile/site` en cada sistema y reinicie el servicio `manifest-import` en cada sistema. Consulte [“Configuración de varios sistemas” \[80\]](#).

Al crear un perfil de sitio, asegúrese de que la configuración definida no entre en conflicto con la configuración definida en otro perfil de sitio para el mismo servicio o instancia de servicio. Cuando SMF encuentra la configuración en conflicto en la misma capa del repositorio de configuración de servicio, la instancia de servicio afectada se coloca en el estado de mantenimiento.

No utilice ubicaciones no estándar para archivos de perfil y manifiesto. Consulte [“Paquetes de servicio” \[25\]](#) para ubicaciones estándar de perfiles y manifiestos.

Al crear un servicio para su propio uso, utilice `site` al principio del nombre de servicio: `svc:/site/service_name:instance_name`.

No modifique la configuración del servicio de reiniciador maestro, `svc:/system/svc/restarter:default`, excepto para configurar niveles de registro como se describe en [“Especificación de la cantidad de mensajes de inicio” \[114\]](#).

Antes de utilizar el comando `svccfg delcust`, utilice el comando `svccfg listcust` con las mismas opciones. El subcomando `delcust` puede eliminar todas las personalizaciones administrativas en un servicio. Utilice el subcomando `listcust` para verificar que las personalizaciones se suprimirán mediante el subcomando `delcust`.

En secuencias de comandos, utilice FMRI de instancia de servicio completo: `svc:/service_name:instance_name`.

Resolución de problemas de servicio

En esta sección se analizan los aspectos siguientes:

- Confirmación de cambios de configuración en la instantánea en ejecución
- Corrección de servicios notificados con problemas
- Transición manual de una instancia al estado `degraded` o `maintenance`
- Corrección de un repositorio de configuración de servicios dañado
- Configuración de la cantidad de mensajes para mostrar o almacenar en el inicio del sistema
- Transición o inicio a un hito especificado
- Uso de SMF para investigar problemas de inicio
- Conversión de servicios `inetd` a servicios SMF

Comprensión de cambios de configuración

En el repositorio de configuración de servicio, SMF almacena los cambios de propiedad por separado de las propiedades en la instantánea en ejecución. Al cambiar la configuración del servicio, los cambios no aparecen inmediatamente en la instantánea en ejecución.

La operación de refrescamiento actualiza la instantánea en ejecución de la instancia de servicio especificada con los valores de la configuración de edición.

De manera predeterminada, el comando `svccfg` muestra propiedades en la instantánea en ejecución, y el comando `svccfg` muestra propiedades en la configuración de edición. Si ha cambiado los valores de propiedad pero no realizó un refrescamiento de la configuración, los

comandos `svcprow` y `svccfg` muestran diferentes valores de propiedad. Después de realizar un refrescamiento de la configuración, los comandos `svcprow` y `svccfg` muestran los mismos valores de propiedad.

Reiniciar no cambia la instantánea en ejecución. El comando `svcadm restart` no refresca la configuración. Use el comando `svcadm refresh svccfg refresh` para confirmar los cambios de configuración en la instantánea en ejecución.

Reparación de una instancia degradada, fuera de línea o en mantenimiento

Utilice el comando `svcs -x` sin argumentos para mostrar información explicativa sobre las instancias de servicio que cumplen cualquiera de las siguientes descripciones:

- El servicio está activado, pero no está en ejecución.
- El servicio impide que otro servicio activado se ejecute.

En la lista siguiente se resume cómo abordar problemas de servicio:

1. Diagnostique el problema, empezando por ver el archivo log de servicio.
2. Solucione el problema. Si solucionar el problema implica modificar la configuración de servicio, refresque el servicio.
3. Mueva los servicios afectados a un estado en ejecución.

▼ Cómo reparar una instancia que está en mantenimiento

Una instancia de servicio que está en mantenimiento está activada, pero no podrá ejecutarse.

1. Determine el motivo por el que la instancia está en mantenimiento.

La instancia puede estar transitando por el estado `maintenance` debido a que una acción administrativa aún no se ha completado. Si la instancia está en transición, su estado debería mostrarse como `maintenance*`.

En el ejemplo siguiente, las líneas “State”(Estado) y “Reason” (Motivo) muestran que el servicio `pkg/depot` está en el estado `maintenance` porque falló el método de inicio.

```
$ svcs -x
svc:/application/pkg/depot:default (IPS Depot)
  State: maintenance since September 11, 2013 01:30:42 PM PDT
Reason: Start method exited with $SMF_EXIT_ERR_FATAL.
  See: http://support.oracle.com/msg/SMF-8000-KS
  See: pkg.depota-config(1M)
  See: /var/svc/log/application-pkg-depot:default.log
```

Impact: This service is not running.

Inicie sesión en el sitio de soporte de Oracle para ver el artículo de conocimientos de reparación automática predictiva al que se hace referencia. En este caso, el artículo le indica ver el archivo log para determinar el motivo del fallo del método de inicio. La salida `svcs` proporciona el nombre del archivo log. Consulte [“Visualización de archivos log de servicio” \[40\]](#) para obtener más información sobre cómo ver el archivo log. En este ejemplo, el archivo log muestra la invocación de método de inicio y el mensaje de error fatal.

```
[ Sep 11 13:30:42 Executing start method ("/lib/svc/method/svc-pkg-depot start"). ]
pkg.depot-config: Unable to get publisher information:
The path '/export/ipsrepos/Solaris11' does not contain a valid package repository.
```

2. Solucione el problema.

Uno o más de los pasos siguientes pueden ser necesarios.

