

# Guía del desarrollador de Oracle® VM Server for SPARC 3.4

**ORACLE®**

Referencia: E71833  
Agosto de 2016



# Contenido

---

<b>Cómo utilizar esta documentación .....</b>	<b>5</b>
<b>1 Desarrollo de plantillas, archivos XML y programas de Oracle VM Server for SPARC .....</b>	<b>7</b>
Desarrollo de plantillas, archivos XML y programas de Oracle VM Server for SPARC .....	7
<b>2 Uso de Plantillas de Oracle VM Server for SPARC .....</b>	<b>9</b>
Acerca de Plantillas de Oracle VM Server for SPARC .....	9
Instalación de las utilidades de Plantilla de Oracle VM Server for SPARC .....	10
Ciclo de vida de Plantilla de Oracle VM Server for SPARC .....	11
Ejemplos de Plantilla de Oracle VM Server for SPARC .....	14
<b>3 Uso de la interfaz XML con el Dominios lógicos Manager .....</b>	<b>21</b>
Transporte de XML .....	21
Servidor XMPP .....	22
Conexiones locales .....	22
Protocolo XML .....	22
Mensajes de solicitud y respuesta .....	23
Mensajes de eventos .....	27
Registro y anulación de registro .....	28
Mensajes <LDM_event> .....	28
Tipos de eventos .....	29
Acciones de Dominios lógicos Manager .....	32
Recursos y propiedades de Dominios lógicos Manager .....	34
Recurso de información de dominio (ldom_info) .....	35
Recurso de CPU (cpu) .....	36
Recurso MAU (mau) .....	38

Recurso de memoria ( <i>memory</i> ) .....	39
Recurso de SAN virtual ( <i>vsan</i> ) .....	39
Recurso de servidor de disco virtual ( <i>vds</i> ) .....	40
Recurso del volumen del servidor del disco virtual ( <i>vds_volume</i> ) .....	40
Recurso de disco ( <i>disk</i> ) .....	41
Recurso de conmutador virtual ( <i>vsw</i> ) .....	42
Recurso de red ( <i>network</i> ) .....	43
Recurso del concentrador de consola virtual ( <i>vcc</i> ) .....	44
Recurso de variable ( <i>var</i> ) .....	44
Recurso de dispositivo de E/S físico ( <i>physio_device</i> ) .....	45
Recurso de configuración SP ( <i>spconfig</i> ) .....	48
Recurso de configuración de directiva de DRM ( <i>policy</i> ) .....	49
Recurso de consola ( <i>console</i> ) .....	50
Migración de dominio .....	50
Esquemas XML .....	51
<b>4 Descubrimiento de Dominios lógicos Manager</b> .....	<b>53</b>
Descubrimiento de sistemas que ejecutan el Dominios lógicos Manager .....	53
Comunicación multidifusión .....	53
Formato del mensaje .....	53
▼ Cómo detectar la ejecución de Dominios lógicos Managers en la subred .....	54
<b>5 Uso de la API y del comando de información de dominio virtual</b> .....	<b>57</b>
Uso del comando de información de dominio virtual .....	57
Uso de la API de información de dominio virtual .....	57
<b>Índice</b> .....	<b>59</b>

## Cómo utilizar esta documentación

---

- **Visión general:** proporciona información y procedimientos detallados que describen las plantillas de Oracle VM Server for SPARC, la interfaz XML, la detección de Dominios lógicos Manager y las API de información de dominio virtual.
- **Destinatarios:** administradores del sistema que gestionan la virtualización en servidores SPARC.
- **Conocimientos necesarios:** los administradores del sistema de dichos servidores deben tener un conocimiento de trabajo de los sistemas UNIX y el sistema operativo Oracle Solaris (SO Oracle Solaris)

## Biblioteca de documentación del producto

La documentación y los recursos para este producto y los productos relacionados se encuentran disponibles en <http://www.oracle.com/technetwork/documentation/vm-sparc-194287.html>.

## Comentarios

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



# ◆◆◆ 1 C A P Í T U L O 1

## Desarrollo de plantillas, archivos XML y programas de Oracle VM Server for SPARC

---

### Desarrollo de plantillas, archivos XML y programas de Oracle VM Server for SPARC

En esta guía, se proporciona información sobre las siguientes funciones de Oracle VM Server for SPARC:

- **Utilidades de plantillas de Oracle VM Server for SPARC.** Use estas utilidades para crear plantillas que le permiten desplegar configuraciones de dominios. Consulte [Capítulo 2, Uso de Plantillas de Oracle VM Server for SPARC](#).
- **Interfaz XML de Oracle VM Server for SPARC.** Use esta interfaz XML para describir la configuración de los dominios lógicos. Consulte [Capítulo 3, Uso de la interfaz XML con el Dominios lógicos Manager](#).
- **API de detección de Dominios lógicos Manager.** Use esta API para escribir programas que pueden detectar qué dominios ejecutan Dominios lógicos Manager. Consulte [Capítulo 4, Descubrimiento de Dominios lógicos Manager](#).
- **API de información de dominio virtual.** Use esta API para escribir programas que proporcionan información sobre dominios virtuales. Consulte [Capítulo 5, Uso de la API y del comando de información de dominio virtual](#).

---

**Nota** - Las funciones que se describen en este manual se pueden utilizar con todas las plataformas compatibles de software y hardware de sistemas que se muestran en [Guía de instalación de Oracle VM Server for SPARC 3.4](#). Sin embargo, algunas funciones solo están disponibles en un subconjunto de plataformas de hardware y software compatibles del sistema. Para obtener información sobre estas excepciones, consulte “[Novedades de esta versión](#)” de [Notas de la versión de Oracle VM Server for SPARC 3.4](#) y [What's New in Oracle VM Server for SPARC Software](#) (<http://www.oracle.com/technetwork/server-storage/vm/documentation/sparc-whatnew-330281.html>).

---

## ◆◆◆ CAPÍTULO 2

# Uso de Plantillas de Oracle VM Server for SPARC

---

Este capítulo trata sobre los siguientes temas:

- [“Acerca de Plantillas de Oracle VM Server for SPARC” \[9\]](#)
- [“Instalación de las utilidades de Plantilla de Oracle VM Server for SPARC” \[10\]](#)
- [“Ciclo de vida de Plantilla de Oracle VM Server for SPARC” \[11\]](#)
- [“Ejemplos de Plantilla de Oracle VM Server for SPARC” \[14\]](#)

## Acerca de Plantillas de Oracle VM Server for SPARC

Los comandos de Plantillas de Oracle VM Server for SPARC permiten crear, desplegar y configurar Plantillas de Oracle VM Server for SPARC para sistemas SPARC. Estos comandos se basan en la especificación de plantilla OVF que incluye archivos de imagen de disco y un descriptor XML de propiedades que se encuentra en un archivo (.ova).

Estos comandos se ejecutan en el dominio de control de un sistema Oracle VM Server for SPARC. Además de ejecutar los comandos en una línea de comandos, se pueden ejecutar de manera completamente automática o desde otros programas como parte de un flujo de trabajo más grande.

Puede usar estas plantillas para desplegar y configurar solamente dominios invitados, dominios de E/S y dominios raíz. Sin embargo, no puede usar plantillas para desplegar y configurar dominios de E/S o dominios raíz que también sean el dominio `primary` ni desplegar y configurar dominios de servicio.

Los comandos de la Plantilla de Oracle VM Server for SPARC son los siguientes:

- `ovmtadm`: permite crear, configurar, desplegar y eliminar una plantilla iniciando los comandos `ovmcreate`, `ovmconfig` y `ovmdeploy`. Consulte la página del comando `man ovmtadm(1M)`.
- `ovmtconfig`: lleva a cabo acciones de configuración en un dominio mediante la transferencia de pares nombre-valor de propiedad a aplicaciones y procesos en el dominio de destino,

por ejemplo, mecanismos de configuración de SO Oracle Solaris y secuencias de comandos del primer inicio. Los pares nombre-valor de propiedad se establecen mediante el comando `ovmtprop`.

Además, puede usar este comando para montar hacia atrás un sistema de archivos ZFS de un dominio de modo que el dominio de control ejecute comandos directamente en esos sistemas de archivos. Este método incluye la capacidad de copiar archivos, instalar y actualizar paquetes del SO Oracle Solaris y llevar a cabo acciones de configuración. Consulte la página del comando `man ovmtconfig(1M)`.

- `ovmcreate`: crea una plantilla a partir de un dominio de Oracle VM Server for SPARC existente. Consulte la página del comando `man ovmtcreate(1M)`.
- `ovmtdeploy`: crea un dominio a partir de una plantilla de Oracle VM Server for SPARC. Consulte la página del comando `man ovmtdeploy(1M)`.
- `ovmtlibrary`: gestiona una base de datos y un repositorio basado en sistema de archivos para Plantillas de Oracle VM Server for SPARC mediante la organización de archivos y el almacenamiento, la recuperación y la edición de información en la base de datos. Consulte la página del comando `man ovmtlibrary(1M)`.
- `ovmtprop`: permite ver y establecer propiedades del SO Oracle Solaris en el dominio que se despliega desde una plantilla. La propiedad se especifica como un par nombre-valor. Este comando se llama desde otras secuencias de comandos y otros programas para llevar a cabo acciones de configuración que pueden ser controladas por propiedades. Consulte la página del comando `man ovmtprop(1M)`.

Para obtener información sobre cómo comenzar a usar Plantillas de Oracle VM Server for SPARC, consulte los siguientes blogs:

- [Deploying a Template \(Despliegue de una plantilla\)](https://blogs.oracle.com/cmt/en/entry/getting_started_with_ovm_templates) ([https://blogs.oracle.com/cmt/en/entry/getting\\_started\\_with\\_ovm\\_templates](https://blogs.oracle.com/cmt/en/entry/getting_started_with_ovm_templates))
- [Creating a Template \(Creación de una plantilla\)](https://blogs.oracle.com/cmt/en/entry/getting_started_with_ovm_templates1) ([https://blogs.oracle.com/cmt/en/entry/getting\\_started\\_with\\_ovm\\_templates1](https://blogs.oracle.com/cmt/en/entry/getting_started_with_ovm_templates1))
- [Using Templates on Oracle SuperCluster \(Uso de plantillas en Oracle SuperCluster\)](https://blogs.oracle.com/cmt/en/entry/getting_started_with_ovm_templates2) ([https://blogs.oracle.com/cmt/en/entry/getting\\_started\\_with\\_ovm\\_templates2](https://blogs.oracle.com/cmt/en/entry/getting_started_with_ovm_templates2))

## Instalación de las utilidades de Plantilla de Oracle VM Server for SPARC

Para instalar el paquete de software de utilidades de la Plantilla de Oracle VM Server for SPARC en el dominio de control, primero debe descargar e instalar el software de Oracle VM Server for SPARC 3.4 en el dominio de control. Consulte [Capítulo 2, “Instalación del software” de Guía de instalación de Oracle VM Server for SPARC 3.4](#).