■ Actualice la configuración de servicio.

Si para solucionar el problema informado necesita modificar la configuración de servicio, use el comando `svccfg refresh` o `svcadm refresh` para cualquier servicio cuya configuración haya cambiado. Verifique que la configuración se actualice en la instantánea en ejecución mediante el comando `svcprop` para comprobar los valores de propiedad o mediante otras pruebas específicas para este servicio.

■ Asegúrese de que las dependencias estén en ejecución.

A veces la línea “Impact” (Impacto) en la salida `svcs -x` le indica que servicios que dependen del servicio que está en estado `maintenance` no están en ejecución. Utilice el comando `svcs -l` para comprobar el estado actual de servicios dependientes. Asegúrese de que todas las dependencias requeridas estén en ejecución. Utilice el comando `svcs -x` para verificar que todos los servicios activados estén en ejecución.

■ Asegúrese de que los procesos de contrato estén detenidos.

Si el servicio que está en estado `maintenance` es un servicio de contrato, determine si los procesos iniciados por el servicio no se han detenido. Cuando una instancia de servicio de contrato está en un estado de mantenimiento, el ID de contrato debe estar en blanco, como se muestra en el ejemplo siguiente, y todos los procesos asociados con esa instancia de servicio deben estar detenidos. Utilice `svcs -l` o `svcs -o ctid` para comprobar que no existe contrato para una instancia de servicio en mantenimiento. Utilice `svcs -p` para comprobar si algún proceso asociado con esta instancia de servicio aún está en ejecución. Cualquier proceso mostrado por `svcs -p` para una instancia de servicio en mantenimiento debe finalizarse.

```
$ svcs -l system-repository
fmri      svc:/application/pkg/system-repository:default
name      IPS System Repository
enabled   true
state     maintenance
```

```

next_state    none
state_time    September 17, 2013 07:18:19 AM PDT
logfile       /var/svc/log/application-pkg-system-repository:default.log
restarter     svc:/system/svc/restarter:default
contract_id   /lib/svc/manifest/application/pkg/pkg-system-repository.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)

```

3. Notifique al reiniciador que la instancia está reparada.

Cuando el problema indicado está solucionado, utilice el comando `svcadm clear` para devolver el servicio al estado `online`. Para servicios en el estado `maintenance`, el subcomando `clear` le indica al reiniciador para ese servicio que el servicio fue reparado.

```
$ svcadm clear pkg/depot:default
```

Si especifica la opción `-s`, el comando `svcadm` espera volver hasta que la instancia alcance el estado `online` o hasta que determine que la instancia no puede alcanzar el estado `online` sin la intervención de un administrador. Utilice la opción `-T` con la opción `-s` para especificar un límite superior en segundos para realizar la transición o determinar que no se puede realizar la transición.

4. Verifique que la instancia se haya reparado.

Utilice el comando `svcs` para verificar que el servicio que estaba en mantenimiento esté ahora en línea. Utilice el comando `svcs -x` para verificar que todos los servicios activados estén en ejecución.

▼ Cómo repara una instancia que está fuera de línea

Una instancia de servicio que está fuera de línea está activada, pero no está en ejecución o disponible para ejecutarse.

1. Determine por qué la instancia está fuera de línea.

La instancia puede estar transitando por el estado `offline` porque sus dependencias aún no se han cumplido. Si la instancia está en transición, su estado debería mostrarse como `offline*`.

2. Solucione el problema.

■ Active dependencias de servicio.

Si las dependencias requeridas están desactivadas, actívelas con el siguiente comando:

```
$ svcadm enable -r FMRI
```

■ Repare el archivo de dependencia.

Un archivo de dependencia puede estar faltando o no se puede leer. Es posible que desee utilizar `pkg fix` o `pkg revert` para solucionar este tipo de problema. Consulte la página del comando `man pkg(1)`.

3. Reinicie la instancia si es necesario.

Si la instancia está fuera de línea porque no se ha cumplido una dependencia necesaria, es posible que al solucionar o activar la dependencia se reinicie la instancia fuera de línea y vuelva a estar en línea sin necesidad de ninguna otra acción administrativa.

Si ha realizado alguna otra corrección para el servicio, entonces reinicie la instancia.

```
$ svcadm restart FMRI
```

4. Verifique que la instancia se haya reparado.

Utilice el comando `svcs` para verificar que la instancia que estaba fuera de línea esté ahora en línea. Utilice el comando `svcs -x` para verificar que todos los servicios activados estén en ejecución.

▼ Cómo repara una instancia degradada

Una instancia de servicio degradada está activada y en ejecución o disponible para su ejecución, pero está funcionando a una capacidad limitada.