A continuación, instale el paquete de software de utilidades de la Plantilla de Oracle VM Server for SPARC, `ovmtutils`.

```
# pkg install -v pkg:/system/ldoms/ovmtutils
```

## Ciclo de vida de Plantilla de Oracle VM Server for SPARC

En esta sección, se describe cada una de las etapas del proceso de creación de plantillas, las acciones que se realizan y cómo se usan las utilidades de la Plantilla de Oracle VM Server for SPARC para facilitar el proceso:

---

**Nota** - La creación y el desarrollo de plantillas con aplicaciones y secuencias de comandos de primer inicio es un proceso iterativo. Asegúrese de sincronizar todos los aspectos de la configuración mediante un sistema de gestión de código de origen, de modo de gestionar las secuencias de comandos y las propiedades.

---

A continuación, se describen las etapas del proceso de creación de plantillas, las acciones que se realizan y cómo se usan las utilidades de la Plantilla de Oracle VM Server for SPARC para facilitar el proceso:

1. **Creación de una plantilla.** Si bien hay plantillas genéricas incorporadas, puede crear una plantilla personalizada a partir de un dominio existente. Este dominio debe tener todos los componentes del sistema operativo, el software de las aplicaciones y otras utilidades que desee instalar.

Normalmente, el entorno se configura de la manera más completa posible, de modo que se necesite solo un pequeño número de acciones para finalizarlo. La configuración de los distintos aspectos del dominio, como la memoria, la CPU virtual, la red virtual y los discos, debe reflejar el despliegue deseado.

En esta etapa, debe crear una o varias secuencias de comandos del “primer inicio”. Incluya estas secuencias de comandos en el entorno que lleva a cabo la configuración final en función de las propiedades que usted proporcione. Asegúrese de registrar y describir estas propiedades en un archivo README para cada plantilla.

---

**Nota** - Si alguna de las secuencias de comandos de primer inicio accede a variables del dominio, asegúrese de que la utilidad `ovmtprop` esté instalada en el dominio invitado.

---

2. **Creación de una plantilla.** Antes de crear una plantilla, asegúrese de que el entorno del dominio de origen no esté configurado, de manera que se lo pueda configurar más adelante mediante acciones prescritas que con frecuencia son parte de las secuencias de comandos de primer inicio.

Por ejemplo, realice los siguientes pasos:

- Elimine las configuraciones específicas de aplicaciones que se volverán a crear más adelante.
- Use valores por defecto para los archivos de configuración.
- Exporte los zpools que no sean el sistema de archivos raíz para que los nuevos dominios puedan reconocerlos.
- Reverta el sistema operativo a un estado no configurado de modo que esté listo para aceptar propiedades de configuración en el primer inicio después del despliegue desde una plantilla. Ejecute el siguiente comando para eliminar las personalizaciones en el nivel del sitio, anular la configuración y detener el sistema operativo:

```
# sysconfig unconfigure -g system -s --include-site-profile --destructive
```

Después de realizar estos pasos, ejecute el comando `ovmcreate` para crear una plantilla desde un dominio.

---

**Nota** - Si ejecuta este comando y no se suministran propiedades en el momento del despliegue, el sistema ejecuta el instalador interactivo de Oracle Solaris en el inicio siguiente.

---

3. **Especificación del nombre de la plantilla.** Use una convención coherente para identificar la plantilla, por ejemplo, el formato siguiente:

```
technology.OS.application.architecture.build.ova
```

Por ejemplo, el siguiente nombre de plantilla corresponde a un dominio que ejecuta la compilación 2 del sistema operativo Oracle Solaris 11.2 en una plataforma SPARC y que ejecuta la versión 12.1.2 de WebLogic Server: `OVM_S11.2_WLS12.1.2_SPARC_B2.ova`.

4. **Distribución de la plantilla.** La plantilla es un único archivo con la extensión `.ova`. El archivo contiene las imágenes de disco comprimidas y los metadatos que se necesitan para el despliegue. La plantilla también contiene un archivo de manifiesto de totales de control de archivo de carga útil, que se pueden combinar con un total de control de archivo general para validar que el contenido no se haya modificado desde la distribución.

Para distribuir la plantilla puede usar servicios basados en la web o mantener un repositorio centralizado en lugar de duplicar las plantillas.

5. **Despliegue de la plantilla.** Como la plantilla captura solo los aspectos del sistema que ve el dominio de origen, debe saber cuáles son los servicios que tienen que estar presentes para que sea posible el despliegue de la plantilla.

Entre los servicios requeridos, se incluyen los siguientes:

- Uno o varios conmutadores virtuales con interfaces apropiadas a los que se puedan conectar las redes virtuales de la plantilla.

- Servicios de disco virtual.
- Servicios de consola.
- Suficiente CPU virtual y memoria para cubrir los requisitos de la plantilla.

Si bien la utilidad `ovmtdploy` puede sustituir muchos de estos valores de configuración, los valores mínimos que se proporcionan con la plantilla representan los requisitos básicos.

Puede usar la utilidad `ovmtdploy` para extraer, descomprimir y copiar automáticamente los discos virtuales en directorios de despliegue y compilar los diversos dispositivos virtuales que describa la plantilla.

Al alcanzar este paso ya puede iniciar el dominio, pero tal vez necesite realizar algunos pasos de configuración manual desde la consola del dominio antes de que el dominio pueda utilizar todas sus funciones.

6. **Configuración automática del dominio.** La configuración de un dominio que se crea por medio de una plantilla incluye varios tipos de acciones. Por ejemplo, puede especificar pares nombre-valor de propiedad para proporcionar secuencias de comandos de primer inicio con la información para la configuración. También puede realizar un montaje en segundo plano de los discos virtuales en el dominio de control para realizar acciones en los sistemas de archivos del dominio, como copiar archivos de configuración.

La utilidad `ovmtconfig` automatiza estas actividades de configuración de dominio y le permite indicar las acciones que desea que se realicen y las propiedades que desea usar para configurar un dominio mediante la especificación de una o varias secuencias de comandos y archivos de propiedades.

Para configurar el SO Oracle Solaris, la utilidad `ovmtconfig` realiza un montaje en segundo plano del sistema de archivos raíz del dominio y crea un archivo `sc_profile.xml` a partir de las secuencias de comandos y las propiedades de configuración proporcionadas. Este perfil permite que el SO Oracle Solaris se configure automáticamente en el primer inicio.

7. **Primera configuración.** Una vez que el SO Oracle Solaris se configura correctamente y se inicia por primera vez, debe configurar las aplicaciones instaladas. Durante la fase de configuración, la utilidad `ovmtconfig` transfiere la información de configuración al dominio desplegado mediante uno de los siguientes métodos:

- **Variables de dominio:** además de un archivo de propiedades local, puede definir variables de dominio mediante la ejecución de la utilidad `ovmtconfig` en el dominio de control. La utilidad `ovmtprop` puede posteriormente utilizar estas variables en el dominio invitado. Este método permite a las secuencias de comandos de primer inicio acceder a las propiedades de manera directa y proporciona información de configuración directamente al dominio invitado una vez que finaliza la configuración.

Por ejemplo, podría automatizar el cambio de un aspecto de configuración que no tenga acceso a la red mediante el uso de una secuencia de comandos de supervisión que ejecute `ovmtprop` en el dominio invitado y la ejecución de `ovmtconfig -v` desde el dominio de control.

- **Acción directa:** la utilidad `ovmtconfig` realiza un montaje en segundo plano de los sistemas de archivos del dominio invitado en el dominio de control y actúa directamente sobre los archivos y los sistemas de archivos. Las acciones pueden incluir la creación de archivos de configuración o la copia de binarios de sistema. Estas acciones se describen en las secuencias de comandos que proporciona a la utilidad `ovmtconfig`. Puede encontrar secuencias de comandos que llevan a cabo acciones de comandos en el directorio `/opt/ovmtutils/share/scripts`.

---

**Nota** - Estas acciones normalmente no incluyen procesos de primer inicio que estén diseñados para ejecutarse en el dominio invitado, ya que dichas acciones pueden afectar al dominio de control.

Use el comando `ovmtconfig -c` para especificar los comandos que desea ejecutar.

---



---

**Atención** - No use propiedades sin cifrar para transferir información confidencial al dominio, como contraseñas. Las propiedades diferentes de las que se usan para configurar el SO Oracle Solaris se transfieren al dominio como variables `ldm` en texto no cifrado. Estos valores de propiedad están visibles para un usuario del dominio de control que está autorizado para ejecutar los comandos `ldm` y para un usuario que ha iniciado sesión en el dominio desplegado.

---

En este punto, el dominio debería estar completamente configurado y en funcionamiento.

## Ejemplos de Plantilla de Oracle VM Server for SPARC

En esta sección, se muestran ejemplos de las siguientes tareas de la Plantilla de Oracle VM Server for SPARC:

- [Ejemplo 1, “Creación de Plantilla de Oracle VM Server for SPARC”](#)
- [Ejemplo 2, “Configuración de propiedades de Plantilla de Oracle VM Server for SPARC”](#)
- [Ejemplo 3, “Implementación de Plantillas de Oracle VM Server for SPARC”](#)
- [Ejemplo 4, “Gestión de la biblioteca de Plantilla de Oracle VM Server for SPARC”](#)

### **EJEMPLO 1** Creación de Plantilla de Oracle VM Server for SPARC

El comando `ovmtcreate` que se muestra a continuación crea una plantilla basada en el dominio `ldg1` llamada `ovmtcreate_example`. Observe que el nombre de la plantilla resultante tiene el sufijo `.ova`.

```

# ovmtcreate -d ldg1 -o ovmtcreate_example
...

STAGE 1 - EXAMINING SYSTEM AND ENVIRONMENT
-----
Performing platform & prerequisite checks
Checking user permissions
Checking for required packages
Checking for required services
Checking directory permissions

STAGE 2 - ANALYZING DOMAIN
-----
Retrieving and processing attributes
Checking domain state
Getting domain resource settings
Discovering network topology
Discovering disk topology

STAGE 3 - ARCHIVE CREATION
-----
Checking destination and current directory capacity
Compressing disk image
Creating XML configuration
Calculating manifest checksums
Creating archive file
Checking archive

PROCESS COMPLETED
-----
Started: Tue Aug 18 15:29:14 PDT 2015
Completed: Tue Aug 18 15:41:25 PDT 2015
Elapsed time: 0:12:11

```

## EJEMPLO 2 Configuración de propiedades de Plantilla de Oracle VM Server for SPARC

Puede usar las utilidades `ovmtconfig` y `ovmtprop` para especificar los valores de las propiedades de la Plantilla de Oracle VM Server for SPARC y el SO Oracle Solaris, respectivamente.