1. Determine por qué la instancia está degradada.

2. Solucione el problema.

3. Solicite al reiniciador que coloque a la instancia en línea.

Cuando el problema indicado está solucionado, utilice el comando `svcadm clear` para devolver la instancia al estado `online`. Para las instancias en estado `degraded`, el subcomando `clear` solicita al reiniciador que pase esa instancia al estado `online`.

```
$ svcadm clear pkg/depot:default
```

4. Verifique que la instancia se haya reparado.

Utilice el comando `svcs` para verificar que la instancia que estaba degradada esté ahora en línea. Utilice el comando `svcs -x` para verificar que todos los servicios activados estén en ejecución.

Marcación de una instancia como degradada o en mantenimiento

Puede marcar una instancia de servicio que esté en estado degraded o maintenance. Es posible que desee hacer esto si la aplicación está parada en un bucle o en punto muerto, por ejemplo. La información sobre el cambio de estado se propaga a las dependencias de la instancia marcada, lo que puede ayudar a depurar otras instancias relacionadas.

Especifique la opción `-I` para solicitar un cambio de estado de inmediato.

Cuando se marca una instancia como maintenance, puede especificar la opción `-t` para solicitar un cambio de estado temporal. Las solicitudes temporales sólo se mantienen hasta el reinicio.

Si especifica la opción `-s` con el comando `svcadm mark`, `svcadm` marca la instancia y espera que la instancia ingrese al estado degraded o maintenance antes de volver. Utilice la opción `-T` con la opción `-s` para especificar un límite superior en segundos para realizar la transición o determinar que no se puede realizar la transición.

Diagnóstico y reparación de problemas del repositorio

En el inicio del sistema, el daemon de repositorio, `svc.configd`, realiza una comprobación de integridad del repositorio de configuración almacenado en `/etc/svc/repository.db`. Si la comprobación de integridad `svc.configd` falla, el daemon `svc.configd` escribe un mensaje en la consola, similar al siguiente:

```
svc.configd: smf(5) database integrity check of:
```

```
    /etc/svc/repository.db
```

```
failed. The database might be damaged or a media error might have
prevented it from being verified. Additional information useful to
your service provider is in:
```

```
    /system/volatile/db_errors
```

```
The system will not be able to boot until you have restored a working
database. svc.startd(1M) will provide a sulogin(1M) prompt for recovery
purposes. The command:
```

```
    /lib/svc/bin/restore_repository
```

```
can be run to restore a backup version of your repository. See
```

<http://support.oracle.com/msg/SMF-8000-MY> for more information.

El daemon `svc.configd` sale. Esa salida es detectada por el daemon `svc.startd`, y `svc.startd` inicia `sulogin`.

En el indicador `sulogin`, introduzca Ctrl-D para salir de `sulogin`. El daemon `svc.startd` reconoce la salida de `sulogin` y reinicia el daemon `svc.configd`, el cual comprueba el repositorio nuevamente. Es posible que el problema no vuelva a aparecer después de este reinicio adicional. No invoque directamente el daemon `svc.configd`. El daemon `svc.startd` inicia el daemon `svc.configd`.

Si `svc.configd` vuelve a informar una comprobación de integridad con fallos y está nuevamente en el indicador `sulogin`, asegúrese de que los sistemas de archivos no estén llenos. Utilizando la contraseña `root`, inicie sesión remotamente o en el indicador `sulogin`. Compruebe que haya espacio disponible en los sistemas de archivos `system/volatile` y `root`. Si alguno de estos sistemas de archivos está lleno, haga espacio y vuelva a iniciar el sistema. Si ninguno de estos sistemas de archivos está lleno, siga el procedimiento [Cómo restaurar un repositorio desde una copia de seguridad \[112\]](#).

El repositorio de configuración de servicio puede dañarse por cualquiera de los siguientes motivos:

- Fallo de disco
- Error de hardware
- Error de software
- Sobrescritura accidental del archivo

El siguiente procedimiento muestra cómo reemplazar un repositorio dañado con una copia de seguridad del repositorio.

▼ Cómo restaurar un repositorio desde una copia de seguridad

1. Inicie sesión.

Utilizando la contraseña `root`, inicie sesión remotamente o en el indicador `sulogin`.

2. Ejecute el comando para restaurar el repositorio:

```
# /lib/svc/bin/restore_repository
```

La ejecución de este comando lo guía por los pasos necesarios para restaurar una copia de seguridad que no está dañada. SMF realiza automáticamente copias de seguridad del repositorio como se describe en [“Copias de seguridad del repositorio” \[27\]](#).

SMF mantiene los datos de configuración persistentes y no persistentes. Consulte [“Repositorio de configuración de servicios” \[24\]](#) para obtener descripciones de estos dos repositorios.

El comando `restore_repository` sólo restaura el repositorio persistente. El comando `restore_repository` también reinicia el sistema, que destruye los datos de configuración no persistentes. Los datos no persistentes son datos de tiempo de ejecución que no son necesarios en el reinicio del sistema.

Al iniciar, el comando `/lib/svc/bin/restore_repository` muestra un mensaje similar al siguiente:

See <http://support.oracle.com/msg/SMF-8000-MY> for more information on the use of this script to restore backup copies of the smf(5) repository.

If there are any problems which need human intervention, this script will give instructions and then exit back to your shell.

Después de que el sistema de archivos `root (/)` se monta con permisos de escritura, o si el sistema es una zona local, se le pide que seleccione la copia de seguridad del repositorio para restaurar:

The following backups of `/etc/svc/repository.db` exists, from oldest to newest:

... *list of backups* ...