- El siguiente comando `ovmtconfig` realiza operaciones de configuración directamente en el sistema de archivos del dominio `ldg1`.

La opción `-c` especifica la secuencia de comandos de `/opt/ovmtutils/share/scripts/ovmt_s11_scprofile.sh` para configurar valores de propiedades. La opción `-p` indica valores específicos para las propiedades `com.oracle.solaris.network.ipaddr` y `com.oracle.solaris.system.computer-name`.

```

# ovmtconfig -v -d ldg1 -f -s \
-c /opt/ovmtutils/share/scripts/ovmt_s11_scprofile.sh \
-p com.oracle.solaris.network.ipaddr.0=10.153.118.211, \
com.oracle.solaris.system.computer-name=system1
...

```

```

STAGE 1/7 - EXAMINING SYSTEM AND ENVIRONMENT

```

```
-----
Checking operating system
Checking platform type
Checking user permissions
Checking packages
Checking host domain name
Checking host domain type
Checking services

STAGE 2/7 - PROCESSING COMMAND LINE OPTIONS
-----
Parsing individual properties
Creating consolidated properties list

STAGE 3/7 - ANALYZING TARGET DOMAIN
-----
Stopping domain ldg1
Analyzing domain disk topology for domain ldg1
Discovering 1 volumes for vDisks
Examining 1 backend devices
unbinding domain ldg1
Creating 1 virtual disks for back mount
Created virtual disk 0

STAGE 4/7 - PERFORMING BACKMOUNT
-----
Finding Solaris device for vdisks
Importing zpools for 1 Solaris devices
Detected conflicting zpool name, attempting rename
Getting boot file system for properties in 1 zpool
Setting properties in 1 zpools
Mounting ZFS file systems
Mounting ZFS found in zpool rpool_1

STAGE 5/7 - PERFORMING ACTIONS ON TARGET DOMAIN
-----

STAGE 6/7 - UNMOUNTING AND RESTORING DOMAIN STATE
-----
Rolling back commands DEBUG [20150819-07:02:42]: Rolling back 8 /usr/sbin/zfs unmount -f
rpool_1/ROOT/solaris/var
completed
```

```
STAGE 7/7 - SETTING TARGET DOMAIN ENVIRONMENT
```

```
-----
```

```
Checking 2 properties to set as domain variables
```

```
Process completed
```

- El siguiente comando `ovmtprop` especifica valores de propiedades del SO Oracle Solaris.

```
primary# ovmtprop set-prop com.oracle.solaris.system.computer-name=test ldg1
```

Use el comando `ldm list -l` para verificar que el valor de la propiedad `com.oracle.solaris.system.computer-name` sea `test`.

```
primary# ldm list -l ldg1
```

```
NAME          STATE      FLAGS    CONS    VCPU  MEMORY  UTIL  NORM  UPTIME
ldg1          active    -n----  5000    8     8G      0.0%  0.0%  23h 23m
```

```
..
```

```
VARIABLES
```

```
  auto-boot?=true
```

```
  boot-file=-k
```

```
  pm_boot_policy=disabled=0;ttfc=2000;ttmr=0;
```

```
VMAPI TO GUEST
```

```
  com.oracle.solaris.fmri.count=0
```

```
  com.oracle.solaris.system.computer-name=test
```

### EJEMPLO 3 Implementación de Plantillas de Oracle VM Server for SPARC

El comando `ovmtdeploy` que se muestra a continuación crea un dominio llamado `ldg1` mediante el uso de la Plantilla de Oracle VM Server for SPARC `ovmcreate_example.ova` en el directorio `/export/ovmtdeploy`.

```
# ovmtdeploy -d ldg1 -o /export/ovmtdeploy ovmcreate_example.ova
```

```
...
```

```
STAGE 1 - EXAMINING SYSTEM AND ENVIRONMENT
```

```
-----
Checking user privilege
Performing platform & prerequisite checks
Checking for required services
Named resourced available
```

```
STAGE 2 - ANALYZING ARCHIVE & RESOURCE REQUIREMENTS
```

```
-----
Checking .ova format and contents
Validating archive configuration
Checking sufficient resources present
WARNING: Virtual switch primary-vsw0 already exists
```

```
STAGE 3 - EXTRACTING ARCHIVE
```

```
-----
Extracting archive
Validating checksums
```

```
Decompressing disk image(s)

STAGE 4 - DEPLOYING DOMAIN
-----
Creating domain and adding resources
Validating deployment
Domain created:
```

La salida de `ldm list` muestra que se creó un nuevo dominio llamado `ldg1`.

```
# ldm list
NAME      STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
primary  active -n-cv- UART  8     40G    1.4%  1.1%  6d 2h 18m
ldg1     active -n---- 5000  8     8G     41%   38%  28s
```

#### **EJEMPLO 4**      Gestión de la biblioteca de Plantilla de Oracle VM Server for SPARC

El comando `ovmtlibrary` gestiona una base de datos y un repositorio basado en sistema de archivos para las Plantillas de Oracle VM Server for SPARC mediante la organización de archivos y el almacenamiento, la recuperación y la edición de información en la base de datos.

- El siguiente comando crea una biblioteca de plantillas en `export/user1/ovmtlibrary_example`:

```
# ovmtlibrary -c init -l /export/user1/ovmtlibrary_example
...
```

```
Init complete
```

- El siguiente comando almacena la plantilla `sol-11_2-ovm-2-sparc.ova` en la biblioteca `export/user1/ovmtlibrary_example`:

```
# ovmtlibrary -c store -d "ovmtlibrary example" -o http://system1.example.com/s11.2/templates/
sol-11_2-ovm-2-sparc.ova -l /export/user1/ovmtlibrary_example
...
```

```
Templates present in path "/export/user1/ovmtlibrary_example"
```

```
event id is 2
```

```
*****
converted 'http://system1.example.com/s11.2/templates/sol-11_2-ovm-2-sparc.ova' (646) ->
'http://system1.example.com/s11.2/templates/sol-11_2-ovm-2-sparc.ova' (UTF-8)
--2015-08-18 16:37:17-- http://system1.example.com/s11.2/templates/sol-11_2-ovm-2-
sparc.ova
Resolving system1.example.com (system1.example.com)... 10.134.127.18
Connecting to system1.example.com (system1.example.com)|10.134.127.18|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1018341888 (971M) [text/plain]
```

```
Saving to: '/export/user1/ovmtlibrary_example/repository/templates/1/1/sol-11_2-ovm-2-sparc.ova'
```

```
/export/user1/ovmtlibrary_example/repo 100%
[=====>]
 971.17M 6.05MB/s in 5m 37s
2015-08-18 16:42:55 (2.88 MB/s) - '/export/user1/ovmtlibrary_example/repository/
templates/1/1/sol-11_2-ovm-2-sparc.ova' saved
[1018341888/1018341888]
*****
Download complete
Extracting the ova file...
Extract complete
Decompress file System.img.gz
Store complete
```

- El siguiente comando muestra una lista de contenidos de la biblioteca `export/user1/ovmtlibrary_example`:

```
# ovmtlibrary -c list -l /export/user1/ovmtlibrary_example
...

Templates present in path "/export/user1/ovmtlibrary_example"
```

ID	Name	Version	Description	Date
1	sol-11_2-ovm-2-sparc	1	ovmtlibrary example	2015-08-18

- El siguiente comando muestra una lista detallada de la biblioteca `export/user1/ovmtlibrary_example`:

```
# ovmtlibrary -c list -i 1 -o -l /export/user1/ovmtlibrary_example
...

Templates present in path "/export/user1/ovmtlibrary_example"
```

ID	Name	Type	Path	Size(bytes)
1	sol-11_2-ovm-2-sparc.ova	ova	/export/user1/ovmtlibrary_example/repository/templates/1/1/sol-11_2-ovm-2-sparc.ova	1018341888
2	sol-11_2-ovm-sparc.ovf	ovf	/export/user1/ovmtlibrary_example/repository/templates/1/1/sol-11_2-ovm-sparc.ovf	3532
3	sol-11_2-ovm-sparc.mf	mf	/export/user1/ovmtlibrary_example/repository/templates/1/1/sol-11_2-ovm-sparc.mf	137
4	System.img	img	/export/user1/ovmtlibrary_example/repository/templates/1/1/System.img	21474836480

- El siguiente comando suprime el ID de plantilla 1 de la biblioteca `export/user1/ovmtlibrary_example`:

```
# ovmtlibrary -c delete -i 1 -l /export/user1/ovmtlibrary_example
```

## Uso de la interfaz XML con el Dominios lógicos Manager

---

En este capítulo, se explica el mecanismo de comunicación del lenguaje de marcas extensible (XML) a través del cual programas para el usuario externos pueden comunicarse mediante interfaz con el software del Oracle VM Server for SPARC. Se tratan estos temas básicos:

- “Transporte de XML” [21]
- “Protocolo XML” [22]
- “Mensajes de eventos” [27]
- “Acciones de Dominios lógicos Manager” [32]
- “Recursos y propiedades de Dominios lógicos Manager” [34]
- “Esquemas XML” [51]

### Transporte de XML

Los programas externos pueden usar el protocolo extensible de mensajería y comunicación de presencia (XMPP – RFC 3920) para comunicar con el Dominios lógicos Manager. El XMPP se admite para las conexiones locales y remotas y está activado de forma predeterminada. La interfaz XML solo admite la versión 1.2 del protocolo de seguridad de la capa de transporte (TLS).

Para desactivar una conexión remota, configure la propiedad de SMF `ldmd/xmpp_enabled` en `false` y reinicie Dominios lógicos Manager.

```
# svccfg -s ldom/ldmd setprop ldmd/xmpp_enabled=false
# svcadm refresh ldmd
# svcadm restart ldmd
```

---

**Nota** - La inhabilitación del servidor XMPP también evita la migración de dominio y la reconfiguración dinámica de memoria.

---

## Servidor XMPP

El Dominios lógicos Manager implementa un servidor XMPP que puede comunicarse con numerosas aplicaciones y bibliotecas de cliente XMPP disponibles. El Dominios lógicos Manager usa los siguientes mecanismos de seguridad:

- La seguridad de la capa de transporte para asegurar el canal de comunicación con el cliente.
- Autenticación simple y capa de seguridad (SASL) para la autenticación. `PLAIN` es el único mecanismo SASL admitido. Debe enviar un nombre de usuario y contraseña al servidor, de manera que le autorice antes de permitir las operaciones de seguimiento o gestión.