Las copias de seguridad se nombran según el tipo y la hora en que la copia de seguridad se ha realizado. Las copias de seguridad que empiezan con `boot` se completan antes de que se realiza el primer cambio en el repositorio después del inicio del sistema. Las copias de seguridad que empiezan con `manifest_import` se completan después de que `svc:/system/manifest-import:default` termina su proceso. La hora de la copia de seguridad se proporciona en formato `YYYYMMDD_HHMMSS`.

3. Introduzca la respuesta adecuada.

Normalmente, se selecciona la opción de copia de seguridad más reciente.

Please enter either a specific backup repository from the above list to restore it, or one of the following choices:

CHOICE	ACTION
-----	-----
<code>boot</code>	restore the most recent post-boot backup
<code>manifest_import</code>	restore the most recent manifest_import backup
<code>-seed-</code>	restore the initial starting repository (All customizations will be lost, including those made by the install/upgrade process.)
<code>-quit-</code>	cancel script and quit

Enter response [boot]:

Si presiona Intro sin especificar una copia de seguridad para restaurar, se selecciona la respuesta predeterminada, encerrada entre `[]`. Al seleccionar `-quit-`, se sale de la secuencia de comandos `restore_repository` y se regresa al indicador de shell.

Nota - Al seleccionar `-seed-`, se restaura el repositorio `seed`. Este repositorio está diseñado para usarse durante la instalación inicial y las actualizaciones. El uso del repositorio `seed` para fines de recuperación debe ser un último recurso.

Después de que la copia de seguridad para restaurar se ha seleccionado, se valida y se comprueba su integridad. Si hay problemas, el comando `restore_repository` imprime mensajes de error y le solicita otra selección. Una vez que se selecciona una copia de seguridad válida, se imprime la siguiente información y se le solicita confirmación final.

After confirmation, the following steps will be taken:

```
svc.startd(1M) and svc.configd(1M) will be quiesced, if running.
/etc/svc/repository.db
-- renamed --> /etc/svc/repository.db_old_YYYYMMDD_HHMMSS
/system/volatile/db_errors
-- copied --> /etc/svc/repository.db_old_YYYYMMDD_HHMMSS_errors
repository_to_restore
-- copied --> /etc/svc/repository.db
and the system will be rebooted with reboot(1M).
```

Proceed [yes/no]?

4. Escriba **yes** para solucionar el fallo.

El sistema se reinicia después de que el comando `restore_repository` ejecuta todas las acciones enumeradas.

Especificación de la cantidad de mensajes de inicio

De manera predeterminada, cada servicio que se inicia durante el inicio del sistema no muestra un mensaje en la consola. Utilice uno de los siguientes métodos para cambiar los mensajes que aparecen en la consola y los que se registran sólo en el archivo `log svc.startd`. El valor de *logging-level* puede ser uno de los valores se muestran en la siguiente tabla.

- Al iniciar un sistema SPARC, especifique la opción `-m` para el comando `boot` en el indicador `ok`. Consulte “Messages options” (Opciones de mensajes) en la página del comando `man kernel(1M)`.

`ok boot -m logging-level`
- Al iniciar un sistema x86, edite el menú GRUB para especificar la opción `-m`. Consulte “Agregación de argumentos del núcleo mediante la edición del menú de GRUB en el inicio” de “Inicio y cierre de sistemas Oracle Solaris 11.2 ” y “Messages options” (Opciones de mensajes) en la página del comando `man kernel(1M)`.

- Antes de reiniciar un sistema, utilice el comando `svccfg` para cambiar el valor de la propiedad `options/logging`. Si esta propiedad nunca se ha cambiado en este sistema, no saldrá y tendrá que agregarla. El siguiente ejemplo cambia a mensajes detallados. El cambio se hace efectivo en el siguiente reinicio del daemon `svc.startd`.

```
$ svccfg -s system/svc/restarter:default listprop options/logging
$ svccfg -s system/svc/restarter:default addpg options application
$ svccfg -s system/svc/restarter:default setprop options/logging=verbose
$ svccfg -s system/svc/restarter:default listprop options/logging
options/logging astring      verbose
```

TABLA A-1 Niveles de registro de mensajes de inicio SMF

Palabra clave de nivel de registro	Descripción
quiet	Muestra en la consola cualquier mensaje de error que requiera de intervención administrativa. También registra estos mensajes en <code>syslog</code> y en <code>/var/svc/log/svc.startd.log</code> .
verbose	Además de los mensajes proporcionados en el nivel quiet, muestra en la consola un único mensaje para cada servicio iniciado, y registra información <code>/var/svc/log/svc.startd.log</code> sobre errores que no requieren información administrativa.
debug	Además de los mensajes proporcionados en el nivel quiet, muestra en la consola un único mensaje para cada servicio iniciado, y registra cualquier mensaje de depuración <code>svc.startd</code> en <code>/var/svc/log/svc.startd.log</code> .

Especificación de hito SMF con el cual reiniciar

Cuando se inicia un sistema, puede especificar el hito SMF con el que se va a iniciar.

De manera predeterminada, todos los servicios para los que el valor de la propiedad `general/enabled` es `true` se inician en el inicio del sistema. Para cambiar el hito con el que se inicia un sistema, utilice uno de los métodos siguientes. El valor de *milestone* puede ser FMRI de un servicio de hitos o una palabra clave como se muestra en [Tabla A-2, “Hitos de inicio SMF y niveles de ejecución correspondientes”](#).

- Al iniciar un sistema SPARC, especifique la opción `-m` para el comando `boot` en el indicador `ok`. Consulte la opción `-m` en la página del comando `man kernel(1M)`.


```
ok boot -m milestone=milestone
```
- Al iniciar un sistema x86, edite el menú GRUB para especificar la opción `-m`. Consulte [“Agregación de argumentos del núcleo mediante la edición del menú de GRUB en el inicio” de “Inicio y cierre de sistemas Oracle Solaris 11.2”](#) y la opción `-m` en la página del comando `man kernel(1M)`.
- Antes de reiniciar un sistema, utilice el comando `svcadm milestone` con la opción `-d`. Tenga en cuenta que con o sin la opción `-d`, este comando restringe y restaura servicios

en ejecución inmediatamente. Con la opción `-d`, el comando también hace que el hito especificado sea el hito de inicio predeterminado. Este nuevo valor predeterminado persiste tras los reinicios.

\$ svcadm milestone -d milestone

Este comando no cambia el nivel de ejecución actual del sistema. Para cambiar el nivel de ejecución actual del sistema, utilice el comando `init`.

Si especifica la opción `-s`, `svcadm` cambia el hito y luego espera que la transición al hito especificado se complete antes de volver. El comando `svcadm` vuelve cuando todas las instancias han pasado al estado necesario para alcanzar el hito especificado o cuando determine que se requiere la intervención del administrador para realizar una transición. Utilice la opción `-T` con la opción `-s` para especificar un límite superior en segundos para completar la operación de cambio de hito o devolución.

La siguiente tabla describe hitos de inicio SMF, incluyendo cualquier nivel de ejecución Oracle Solaris correspondiente. Un *nivel de ejecución* del sistema define los servicios y recursos disponibles para los usuarios. Un sistema sólo puede estar en un nivel de ejecución a la vez. Para obtener información sobre niveles de ejecución, consulte [“Cómo funcionan los niveles de ejecución”](#) de [“Inicio y cierre de sistemas Oracle Solaris 11.2”](#), la página del comando `man inittab(4)` y el archivo `/etc/init.d/README`. Para obtener más información sobre hitos de inicio SMF, consulte el subcomando `milestone` en la página del comando `man svcadm(1M)`.

TABLA A-2 Hitos de inicio SMF y niveles de ejecución correspondientes

FMRI de hito SMF o palabra clave	Nivel de ejecución correspondiente	Descripción
none		La palabra clave <code>none</code> representa un hito donde ningún servicio se está ejecutando a excepción del reiniciador maestro. Cuando se especifica <code>none</code> , todos los servicios excepto <code>svc:/system/svc/restarter:default</code> se desactivan temporalmente. El hito <code>none</code> puede ser útil para depurar problemas de inicio. Consulte Cómo investigar problemas de inicio de servicios durante el inicio del sistema [117] para obtener instrucciones específicas.
all		La palabra clave <code>all</code> representa un hito que depende de cada servicio. Cuando se especifica <code>all</code> , se ignoran solicitudes de desactivación o activación temporal para todos los servicios. Este es el hito predeterminado utilizado por <code>svc.startd</code> .
svc:/milestone/single-user	s o S	Ignora solicitudes de desactivación o activación temporal para <code>svc:/milestone/single-user:default</code> y todos los servicios en los que depende directa o indirectamente. Desactiva temporalmente todos los demás servicios.
svc:/milestone/multi-user	2	Ignora solicitudes de desactivación o activación temporal para <code>svc:/milestone/multi-user:default</code> y todos los servicios en los que depende directa o indirectamente. Desactiva temporalmente todos los demás servicios.

FMRI de hito SMF o palabra clave	Nivel de ejecución correspondiente	Descripción
svc:/milestone/multi-user-server	3	Ignora solicitudes de desactivación o activación temporal para svc:/milestone/multi-user-server:default y todos los servicios en los que depende directa o indirectamente. Desactiva temporalmente todos los demás servicios.

Para determinar el hito con el que un sistema se inicia actualmente, use el comando `svcs`. El siguiente ejemplo muestra que el sistema se inicia para nivel de ejecución 3, `milestone/multi-user-server`:

```
$ svcs 'milestone/*'
STATE      STIME      FMRI
online     9:08:05   svc:/milestone/unconfig:default
online     9:08:06   svc:/milestone/config:default
online     9:08:07   svc:/milestone/devices:default
online     9:08:25   svc:/milestone/network:default
online     9:08:31   svc:/milestone/single-user:default
online     9:08:51   svc:/milestone/name-services:default
online     9:09:13   svc:/milestone/self-assembly-complete:default
online     9:09:23   svc:/milestone/multi-user:default
online     9:09:24   svc:/milestone/multi-user-server:default
```

Uso de SMF para investigar problemas de inicio de sistema

En esta sección se describen las acciones que se deben realizar si el sistema se bloquea durante el inicio o si un servicio clave no se puede iniciar durante el inicio.