## Conexiones locales

Dominios lógicos Manager detecta si los clientes usuarios están en ejecución en el mismo dominio que él y, si es así, realiza un protocolo de enlace XMPP mínimo con el cliente. Específicamente, el paso de autenticación SASL después de la configuración de un canal seguro a través de TLS se omite. La autenticación y la autorización se realizan según las credenciales del proceso que implementa la interfaz del cliente.

Los clientes pueden elegir si implementar un cliente XMPP completo o simplemente ejecutar un analizador XML de transmisión, como el Simple API `libxm12` para analizador XML (SAX). En cualquier caso, el cliente tiene que administrar el protocolo de enlace XMPP hasta el punto de la negociación TLS. Consulte la especificación XMPP para conocer la secuencia necesaria.

## Protocolo XML

Después de completar la inicialización de la comunicación, los mensajes definidos en XML de Oracle VM Server for SPARC se envían a continuación. Existen dos tipos generales de mensajes XML:

- Mensajes de solicitud y de respuesta, que utilizan la etiqueta `<LDM_interface>`. Este tipo de mensaje XML se usa para los comandos de comunicación y obtener resultados del Dominios lógicos Manager, análogo a los comandos de ejecución usando la interfaz de línea de comandos (CLI). Esta etiqueta también se usa para el registro y anulación de registro de eventos.
- Los mensajes de evento usan la etiqueta `<LDM_event>`. Este tipo de mensaje XML se usa para informar de manera asincrónica de los eventos publicados por Dominios lógicos Manager.

## Mensajes de solicitud y respuesta

La interfaz XML en Oracle VM Server for SPARC tiene dos formatos diferentes:

- Un formato para enviar comandos a Dominios lógicos Manager.
- Otro formato para que Dominios lógicos Manager responda sobre el estado del mensaje entrante y las acciones solicitadas en ese mensaje.

Los dos formatos comparten muchas estructuras XML comunes, pero están separados en esta sección para entender mejor las diferencias entre ellos.

### Mensajes de solicitud

Una solicitud de XML entrante a Dominios lógicos Manager en el nivel más básico incluye una descripción de un solo comando que opera en un solo objeto. Las solicitudes más complicadas pueden manejar múltiples comandos y múltiples objetos por comando. El ejemplo siguiente muestra la estructura de un comando XML básico.

#### EJEMPLO 5 Formato de un solo comando operando en un solo objeto

```
<LDM_interface version="1.3">
  <cmd>
    <action>Place command here</action>
    <options>Place options for certain commands here</options>
    <arguments>Place arguments for certain commands here</arguments>
    <data version="3.0">
      <Envelope>
        <References/>
        <!-- Note a <Section> section can be here instead of <Content>
        <Content xsi:type="ovf:VirtualSystem_Type" id="Domain name">
          <Section xsi:type="ovf:ResourceAllocationSection_type">
            <Item>
              <rasd:OtherResourceType>LDom Resource Type</rasd:OtherResourceType>
              <gprop:GenericProperty
                key="Property name">Property Value</gprop:GenericProperty>
            </Item>
          </Section>
        </Content>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>
```

## Etiqueta <LDM\_interface>

Todos los comandos enviados al Dominios lógicos Manager deben empezar por la etiqueta <LDM\_interface>. Cualquier documento enviado al Dominios lógicos Manager debe tener solo una etiqueta <LDM\_interface> contenida en el mismo. La etiqueta <LDM\_interface> debe incluir un atributo de versión, como se muestra en el [Ejemplo 5, “Formato de un solo comando operando en un solo objeto”](#).

## La etiqueta <cmd>

En la etiqueta <LDM\_interface>, el documento debe contener al menos una etiqueta <cmd>. Cada sección <cmd> debe tener solo una etiqueta <action>. Use la etiqueta <action> para describir qué comando ejecutar. Cada etiqueta <cmd> debe incluir al menos una etiqueta <data> para describir los objetos en los que debe operar el comando.

La etiqueta <cmd> también puede tener una etiqueta <options>, que se usa para las opciones y etiquetas que están asociadas con algunos comandos. Los siguientes comandos usan las opciones:

- El comando `ldm remove-domain` puede usar la opción `-a`.
- El comando `ldm bind-domain` puede usar la opción `-f`.
- El comando `ldm add-vdsdev` puede usar la opción `-f`.
- El comando `ldm cancel-operation` puede usar la opción `migration` o `reconf`.
- El comando `ldm add-spconfig` puede usar la opción `-r autosave-name`.
- El comando `ldm remove-spconfig` puede usar la opción `-r`.
- El comando `ldm list-spconfig` puede usar la opción `-r [autosave-name]`.
- El comando `ldm stop-domain` puede usar las siguientes etiquetas para establecer los argumentos del comando:
  - <force> representa la opción `-f`.
  - <halt> representa la opción `-h`.
  - <message> representa la opción `-m`.
  - <quick> representa la opción `-q`.
  - <reboot> representa la opción `-r`.
  - <timeout> representa la opción `-t`.

Tenga en cuenta que las etiquetas no deben tener ningún valor de contenido. Sin embargo, las opciones `-t` y `-m` deben tener un valor no nulo, por ejemplo, `<timeout>10</timeout>` o `<message>Shutting down now</message>`.

En el siguiente fragmento de ejemplo de XML, se muestra cómo pasar una solicitud de reinicio con un mensaje de reinicio al comando `ldm stop-domain`:

```

<action>stop-domain</action>
<arguments>
  <reboot/>
  <message>my reboot message</message>
</arguments>

```

## La etiqueta <data>

Cada sección <data> contiene una descripción de un objeto pertinente al comando especificado. El formato de la sección <data> se basa en la porción del esquema XML del borrador de especificación del formato abierto de virtualización (OVF). Este esquema define una sección <Envelope> que contiene una etiqueta <References> (no usada por Oracle VM Server for SPARC) y secciones <Content> y <Section>.

Para Oracle VM Server for SPARC, la sección <Content> se usa para identificar y describir un dominio especial. El nombre de dominio en el id= attribute del nodo <Content> identifica el dominio. En la sección <Content> hay una o varias secciones <Section> que describen los recursos del dominio según lo necesita un comando específico.

Si solo necesita identificar un nombre de dominio, no necesita usar las etiquetas <Section>. Por el contrario, si no se necesita ningún identificador de dominio para el comando, debe incluir una sección <Section>, que describa los recursos necesarios para el comando, fuera de la sección <Content>, pero dentro de la sección <Envelope>.

Una sección <data> no necesita contener una etiqueta <Envelope> en casos donde la información del objeto puede deducirse. Esta situación afecta sobre todo al seguimiento de todos los objetos aplicables a una acción, y a las solicitudes de registro y eliminación del registro de los eventos.

Dos tipos OVF adicionales permiten el uso del esquema de la especificación OVF para definir correctamente todos los tipos de objetos:

- Etiqueta <gprop:GenericProperty>
- Etiqueta <Binding>

La etiqueta <gprop:GenericProperty> se maneja cualquier propiedad del objeto para la que la especificación OVF no tiene una definición. El nombre de la propiedad se define en el atributo key= del nodo y el valor de la propiedad son los contenidos del nodo. La etiqueta <binding> se usa en la salida del comando `ldm list-bindings` para definir los recursos que están enlazados a otros recursos.

## Mensajes de respuesta

Una respuesta XML saliente corresponde estrechamente con la estructura de solicitud entrante en términos de los comandos y objetos incluidos, con adición de una sección <Response> para

cada objeto y comando especificado, así como una sección general <Response> para la solicitud. Las secciones <Response> proporcionan información sobre el estado y el mensaje. El siguiente ejemplo muestra la estructura de una respuesta a una solicitud XML básica.

**EJEMPLO 6** Formato de una respuesta a un comando único operando en un objeto único

```
<LDM_interface version="1.3">
  <cmd>
    <action>Place command here</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <!-- Note a <Section> section can be here instead of <Content>
        <Content xsi:type="ovf:VirtualSystem_Type" id="Domain name">
          <Section xsi:type="ovf:ResourceAllocationSection_type">
            <Item>
              <rasd:OtherResourceType>
                LDom Resource Type
              </rasd:OtherResourceType>
              <gprop:GenericProperty
                key="Property name">
                Property Value
              </gprop:GenericProperty>
            </Item>
          </Section>
          <!-- Note: More <Section>
        </Content>
      </Envelope>
      <response>
        <status>success or failure</status>
        <resp_msg>Reason for failure</resp_msg>
      </response>
    </data>

    <response>
      <status>success or failure</status>
      <resp_msg>Reason for failure</resp_msg>
    </response>
  </cmd>

  <response>
    <status>success or failure</status>
    <resp_msg>Reason for failure</resp_msg>
  </response>
</LDM_interface>
```

**Respuesta general**

Esta sección <response>, que es el descendiente directo de la sección <LDM\_interface>, indica el éxito o fallo general de toda la solicitud. A menos que el documento XML esté mal formado, la sección <response> incluye solo una etiqueta <status>. Si este estado de respuesta indica un resultado correcto, todos los comandos en todos los objetos se han efectuado correctamente. Si este estado de respuesta es un fallo y no hay etiqueta <resp\_msg>, entonces uno de los comandos

incluidos en la solicitud original falla. La etiqueta `<resp_msg>` se usa solo para describir algún problema con el mismo documento XML.

## Respuesta de comando

La sección `<response>` bajo la sección `<cmd>` alerta al usuario del éxito o fallo de este comando particular. La etiqueta `<status>` muestra si ese comando se ejecuta correctamente o falla. Como con la respuesta general, si el comando falla, la sección `<response>` incluye solo una etiqueta `<resp_msg>` si los contenidos de la sección `<cmd>` de la solicitud está mal formada. En caso contrario, el estado de fallo significa que uno de los objetos contra el que se ha ejecutado el comando ha provocado un fallo.

## Respuesta de objeto

Finalmente, cada sección `<data>` en la sección `<cmd>` también tiene una sección `<response>`. Esta sección muestra si el comando que se ejecuta en este objeto específico es satisfactorio o falla. Si el estado de la respuesta es `SUCCESS`, no hay etiqueta `<resp_msg>` en la sección `<response>`. Si el estado es `FAILURE`, hay una o más etiquetas `<resp_msg>` en el campo `<response>`, en función de los errores detectados al ejecutar el comando parra ese objeto. Los errores de objeto pueden derivar de problemas detectados cuando se ha ejecutado el comando, o el objeto está mal formado o es desconocido.

Además de la sección `<response>`, la sección `<data>` puede contener otra información. Esta información está en el mismo formato que el campo entrante `<data>`, que describe el objeto que ha provocado el fallo. Consulte [“La etiqueta `<data>`” \[25\]](#). Esta información adicional es especialmente útil en los siguientes casos:

- Cuando un comando falla contra una sección especial `<data>` pero pasa cualquier sección adicional `<data>`.
- Cuando una sección `<data>` vacía se pasa en un comando y falla para algunos comandos pero pasa para otros.

## Mensajes de eventos

En lugar de esperar respuesta, puede suscribirse para recibir notificaciones de eventos de determinados cambios de estado que suceden. Hay tres tipos de eventos a los que puede suscribirse, individual o colectivamente. Consulte [“Tipos de eventos” \[29\]](#) para conocer todos los detalles.

## Registro y anulación de registro

Use un mensaje `<LDM_interface>` para registrarse para eventos. Consulte [“Etiqueta `<LDM\_interface>`” \[24\]](#). La etiqueta `<action>` detalla el tipo de evento para el que desea registrarse o eliminar el registro y la sección `<data>` se deja vacía.

### EJEMPLO 7 Ejemplo de mensaje de solicitud de registro de evento

```
<LDM_interface version="1.3">
  <cmd>
    <action>reg-domain-events</action>
    <data version="3.0"/>
  </cmd>
</LDM_interface>
```

El Dominios lógicos Manager responde con un mensaje de respuesta `<LDM_interface>` que indica si el registro o la eliminación de registro ha tenido un resultado satisfactorio.

### EJEMPLO 8 Ejemplo de mensaje de respuesta de registro de evento

```
<LDM_interface version="1.3">
  <cmd>
    <action>reg-domain-events</action>
    <data version="3.0"/>
    <response>
      <status>success</status>
    </response>
  </data>
  <response>
    <status>success</status>
  </response>
</cmd>
<response>
  <status>success</status>
</response>
</LDM_interface>
```

La cadena de acción para cada tipo de evento se enumera en la subsección de eventos.

## Mensajes `<LDM_event>`

Los mensajes de evento tienen el mismo formato que un mensaje entrante `<LDM_interface>` con la excepción que la etiqueta de inicio para el mensaje es `<LDM_event>`. La etiqueta `<action>` del mensaje es la acción que se ha realizado para accionar el evento. La sección `<data>` del mensaje describe el objeto asociado con el evento; los detalles dependen del tipo de evento que se ha producido.

**EJEMPLO 9** Ejemplo, notificación <LDM\_event>

```

<LDM_event version='1.1'>
  <cmd>
    <action>Event command here</action>
    <data version='3.0'>
      <Envelope
        <References/>
        <Content xsi:type='ovf:VirtualSystem_Type' ovf:id='ldg1' />
        <Section xsi:type="ovf:ResourceAllocationSection_type">
          <Item>
            <rasd:OtherResourceType>LDom Resource Type</rasd:OtherResourceType>
            <gprop:GenericProperty
              key="Property name">Property Value</gprop:GenericProperty>
          </Item>
        </Section>
      </Envelope>
    </data>
  </cmd>
</LDM_event>

```

## Tipos de eventos

Puede suscribirse a los siguientes tipos de evento:

- Eventos de dominio
- Eventos de hardware
- Eventos de progreso
- Eventos de recursos

Todos los eventos corresponden a los subcomandos `ldm`.

## Eventos de dominio

Los eventos de dominio describen las acciones que pueden realizarse directamente en un dominio. A continuación, se muestran los eventos de dominio que se pueden especificar en la etiqueta <action> del mensaje <LDM\_event>:

- `add-domain`
- `bind-domain`
- `domain-reset`
- `migrate-domain`
- `panic-domain`
- `remove-domain`
- `start-domain`

- stop-domain
- unbind-domain

Estos eventos siempre contienen *solo* una etiqueta <Content> en la sección <data> de OVF que describe el dominio en el que se ha producido el evento. Para registrar para los eventos del dominio, envíe un mensaje <LDM\_interface con la etiqueta <action configurada como reg-domain-events. Para eliminar el registro para estos eventos, envíe un mensaje <LDM\_interface con la etiqueta <action configurada como unreg-domain-events.

## Eventos de hardware

Los eventos de hardware pertenecen al cambio del hardware del sistema físico. En el caso del software Oracle VM Server for SPARC, los únicos cambios de hardware que pueden realizarse son los del procesador de servicio (SP) cuando se agrega, se elimina o se establece la configuración de un SP. Actualmente, los únicos tres eventos para este tipo son:

- add-spconfig
- set-spconfig
- remove-spconfig

Los eventos de hardware contienen *solo* una etiqueta <Section en la sección <data> de OVF que describe qué configuración del SP para qué evento se está produciendo. Para registrarse para estos eventos, envíe un mensaje <LDM\_interface con la etiqueta <action configurada como reg-hardware-events. Para eliminar el registro para estos eventos, envíe un mensaje <LDM\_interface con la etiqueta <action configurada como unreg-hardware-events.

## Eventos de progreso

Los eventos de progreso se expiden para comandos de ejecución larga, como una migración de dominio. Estos eventos indican la cantidad de progreso que se ha realizado durante la vida del comando. En este momento, solo se indica el evento migration-process.

Los eventos de progreso siempre contienen solo una etiqueta <Section en la sección <data> de OVF que describe la configuración del SP que se ve afectada por el evento. Para registrarse para estos eventos, envíe un mensaje <LDM\_interface> con la etiqueta <action> fijada en reg-hardware-events. Para eliminar el registro para estos eventos, envíe un mensaje <LDM\_interface con la etiqueta <action configurada como unreg-hardware-events.

La sección <data> de un evento de progreso consiste en una sección <content> que describe el dominio afectado. Esta sección <content> usa una etiqueta ldom\_info <Section> para actualizar el progreso. Las siguientes propiedades genéricas se muestran en la sección ldom\_info:

- --progress – Porcentaje del progreso realizado por el comando

- `--status` – Estado del comando, que puede ser continuo, error o realizado
- `--source` – Equipo que está informando del progreso

## Eventos de recursos

Los eventos de recursos se producen cuando los recursos se agregan, se eliminan, o cambian en cualquier dominio. La sección `<data>` para algunos de estos eventos contiene la etiqueta `<Content` con una etiqueta `<Section` que proporciona un nombre de servicio en la sección `<data>` de OVF.

A continuación, se muestran los eventos que se pueden especificar en la etiqueta `<action>` del mensaje `<LDM_event>`:

- `add-vdiskserverdevice`
- `remove-vdiskserverdevice`
- `set-vdiskserverdevice`
- `remove-vdiskserver`
- `set-vconscon`
- `remove-vconscon`
- `set-vswitch`
- `remove-vswitch`

Los siguientes eventos de recursos siempre contienen *solo* la etiqueta `<Content` en la sección `<data>` de OVF que describe el dominio en el que se ha producido el evento:

- `add-vcpu`
- `add-crypto`
- `add-memory`
- `add-io`
- `add-variable`
- `add-vconscon`
- `add-vdisk`
- `add-vdiskserver`
- `add-vnet`
- `add-vsan`
- `add-vswitch`
- `remove-crypto`
- `remove-io`
- `remove-memory`
- `remove-variable`

- `remove-vcpu`
- `remove-vdisk`
- `remove-vnet`
- `set-crypto`
- `set-memory`
- `set-variable`
- `set-vconsole`
- `set-vcpu`
- `set-vdisk`
- `set-vnet`

Para registrarse para los eventos de recursos, envíe un mensaje `<LDM_interface` con la etiqueta `<action` configurada como `reg-resource-events`. La eliminación de un registro para estos eventos requiere un mensaje `<LDM_interface` con la etiqueta `<action` configurada como `unreg-resource-events`.

## Todos los eventos

También se puede registrar para los tres tipos de eventos sin tener que registrarse para cada uno individualmente. Para registrarse para los tres tipos de eventos simultáneamente, envíe un mensaje `<LDM_interface` con la etiqueta `<action` configurada como `reg-all-events`. La eliminación de un registro para estos eventos requiere un mensaje `<LDM_interface` con la etiqueta `<action` configurada como `unreg-all-events`.

## Acciones de Dominios lógicos Manager

Los comandos especificados en la etiqueta `<action>`, con la excepción de los comandos `--events`, corresponden a los de la interfaz de línea de comandos `ldm`. Para más detalles sobre los subcomandos `ldm`, véase la página de comando `man ldm(1M)`.

---

**Nota** - La interfaz XML no admite el verbo o comando no los alias de comando que admite la CLI de Dominios lógicos Manager.

---

Las cadenas admitidas en la etiqueta `<action>` son las siguientes:

- `add-domain`
- `add-io`
- `add-mau`

- add-memory
- add-spconfig
- add-variable
- add-vconscon
- add-vcpu
- add-vdisk
- add-vdiskserver
- add-vdiskserverdevice
- add-vhba
- add-vnet
- add-vsan
- add-vswitch
- bind-domain
- cancel-operation
- list-bindings
- list-constraints
- list-dependencies
- list-devices
- list-domain
- list-hba
- list-rsrc-group
- list-services
- list-spconfig
- list-variable
- list-vmapi
- migrate-domain
- reg-all-events
- reg-domain-events
- reg-hardware-events
- reg-resource-events
- remove-domain
- remove-io
- remove-mau
- remove-memory
- remove-reconf
- remove-spconfig
- remove-variable

- remove-vconscon
- remove-vcpu
- remove-vdisk
- remove-vdiskserver
- remove-vdiskserverdevice
- remove-vhba
- remove-vmapi
- remove-vnet
- remove-vsan
- remove-vswitch
- rescan-vhba
- set-domain
- set-mau
- set-memory
- set-spconfig
- set-variable
- set-vconscon
- set-vconsole
- set-vcpu
- set-vhba
- set-vmapi
- set-vnet
- set-vswitch
- start-domain
- stop-domain
- unbind-domain
- unreg-all-events
- unreg-domain-events
- unreg-hardware-events
- unreg-resource-events

## Recursos y propiedades de Dominios lógicos Manager

En esta sección, se proporcionan ejemplos de los recursos y las propiedades de Dominios lógicos Manager que se pueden definir para cada uno de esos recursos. Los recursos y las propiedades se muestran en **negrita** en los ejemplos de XML. Estos ejemplos muestran los

recursos, no la salida enlazada. La salida de restricción puede usarse para crear una entrada para las acciones de Dominios lógicos Manager, excepto la salida de migración de dominio. Consulte [“Migración de dominio” \[50\]](#). Cada recurso se define en una sección OVF `<Section>` y es especificado por una etiqueta `<rasd:OtherResourceType>`.