▼ Cómo investigar problemas de inicio de servicios durante el inicio del sistema

Si se producen problemas al iniciar los servicios en el inicio del sistema, en ocasiones, el sistema se bloquea durante el inicio. Este procedimiento muestra cómo investigar los problemas de servicio que se producen en el momento del inicio.

1. Inicie sin iniciar los servicios.

El siguiente comando indica al daemon `svc.startd` que desactive temporalmente todos los servicios e inicie `sulogin` en la consola.

```
ok boot -m milestone=none
```

Consulte [“Especificación de hito SMF con el cual reiniciar” \[115\]](#) para obtener una lista de hitos SMF que puede usar con el comando `boot -m`.

2. Inicie sesión en el sistema como root.

3. Active todos los servicios.

```
# svcadm milestone all
```

4. Determine dónde se bloqueó el proceso.

Cuando el proceso de inicio se bloquea, determine qué servicios no se están ejecutando mediante la ejecución de `svcs -a`. Busque mensajes de error en los archivos log, en `/var/svc/log`.

5. Después de solucionar los problemas, verifique que todos los servicios se hayan iniciado.

a. Verifique que todos servicios necesarios estén en línea.

```
# svcs -x
```

b. Verifique que las dependencias de servicio `console-login` se hayan cumplido.

Este comando verifica que el proceso login en la consola se ejecutará.

```
# svcs -l system/console-login:default
```

6. Continúe con el proceso de inicio normal.

▼ **Cómo forzar un inicio de sesión de usuario único si el servicio del sistema de archivos local falla durante el inicio**

Los sistemas de archivos locales que no son necesarios para iniciar el sistema son montados por el servicio `svc:/system/filesystem/local:default`. Cuando algunos de esos sistemas de archivos no se pueden montar, el servicio `filesystem/local` entra en estado de mantenimiento. El inicio del sistema continúa, y cualquier servicio que no depende de `filesystem/local` se inicia. Los servicios que tienen una dependencia necesaria en el servicio `filesystem/local` no se iniciaron.

Este procedimiento explica cómo cambiar la configuración del sistema, de manera que un indicador `sulogin` aparezca inmediatamente después de que el servicio falla, en lugar de permitir que el inicio del sistema continúe.

1. Modifique el servicio `system/console-login`.

```
$ svccfg -s svc:/system/console-login
svc:/system/console-login> addpg site,filesystem-local dependency
svc:/system/console-login> setprop site,filesystem-local/entities = fmri: svc:/system/filesystem/local
```

```
svc:/system/console-login> setprop site,filesystem-local/grouping = astring: require_all
svc:/system/console-login> setprop site,filesystem-local/restart_on = astring: none
svc:/system/console-login> setprop site,filesystem-local/type = astring: service
svc:/system/console-login> end
```

2. Actualice el servicio.

```
$ svcadm refresh console-login
```

Cuando se produce un fallo con el servicio `system/filesystem/local:default`, el comando `svcs -vx` se debe utilizar para identificar el fallo. Después de que se ha reparado el fallo, utilice el siguiente comando para borrar el estado de error y permitir que el inicio del sistema continúe:

```
$ svcadm clear filesystem/local
```

Conversión de servicios `inetd` a los servicios SMF

El archivo `inetd.conf` en el sistema no debe contener entradas. El archivo `inetd.conf` sólo debe contener comentarios de que se trata de un archivo heredado que ya no se utiliza directamente. Si el archivo `inetd.conf` contiene cualquier entrada, siga las instrucciones de esta sección para convertir estas configuraciones a servicios SMF. Los servicios que están configurados en el archivo `inetd.conf` pero no están configurados como un servicio SMF no están disponibles para su uso. Los servicios que se configuran en el archivo `inetd.conf` no son reiniciados por el comando `inetd` directamente. En cambio, el comando `inetd` es el iniciador delegado para los servicios convertidos.

Durante el inicio del sistema inicial, las configuraciones en el archivo `inetd.conf` se convierten automáticamente a servicios SMF. Después del inicio del sistema inicial, se pueden agregar entradas al archivo `inetd.conf` mediante la instalación de software adicional no entregado por paquetes Image Packaging System (IPS). El software entregado por paquetes IPS incluye cualquier manifiesto SMF requerido, y ese manifiesto SMF inicia esa instancia de servicio con los valores de propiedad correctos.

Si el archivo `inetd.conf` en su sistema contiene entradas, use el comando `inetconv` para convertir esas configuraciones a servicios SMF. El comando `inetconv` convierte entradas `inetd.conf` en archivos de manifiesto de servicio SMF e importa esos manifiestos al repositorio SMF para iniciar las instancias de servicio. Consulte la página del comando [man inetconv\(1M\)](#) para obtener información sobre opciones de comando y para ver ejemplos de uso del comando.

El nombre del nuevo manifiesto SMF incorpora `service_name` a partir de la entrada `inetd.conf`. La entrada del archivo `inetd.conf` se guarda como propiedad de la nueva instancia de servicio. El nuevo manifiesto SMF especifica los grupos de propiedades y las propiedades para definir las acciones enumeradas en la entrada `inetd.conf`. Después de

ejecutar el comando `inetconv`, utilice los comandos `svcs` y `svccprop` para asegurarse de que la nueva instancia de servicio se haya creado y tenga los valores de propiedad correctos.