## Recurso de información de dominio (1dom\_info)

El ejemplo siguiente muestra las propiedades opcionales del recurso 1dom\_info:

**EJEMPLO 10** Ejemplo, salida SML 1dom\_info

El ejemplo siguiente muestra los valores especificados para varias propiedades 1dom\_info, como uuid, hostid y Address.

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="primary">
    <Section xsi:type="ovf:ResourceAllocationSection_type">
      <Item>
        <rasd:OtherResourceType>1dom_info</rasd:OtherResourceType>
        <uuid>c2c3d93b-a3f9-60f6-a45e-f35d55c05fb6</uuid>
        <rasd:Address>00:03:ba:d8:ba:f6</rasd:Address>
        <gprop:GenericProperty key="hostid">83d8baf6</gprop:GenericProperty>
        <gprop:GenericProperty key="master">plum</gprop:GenericProperty>
        <gprop:GenericProperty key="failure-policy">reset</gprop:GenericProperty>
        <gprop:GenericProperty key="extended-mapin-space">on</gprop:GenericProperty>
        <gprop:GenericProperty key="progress">45%</gprop:GenericProperty>
        <gprop:GenericProperty key="status">ongoing</gprop:GenericProperty>
        <gprop:GenericProperty key="source">system1</gprop:GenericProperty>
        <gprop:GenericProperty key="rc-add-policy"></gprop:GenericProperty>
        <gprop:GenericProperty key="perf-counters">global</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

El recurso 1dom\_info siempre está contenido en una sección `<Content>`. Las siguientes propiedades en el recurso 1dom\_info son opcionales:

- Etiqueta `<uuid>`, que especifica el UUID del dominio.
- `<rasd:Address>`, que especifica la dirección MAC que se debe asignar a un dominio.
- Etiqueta `<gprop:GenericProperty key="extended-mapin-space">`, que especifica si el espacio de asignación extendido está activado (`on`) o desactivado (`off`) para ese dominio. El valor predeterminado es `off`.
- Etiqueta `<gprop:GenericProperty key="failure-policy">`, que especifica cómo deben comportarse los dominios esclavos si el dominio maestro falla. El valor predeterminado es `ignore`. A continuación se incluyen los valores de propiedad válidos:
  - `ignore` ignora los fallos del dominio maestro (no afecta a los dominios esclavos).

- `panic` se genera el mensaje de error grave en cualquier dominio esclavo cuando falla el dominio maestro.
  - `reset` se restablece cualquier dominio esclavo cuando falla el dominio maestro.
  - `stop` se para cualquier dominio esclavo cuando falla el dominio maestro.
  - Etiqueta `<gprop:GenericProperty key="hostid">`, que especifica el ID del host que se debe asignar al dominio.
  - Etiqueta `<gprop:GenericProperty key="master">`, que especifica hasta cuatro nombres de dominio maestro separados por comas.
  - Etiqueta `<gprop:GenericProperty key="progress">`, que especifica el porcentaje de progreso realizado por el comando.
  - Etiqueta `<gprop:GenericProperty key="source">`, que especifica el equipo que informa el progreso del comando.
  - Etiqueta `<gprop:GenericProperty key="status">`, que especifica el estado del comando (`done`, `failed` o `ongoing`).
  - Etiqueta `<gprop:GenericProperty key="rc-add-policy">`, que especifica si se deben activar o desactivar las operaciones de virtualización de SR-IOV y E/S directa para cualquier complejo raíz que se pueda agregar al dominio especificado. Los valores válidos son `iov` y ningún valor (`rc-add-policy=`).
  - Etiqueta `<gprop:GenericProperty key="perf-counters">`, que especifica los conjuntos de registro de rendimiento a los que se debe acceder (`global`, `htstrand`, `strand`).
- Si la plataforma no tiene la capacidad de acceso de rendimiento, el valor de la propiedad `perf-counters` se ignora.

## Recurso de CPU (`cpu`)

Tenga en cuenta que la propiedad de unidades de asignación, `<rasd:AllocationUnits>`, para el recurso `cpu` siempre especifica el número de CPU virtuales y no el número de núcleos.

Un recurso `cpu` siempre está contenida en una sección `<Content>`.

**EJEMPLO 11** Salida de sección XML `cpu` del comando `ldm list-bindings`

En el ejemplo siguiente, se muestra la salida XML para la sección `<cpu>` mediante la ejecución del comando `ldm list-bindings`.

```
<?xml version="1.0"?>
<LDM_interface
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ovf="./schemas/envelope"
xmlns:rasd="./schemas/CIM_ResourceAllocationSettingData"
xmlns:vssd="./schemas/CIM_VirtualSystemSettingData"
xmlns:gprop="./schemas/GenericProperty"
```

```

xmlns:bind="./schemas/Binding"
version="1.3"
xsi:noNamespaceSchemaLocation="./schemas/combined-v3.xsd">
  <cmd>
    <action>list-bindings</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="primary">
          <Section xsi:type="ovf:ResourceAllocationSection_Type">
            <Item>
              <rasd:OtherResourceType>lom_info</rasd:OtherResourceType>
              <uuid>1e04cdbc-472a-e8b9-ba4c-d3eee86e7725</uuid>
              <rasd:Address>00:21:28:f5:11:6a</rasd:Address>
              <gprop:GenericProperty key="hostid">0x8486632a</gprop:GenericProperty>
              <failure-policy>fff</failure-policy>
              <wcore>0</wcore>
              <extended-mapin-space>0</extended-mapin-space>
              <cpu-arch>native</cpu-arch>
              <rc-add-policy/>
              <gprop:GenericProperty key="state">active</gprop:GenericProperty>
            </Item>
          </Section>
          <Section xsi:type="ovf:VirtualHardwareSection_Type">
            <Item>
              <rasd:OtherResourceType>cpu</rasd:OtherResourceType>
              <rasd:AllocationUnits>8</rasd:AllocationUnits>
              <bind:Binding>
                <Item>
                  <rasd:OtherResourceType>cpu</rasd:OtherResourceType>
                  <gprop:GenericProperty key="vid">0</gprop:GenericProperty>
                  <gprop:GenericProperty key="pid">0</gprop:GenericProperty>
                  <gprop:GenericProperty key="cid">0</gprop:GenericProperty>
                  <gprop:GenericProperty key="strand_percent">100</gprop:GenericProperty>
                  <gprop:GenericProperty key="util_percent">1.1%</gprop:GenericProperty>
                  <gprop:GenericProperty key="normalized_utilization">0.1%</gprop:
GenericProperty>
                </Item>
              </Section>
            </Content>
          </Envelope>
        </data>
      </cmd>
    </LDM_interface>

```

**EJEMPLO 12** Salida de sección XML cpu del comando `ldm list-domain`

En el ejemplo siguiente, se muestra la salida XML para la sección `<cpu>` mediante la ejecución del comando `ldm list-domain`.

```

<?xml version="1.0"?>
<LDM_interface
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ovf="./schemas/envelope"
xmlns:rasd="./schemas/CIM_ResourceAllocationSettingData"
xmlns:vssd="./schemas/CIM_VirtualSystemSettingData"
xmlns:gprop="./schemas/GenericProperty"
xmlns:bind="./schemas/Binding"
version="1.3"
xsi:noNamespaceSchemaLocation="./schemas/combined-v3.xsd">

```

```

<cmd>
  <action>list-domain</action>
  <data version="3.0">
    <Envelope>
      <References/>
      <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="primary">
        <Section xsi:type="ovf:ResourceAllocationSection_Type">
          <Item>
            <rasd:OtherResourceType>ldom_info</rasd:OtherResourceType>
            <gprop:GenericProperty key="state">active</gprop:GenericProperty>
            <gprop:GenericProperty key="flags">-n-cv-</gprop:GenericProperty>
            <gprop:GenericProperty key="utilization">0.7%</gprop:GenericProperty>
            <gprop:GenericProperty key="uptime">3h</gprop:GenericProperty>
            <gprop:GenericProperty key="normalized_utilization">0.1%</gprop:
GenericProperty>
          </Item>
        </Section>
      </Content>
    </Envelope>
  </data>
</cmd>
</LDM_interface>

```

## Recurso MAU (mau)

Un recurso `mau` siempre está contenido en una sección `<Content>`. La única propiedad es la etiqueta `<rasd:AllocationUnits>`, que significa el número de MAU u otras unidades criptográficas.

---

**Nota** - El recurso `mau` es cualquier unidad criptográfica admitida en un servidor admitido. Actualmente, las dos unidades criptográficas admitidas son unidad aritmética modular (MAU) y el Control Word Queue (CWQ).

---

### EJEMPLO 13 Ejemplo de XML `mau`

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>mau</rasd:OtherResourceType>
        <rasd:AllocationUnits>1</rasd:AllocationUnits>
      </Item>
    </Section>
  </Content>
</Envelope>

```

## Recurso de memoria (memory)

Un recurso de memoria siempre está contenido en una sección <Content>. La única propiedad es la etiqueta <rasd:AllocationUnits>, que significa la cantidad de memoria.

### EJEMPLO 14 Ejemplo de XML de memory

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>memory</rasd:OtherResourceType>
        <rasd:AllocationUnits>4G</rasd:AllocationUnits>
      </Item>
    </Section>
  </Content>
</Envelope>
```

## Recurso de SAN virtual (vsan)

Un recurso de SAN virtual (vsan) puede encontrarse en una sección <Content>. Debe usar las etiquetas <gprop:GenericProperty> con las siguientes claves:

- service\_name: nombre de SAN virtual
- vsan\_iport: puerto de iniciador de la SAN virtual

Debe haber un elemento que contenga una propiedad vol\_name única en la sección vsan\_volume. Esta propiedad vol\_name debe establecerse en \*.

### EJEMPLO 15 Ejemplo de XML de vsan

```
<Envelope>
  <References/>
  <Section xsi:type="ovf:VirtualHardwareSection_Type">
    <Item>
      <rasd:OtherResourceType>vsan</rasd:OtherResourceType>
      <gprop:GenericProperty key="service_name">vs1</gprop:GenericProperty>
      <gprop:GenericProperty key="vsan_iport">
        /pci@400/pci@1/pci@0/pci@8/SUNW,emlxs@0/fp@0,0</gprop:GenericProperty>
      <bind:Binding>
        <Item>
          <rasd:OtherResourceType>vsan_volume</rasd:OtherResourceType>
          <gprop:GenericProperty key="vol_name">*</gprop:GenericProperty>
        </Item>
      </bind:Binding>
    </Item>
  </Section>
</Envelope>
```

```
</Section>
</Envelope>
```

## Recurso de servidor de disco virtual (vds)

Un recurso de servidor de disco virtual (vds) puede estar en una sección <Content> como parte de la descripción de un dominio, o puede aparecer por sí misma en la sección <Envelope>. La única propiedad es la etiqueta <gprop:GenericProperty> con una clave de service\_name, que contiene el nombre del recurso vds que se describe.

**EJEMPLO 16** Ejemplo de XML de vds

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vds</rasd:OtherResourceType>
        <gprop:GenericProperty
          key="service_name">vdstmp</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

## Recurso del volumen del servidor del disco virtual (vds\_volume)

Un recurso vds\_volume puede estar en una sección <Content> contenido como parte de la descripción del dominio, o puede aparecer por sí mismo en una sección <Envelope>. Tiene que tener las etiquetas <gprop:GenericProperty> con las siguientes claves:

- vol\_name – Nombre del volumen
- service\_name – Nombre del servidor de disco virtual al que está enlazado el volumen
- block\_dev – Nombre de archivo o dispositivo que se ha de asociar con este volumen

De manera opcional, un recurso vds\_volume también puede tener las siguientes propiedades:

- vol\_opts – Una o varias de las siguientes, separadas por comas, con una cadena: {ro,slice,excl}
- mpgroup – Nombre del grupo de ruta múltiple (conmutación por error)

**EJEMPLO 17** Ejemplo de XML vds\_volume

```

<Envelope>
  <References/>
  <Section xsi:type="ovf:VirtualHardwareSection_Type">
    <Item>
      <rasd:OtherResourceType>vds_volume</rasd:OtherResourceType>
      <gprop:GenericProperty key="vol_name">vdsdev0</gprop:GenericProperty>
      <gprop:GenericProperty key="service_name">primary-vds0</gprop:GenericProperty>
      <gprop:GenericProperty key="block_dev">
        opt/SUNWldm/domain_disks/testdisk1</gprop:GenericProperty>
      <gprop:GenericProperty key="vol_opts">ro</gprop:GenericProperty>
      <gprop:GenericProperty key="mpgroup">mpgroup-name</gprop:GenericProperty>
    </Item>
  </Section>
</Envelope>

```

## Recurso de disco (disk)

Un recurso de `disk` siempre está contenido en una sección `<Content>`. Tiene que tener las etiquetas `<gprop:GenericProperty>` con las siguientes claves:

- `vdisk_name` – Nombre del disco virtual
- `service_name` – Nombre del servidor de disco virtual al que está enlazado el disco virtual
- `vol_name` – Dispositivo del servicio de disco virtual al que debe asociarse este disco virtual

Opcionalmente, el recurso `disk` también puede tener la propiedad `timeout`, que es el valor de tiempo de espera en segundos para el establecimiento de una conexión entre un cliente de disco virtual (`vdv`) y un servidor de disco virtual (`vds`). Si hay múltiples rutas de disco virtual (`vdisk`), `vdv` puede intentar conectarse a un `vds` diferente. El tiempo de espera garantiza que se establezca una conexión a cualquier `vds` en la cantidad de tiempo especificada.

**EJEMPLO 18** Ejemplo de XML de disco

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>disk</rasd:OtherResourceType>
        <gprop:GenericProperty key="vdisk_name">vdisk0</gprop:GenericProperty>
        <gprop:GenericProperty key="service_name">primary-vds0</gprop:GenericProperty>
        <gprop:GenericProperty key="vol_name">vdsdev0</gprop:GenericProperty>
        <gprop:GenericProperty key="timeout">60</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>

```

## Recurso de conmutador virtual (vsw)

Un recurso `vsw` puede estar en una sección `<content>` como parte de la descripción del dominio, o puede aparecer por sí mismo en una sección `<Envelope>`. Debe tener una etiqueta `<gprop:genericProperty>` con la clave `service_name`, que es el nombre que se asignará al conmutador virtual.

De manera opcional, el recurso `vsw` también puede tener las siguientes propiedades:

- `<rasd:Address>` – Asigna una dirección al conmutador virtual.
- `default-vlan-id` – Especifica la red de área local virtual (VLAN) predeterminada de la que debe ser miembro un dispositivo de red virtual o conmutador virtual, en modo con etiquetas. El primer ID de VLAN (`vid1`) se reserva para `default-vlan-id`.
- `dev_path` – Ruta del dispositivo de red que se debe asociar con este conmutador virtual.
- `id` – Especifica el ID de un nuevo dispositivo de conmutador virtual. De manera predeterminada, estos valores de ID se generan automáticamente, así que debe configurar esta propiedad si necesita que coincida con un nombre de un dispositivo existente en el SO.
- `inter_vnet_link` – Especifica si se asignarán canales LDC para la comunicación entre redes virtuales. Los valores válidos son `on`, `off` y `auto`. El valor predeterminado es `auto`.
- `linkprop`: especifica que el dispositivo virtual obtiene actualizaciones de estado del enlace físico (el valor por defecto de `phys-state`). Cuando el valor está en blanco, el dispositivo virtual no obtiene las actualizaciones de estado del vínculo físico.
- `mode – sc` para la asistencia técnica de respuesta de Cluster de Oracle Solaris.
- `pvid` – Identificador (ID) de la red de área local virtual (VLAN) del puerto, que indica la VLAN de la que debe ser miembro la red virtual, en modo sin etiquetas.
- `mtu` – Especifica la unidad de transmisión máxima (MTU) de un conmutador virtual, los dispositivos de red virtual que están enlazados al conmutador virtual o ambos. Los valores válidos son en el rango de 1500-16000. El comando `ldm` genera un error si se especifica un valor no válido.
- `vid` – Identificador (ID) de la red de área local virtual (VLAN) indica la VLAN de la que una red virtual y un conmutador virtual necesitan ser miembro, en modo con etiquetas.

### EJEMPLO 19 Ejemplo de XML `vsw`

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg2">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vsw</rasd:OtherResourceType>
        <rasd:Address>00:14:4f:fb:ec:00</rasd:Address>
        <gprop:GenericProperty key="service_name">test-vsw1</gprop:GenericProperty>
        <gprop:GenericProperty key="inter_vnet_link">auto</gprop:GenericProperty>
        <gprop:GenericProperty key="default-vlan-id">1</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

```

    <gprop:GenericProperty key="pvid">1</gprop:GenericProperty>
    <gprop:GenericProperty key="mtu">1500</gprop:GenericProperty>
    <gprop:GenericProperty key="dev_path">switch@0</gprop:GenericProperty>
    <gprop:GenericProperty key="id">0</gprop:GenericProperty>
  </Item>
</Section>
</Content>
</Envelope>

```

## Recurso de red (network)

Un recurso de `network` siempre está contenido en una sección `<Content>`. Tiene que tener las etiquetas `<gprop:GenericProperty>` con las siguientes claves:

- `linkprop`: especifica que el dispositivo virtual obtiene actualizaciones de estado del enlace físico (el valor por defecto de `phys-state`). Cuando el valor está en blanco, el dispositivo virtual no obtiene las actualizaciones de estado del vínculo físico.
- `vnet_name` – Nombre de la red virtual (`vnet`).
- `service_name` – Nombre del conmutador virtual (`vswitch`) al que está enlazada esta red virtual.
- `custom`: especifica si activar o desactivar la configuración personalizada para el número máximo de VLAN y direcciones MAC que se pueden asignar a un dispositivo de red virtual desde un host de confianza. El valor por defecto es `disable`.
- `custom/max-mac-addr`s: especifica el número máximo de direcciones MAC que se pueden asignar a un dispositivo de red virtual desde un host de confianza. El valor por defecto es 4096.
- `custom/max-vlans`: especifica el número máximo de VLAN que se pueden asignar a un dispositivo de red virtual desde un host de confianza. El valor por defecto es 4096.

De manera opcional, el recurso `network` también puede tener las siguientes propiedades:

- `<rasd:Address>` – Asigna una dirección al conmutador virtual.
- `pvid` – Identificador (ID) de la red de área local virtual (VLAN) del puerto, que indica la VLAN de la que debe ser miembro la red virtual, en modo sin etiquetas.
- `vid` – Identificador (ID) de la red de área local virtual (VLAN) indica la VLAN de la que una red virtual y un conmutador virtual necesitan ser miembro, en modo con etiquetas.

### EJEMPLO 20 Ejemplo de XML de `network`

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>

```

```

    <rasd:OtherResourceType>network</rasd:OtherResourceType>
    <gprop:GenericProperty key="linkprop">phys-state</gprop:GenericProperty>
    <gprop:GenericProperty key="custom">enable</gprop:GenericProperty>
    <gprop:GenericProperty key="custom/max-mac-addr">4096</gprop:GenericProperty>
    <gprop:GenericProperty key="custom/max-vlans">12</gprop:GenericProperty>
    <gprop:GenericProperty key="vnet_name">ldg1-vnet0</gprop:GenericProperty>
    <gprop:GenericProperty
      key="service_name">primary-vsw0</gprop:GenericProperty>
    <rasd:Address>00:14:4f:fc:00:01</rasd:Address>
  </Item>
</Section>
</Content>
</Envelope>

```

## Recurso del concentrador de consola virtual (vcc)

Un recurso vcc puede estar en una sección <content> como parte de la descripción del dominio, o puede aparecer por sí mismo en una sección <Envelope>. Puede tener etiquetas <gprop:GenericProperty> con las siguientes claves:

- service\_name – Nombre que se debe asignar al servicio de concentrador de consola virtual
- min\_port – Número de puerto mínimo que se debe asignar con este vcc
- max\_port – Número de puerto máximo que se debe asociar con este vcc

### EJEMPLO 21 Ejemplo de XML vcc

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vcc</rasd:OtherResourceType>
        <gprop:GenericProperty key="service_name">vcc1</gprop:GenericProperty>
        <gprop:GenericProperty key="min_port">6000</gprop:GenericProperty>
        <gprop:GenericProperty key="max_port">6100</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>

```

## Recurso de variable (var)

Un recurso var siempre está contenido en una sección <Content>. Puede tener etiquetas <gprop:GenericProperty> con las siguientes claves:

- name – Nombre de la variable
- value – Valor de la variable

**EJEMPLO 22** Ejemplo de XML de var

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>var</rasd:OtherResourceType>
        <gprop:GenericProperty key="name">test_var</gprop:GenericProperty>
        <gprop:GenericProperty key="value">test1</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>

```

## Recurso de dispositivo de E/S físico (physio\_device)

Un recurso `physio_device` siempre está contenido en una sección `<Content>`. Este recurso se puede modificar mediante los subcomandos `add-io`, `set-io`, `remove-io`, `create-vf`, `destroy-vf` y `set-domain`.