El comando `inetd` es el reiniciador delegado para servicios de Internet SMF. No utilice el comando `inetd` directamente para gestionar estos servicios. Utilice el comando `inetadm` sin opciones u operandos para ver una lista de los servicios controlados por `inetd`. Utilice los comandos `inetadm`, `svcadm` y `svccfg` para configurar y gestionar estos servicios convertidos.

El comando `inetconv` no modifica el archivo `inetd.conf` de entrada. Debe suprimir manualmente las entradas en el archivo `inetd.conf` después de ejecutar correctamente `inetconv`.

Para obtener más información sobre la configuración de servicios `inetd` que ya se convirtieron a servicios SMF, consulte [“Modificación de los servicios controlados por `inetd`” \[82\]](#).

Índice

A

- acciones, 18
- activar un servicio, 53
- actualizar configuración de servicio, 59
- admisión de paquetes, 77
- archivo `inetd.conf`, 28, 119
- archivo `resolv.conf`, 28
- archivos de cliché, 16, 98, 98
- archivos de configuración, 15, 16, 28, 98
- archivos log, 40
- ASM, 89
- Automatic Storage Management (ASM), 89
- autorizaciones, 30

B

- base de datos SMF *Ver* repositorio de configuración de servicio
- biblioteca de Utilidad de gestión de servicios `libscf`, 24
- biblioteca `libscf`, 24, 98
 - funciones de operación de lote, 16

C

- capa `admin`, 26, 48
- capa `manifest`, 26, 48
- capa `site-profile`, 26, 48
- capa `system-profile`, 26
- capas, 26
 - capa `admin`, 48
 - capa `manifest`, 48
 - capa `site-profile`, 48
- columna STATE, 31
- comando `boot`

- hito `none`, 117
- opción de hitos, 115
- comando `inetadm`, 119
 - ejemplo, 82
- comando `inetconv`, 119
- comando `inetd`, 119
- comando `restore_repository`, 112
- comando `sulogin`, 112
- comando `svcadm`, 53
 - opciones de sincronía, 16
 - subcomando `clear`, 107
 - subcomando `disable`, 56
 - subcomando `enable`, 53
 - subcomando `mark`, 111
 - subcomando `milestone`, 115, 117
 - subcomando `refresh`, 59
 - subcomando `restart`, 58, 107
 - subcomando `restart manifest-import`, 58, 60
- comando `svcbundle`
 - creación de manifiestos, 87
 - creación de perfiles, 81
 - instalación automáticamente, 81, 89
 - servicio `rc-script`, 97
- comando `svccfg`
 - archivo de entrada de comando, 68
 - subcomando `add`, 78
 - subcomando `addpg`, 73
 - subcomando `addpropvalue`, 68, 73
 - subcomando `apply`, 79
 - subcomando `delcust`, 74, 76
 - subcomando `delcust -M`, 77
 - subcomando `delete`, 60
 - subcomando `delpg`, 74
 - subcomando `delprop`, 74

- subcomando delpropvalue, 74
- subcomando describe, 42
- subcomando editprop, 66
- subcomando extract, 81
- subcomando import, 79
- subcomando listcust, 45, 48
- subcomando listcust -M, 60, 77
- subcomando listpg, 46
- subcomando listprop, 45
- subcomando listsnap, 48
- subcomando refresh, 59
- subcomando revert, 79
- subcomando selectsnap, 44, 79
- subcomando setenv, 68, 71
- subcomando setprop, 68, 73, 74
- subcomando unsetenv, 71
- subcomando validate, 88
- uso interactivo, 67
- visualización de propiedades, 42
- comando svcprop
 - visualización de propiedades, 42, 43
- comando svcs, 31
- configuración
 - agregación de grupos de propiedades y propiedades, 73
 - agregación de valores de propiedad, 68
 - configuración de valores de propiedad, 68
 - desenmascaramiento, 77
 - enmascaramiento, 77
 - modificación, 65
 - refrescamiento, 59
 - supresión de configuración que admite paquetes, 77
 - supresión de grupos de propiedades, 74
 - supresión de personalizaciones, 74, 76
 - supresión de propiedades, 74
 - supresión de valores de propiedad, 74
- configuración de notificación, 62
- configuración enmascarada, 77
- copias de seguridad, 27
- copias de seguridad boot, 27
- copias de seguridad manifest-import, 27

D

- daemon de reiniciador maestro, 22
- daemon de reiniciador maestro svc.startd, 22, 112
- daemon de repositorio svc.configd, 111
- datos de plantilla, 24
- dependencias, 21, 21
 - agrupaciones, 35
 - efecto en el estado de un servicio dependiente, 38
 - enumeración, 34
 - instancias de las que depende un servicio, 36
 - instancias que dependen de un servicio, 37
- desactivar un servicio, 56
- detener un servicio, 56
- DTD, 87

E

- eliminar un servicio, 60
- entidades enmascaradas, 48, 60
- entradas inittab, 28
- estado de servicio degraded, 21, 107, 111
- estado de servicio disabled, 21
- estado de servicio legacy_run, 31
- estado de servicio maintenance, 21, 107, 111
- estado de servicio offline, 21, 107
- estado de servicio online, 21
- estado de servicio uninitialized, 21
- estados de servicio
 - descripción, 21
 - enumeración, 31
 - modificación manualmente, 107, 111
 - transiciones, 21
- estados init Ver niveles de ejecución
- archivo /etc/inetd.conf, 28, 84, 119
- secuencias de comandos /etc/init.d, 28
- entradas /etc/inittab, 28
- secuencias de comandos /etc/rc?.d, 28

F

- FMRI, 20, 97

G

- grupo de propiedad config, 23
- grupo de propiedad general, 23
- grupo de propiedad restarter, 23
- grupo de propiedad start, 23
- grupos de propiedades, 23, 42
 - denominación, 96
 - no persistentes, 46
 - tipo, 45, 97
- grupos de propiedades de servicio, 23, 42
- grupos de propiedades no persistentes, 46

H

- hitos
 - actualmente iniciados, 117
 - niveles de ejecución correspondientes, 115
 - none, 117
 - reinicio, 115

I

- Identificador de recurso de gestión de errores *Ver* FMRI
- iniciar un servicio, 53
- inicio
 - hito actual, 117
 - niveles de registro, 114
 - para un hito SMF, 117
 - sin inicio de servicios, 117
- inicio de sesión de usuario único, 112
- instancias, 18
 - agregación, 78
 - asignación de nombres, 20
 - denominación, 96
- instancias de servicio, 18
 - agregación, 78
- instantánea en ejecución, 43
- instantánea initial, 27
- instantánea previous, 27
- instantánea running, 27
- instantánea start, 27
- instantáneas, 44, 48
 - actualización de configuración en ejecución, 59

- initial, 27
- instantánea en ejecución, 43
- previous, 27
- reversión, 79
- running, 27
- start, 27

K

- Kerberos
 - ejemplo de servicio de cliché, 104

M

- manifiestos, 18
 - directorio site, 87
 - formato, 87
 - ubicación estándar, 25, 87
- métodos, 18
- modelos de servicio, 20

N

- niveles de ejecución
 - actual, 117
 - hitos SMF correspondientes, 115
 - nivel predeterminado, 115
- notificación de evento, 62
- notificación de transición de estado, 62

O

- Oracle Database, 89
 - ASM, 89
 - servicio de base de datos, 90
 - servicio de listener, 93

P

- paquetes, 25
- paquetes de servicio
 - DTD, 87

- ubicaciones estándar, 25
- parámetros de notificación
 - visualización, 49
- perfiles, 18
 - creación, 80
 - directorio site, 25, 80
 - ubicación estándar, 25
- perfiles de derechos, 30
- permisos, 30
- personalizaciones
 - enumeración, 48
- privilegios, 30
- propiedades, 23, 24, 42
 - denominación, 96
 - en la instantánea en ejecución, 43
 - no persistentes, 46
 - vista compuesta de, 24, 43
- propiedades de servicio, 23, 24, 42
- propiedades no persistentes, 47
- Puppet
 - ejemplo de servicio de cliché, 99

R

- refrescar configuración de servicio, 59
- registro de errores, 40
- reiniciador de servicio, 22
- reiniciador predeterminado, 22
- reiniciadores, 18, 22
 - daemon de reiniciador maestro `svc.startd`, 22
 - reiniciador predeterminado, 22
 - reiniciadores delegados, 23
- reiniciadores delegados, 23
- reiniciar el servicio `manifest-import`, 58
- reiniciar un servicio, 58
- reinicio
 - hitos, 115
- repositorio *Ver* repositorio de configuración de servicio
- repositorio de configuración *Ver* repositorio de configuración de servicio
- repositorio de configuración de servicio, 18, 24
 - capas, 26
 - copias de seguridad, 27
 - daemon `svc.configd`, 111
 - datos de plantilla, 24
 - instantáneas, 27

- interfaces de biblioteca, 24
- modificación, 65
- reparación, 111
- roles, 30

S

- secuencias de comandos de control de ejecución, 28
 - conversión a servicio SMF, 97
- secuencias de comandos `init.d`, 28
- secuencias de comandos `rc?.d`, 28
- servicio de cliché, 98
 - ejemplo de Kerberos, 104
 - ejemplo Puppet, 99
- servicio de contrato, 20, 20
- servicio de daemon, 20
- servicio de espera, 20
- servicio `inetd`
 - convertir a servicio SMF, 119
 - modificación de argumentos de línea de comando, 82
 - modificación de propiedades, 82
- servicio secundario, 20
- servicio transitorio, 20
- servicios, 18
 - activación, 53
 - asignación de nombres, 20
 - denominación, 96
 - desactivación, 56
 - detención, 56
 - inicio, 53
 - modificación de configuración, 65
 - refrescamiento de configuración, 59
 - reinicio, 58
 - supresión, 60
- servicios de hitos, 18
- servicios `lrc`, 28
- subcomando `describe`
 - comando `svccfg`, 42
- subcomando `listcust`
 - comando `svccfg`, 45, 48
- subcomando `listpg`
 - comando `svccfg`, 46
- subcomando `listprop`
 - comando `svccfg`, 45

- subcomando listsnap
 - comando svccfg, 48
- subcomando selectsnap
 - comando svccfg, 44
- suprimir un servicio, 60

U

- utilidad svcio, 16, 98

V

- variables de entorno
 - modificación en los métodos, 71
- vista compuesta, 24, 43