**EJEMPLO 23** Ejemplo de XML de `physio_device`

Los siguientes ejemplos muestran cómo realizar acciones en las funciones virtuales, las funciones físicas y los complejos raíz.

- El siguiente fragmento de ejemplo de XML muestra cómo utilizar el comando `ldm add-io` para agregar la función virtual `/SYS/MB/NET0/IOVNET.PF0.VF0` al dominio `ldg1`.

```

<LDM_interface version="1.3">
  <cmd>
    <action>add-io</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="ldg1">
          <Section xsi:type="ovf:VirtualHardwareSection_Type">
            <Item>
              <rasd:OtherResourceType>physio_device</rasd:OtherResourceType>
              <gprop:GenericProperty key="name">
                /SYS/MB/NET0/IOVNET.PF0.VF0</gprop:GenericProperty>
            </Item>
          </Section>
        </Content>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>

```

```

    </data>
  </cmd>
</LDM_interface>

```

- El siguiente fragmento de ejemplo de XML muestra cómo utilizar el comando `ldm set-io` para establecer el valor de propiedad `iov_bus_enable_iov` en `on` para el complejo raíz `pci_1`.

```

<LDM_interface version="1.3">
  <cmd>
    <action>set-io</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Section xsi:type="ovf:VirtualHardwareSection_Type">
          <Item>
            <rasd:OtherResourceType>physio_device</rasd:OtherResourceType>
            <gprop:GenericProperty key="name">pci_1</gprop:GenericProperty>
            <gprop:GenericProperty key="iov_bus_enable_iov">
              on</gprop:GenericProperty>
          </Item>
        </Section>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>

```

- El siguiente fragmento de ejemplo de XML muestra cómo utilizar el comando `ldm set-io` para establecer el valor de propiedad `unicast-slots` en `6` para la función física `/SYS/MB/NET0/IOVNET.PF1`.

```

<LDM_interface version="1.3">
  <cmd>
    <action>set-io</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Section xsi:type="ovf:VirtualHardwareSection_Type">
          <Item>
            <rasd:OtherResourceType>physio_device</rasd:OtherResourceType>
            <gprop:GenericProperty key="name">
              /SYS/MB/NET0/IOVNET.PF1</gprop:GenericProperty>
            <gprop:GenericProperty key="unicast-slots">6</gprop:GenericProperty>
          </Item>
        </Section>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>

```

```

    </cmd>
  </LDM_interface>

```

- El siguiente fragmento de ejemplo de XML muestra cómo utilizar el comando `ldm create-vf` para crear la función virtual `/SYS/MB/NET0/IOVNET.PF1.VF0` con los siguientes valores de propiedad.

- `unicast-slots=6`
- `pvid=3`
- `mtu=1600`

```

<LDM_interface version="1.3">
  <cmd>
    <action>create-vf</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Section xsi:type="ovf:VirtualHardwareSection_Type">
          <Item>
            <rasd:OtherResourceType>vf_device</rasd:OtherResourceType>
            <gprop:GenericProperty key="iov_pf_name">
              /SYS/MB/NET0/IOVNET.PF1</gprop:GenericProperty>
            <gprop:GenericProperty key="unicast-slots">6</gprop:GenericProperty>
            <gprop:GenericProperty key="pvid">3</gprop:GenericProperty>
            <gprop:GenericProperty key="mtu">1600</gprop:GenericProperty>
          </Item>
        </Section>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>

```

- El siguiente fragmento de ejemplo XML muestra cómo usar el comando `ldm create-vf` para crear el número de funciones virtuales especificado por el valor `iov_pf_repeat_count_str` (3) con la función física `/SYS/MB/NET0/IOVNET.PF1`. No puede especificar otros valores de propiedad cuando se crean varias funciones virtuales con la propiedad `iov_pf_repeat_count_str`.

```

<LDM_interface version="1.3">
  <cmd>
    <action>create-vf</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Section xsi:type="ovf:VirtualHardwareSection_Type">
          <Item>

```

```

        <rasd:OtherResourceType>vf_device</rasd:OtherResourceType>
        <gprop:GenericProperty key="iov_pf_name">
          /SYS/MB/NET0/IOVNET.PF1</gprop:GenericProperty>
        <gprop:GenericProperty key="iov_pf_repeat_count_str">
          3</gprop:GenericProperty>
      </Item>
    </Section>
  </Envelope>
</data>
</cmd>
</LDM_interface>

```

## Recurso de configuración SP (spconfig)

Un recurso de configuración del procesador de servicio (SP) (spconfig) siempre aparece por sí mismo en una sección <Envelope>. Puede tener etiquetas <gprop:GenericProperty> con las siguientes claves:

- spconfig\_name – Nombre de la configuración que debe guardarse en el SP
- spconfig\_status – El estado actual de una determinada configuración SP. La propiedad se usa en la salida de un comando `ldm list-spconfig`.

### EJEMPLO 24 Ejemplo de XML de spconfig

```

<Envelope>
  <Section xsi:type="ovf:ResourceAllocationSection_type">
    <Item>
      <rasd:OtherResourceType>spconfig</rasd:OtherResourceType>
      <gprop:GenericProperty
        key="spconfig_name">primary</gprop:GenericProperty>
      <gprop:GenericProperty
        key="spconfig_status">current</gprop:GenericProperty>
      <gprop:GenericProperty
        key="spconfig_aux_status">degraded</gprop:GenericProperty>
    </Item>
  </Section>
</Envelope>

```

## Recurso de configuración de directiva de DRM (policy)

Aparece un recurso de directiva de DRM (policy) en la sección <Envelope> y puede tener etiquetas <gprop:GenericProperty> con las siguientes claves:

- policy\_name: nombre de la directiva de DRM
- policy\_enable: especifica si la directiva DRM está habilitada o inhabilitada
- policy\_priority: prioridad de la directiva de DRM
- policy\_vcpu\_min: número mínimo de recursos de CPU virtuales para un dominio
- policy\_vcpu\_max: número máximo de recursos de CPU virtuales para un dominio
- policy\_util\_lower: nivel de uso inferior en el que se activa el análisis de directiva
- policy\_util\_upper: nivel de uso superior en el que se activa el análisis de directiva
- policy\_tod\_begin: hora de inicio efectiva de la directiva de DRM
- policy\_tod\_end: hora de detención efectiva de la política de DRM
- policy\_sample\_rate: la frecuencia de muestreo, que es el tiempo de ciclo en segundos
- policy\_elastic\_margin: cantidad de búfer entre los límites de uso de la CPU inferior y superior
- policy\_attack: cantidad máxima de un recurso que se añadirá durante cualquier ciclo de control de recursos
- policy\_decay: cantidad máxima de un recurso que se eliminará durante cualquier ciclo de control de recursos

### EJEMPLO 25 Ejemplo de XML policy

```
<Envelope>
  <Section xsi:type="ovf:VirtualHardwareSection_Type">
    <Item>
      <rasd:OtherResourceType>policy</rasd:OtherResourceType>
      <gprop:GenericProperty key="policy_name">test-policy</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_enable">on</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_priority">1</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_vcpu_min">12</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_vcpu_max">13</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_util_lower">8</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_util_upper">9</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_tod_begin">07:08:09</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_tod_end">09:08:07</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_sample_rate">1</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_elastic_margin">8</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_attack">8</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_decay">9</gprop:GenericProperty>
    </Item>
  </Section>
</Envelope>
```

## Recurso de consola (console)

Un recurso `console` siempre está contenido en una sección `<Content`. Puede tener etiquetas `<gprop:GenericProperty>` con las siguientes claves:

- `port` – Puerto al que se debe cambiar esta consola virtual (`console`)
- `service_name` – Servicio de concentrador de consola virtual (`vcc`) al que se enlaza esta consola
- `group` – Nombre del grupo al que enlazar esta consola
- `enable-log` – Activar o desactivar el registro de la consola virtual para esta consola

### EJEMPLO 26 Ejemplo de XML de `console`

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>console</rasd:OtherResourceType>
        <gprop:GenericProperty key="port">6000</gprop:GenericProperty>
        <gprop:GenericProperty key="service_name">vcc2</gprop:GenericProperty>
        <gprop:GenericProperty key="group">group-name</gprop:GenericProperty>
        <gprop:GenericProperty key="enable-log">on</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

## Migración de dominio

Este ejemplo muestra lo que contiene la sección `<data` para un comando `ldm migrate-domain`.

- Primero, el nodo `<Content>` (sin una sección `<ldom_info>`) es el dominio de origen para migrar.
- Segundo, el nodo `<Content>` (con una sección `<ldom_info>`) es el dominio de destino al que migrar. Los nombres del dominio de origen y destino pueden ser los mismos.
- La sección `<ldom_info>` para el dominio de destino describe el equipo al que migrar y los detalles necesarios para migrar a dicho equipo:
  - `target-host` es el equipo de destino al que migrar.
  - `user-name` es el nombre de usuario de inicio de sesión para el equipo de destino, que debe estar codificado para SASL de 64 bits.
  - `password` es la contraseña que se debe usar para iniciar sesión en el equipo de destino, que debe estar codificado para SASL de 64 bits.

---

**Nota** - El Dominios lógicos Manager usa `sas1_decode64()` para decodificar el nombre de usuario de destino y la contraseña y usa `sas1_encode64()` para codificar estos valores. La codificación SASL 64 es equivalente a la codificación base64.

---

**EJEMPLO 27** Ejemplo migrate-domain Sección <data>

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="ldg1"/>
  <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="ldg1"/>
  <Section xsi:type="ovf:ResourceAllocationSection_Type">
    <Item>
      <rasd:OtherResourceType>ldom_info</rasd:OtherResourceType>
      <gprop:GenericProperty key="target">target-host</gprop:GenericProperty>
      <gprop:GenericProperty key="username">user-name</gprop:GenericProperty>
      <gprop:GenericProperty key="password">password</gprop:GenericProperty>
    </Item>
  </Section>
</Content>
</Envelope>
```

## Esquemas XML

Los esquemas XML que utiliza Dominios lógicos Manager se encuentran en el directorio `/opt/SUNWldm/bin/schemas`. Los nombres de archivo son los siguientes:

- `cim-common.xsd`: esquema `cim-common.xsd`
- `cim-rasd.xsd`: esquema `cim-rasd.xsd`
- `cim-vssd.xsd`: esquema `cim-vssd.xsd`
- `cli-list-constraint-v3.xsd`: esquema `cli-list-constraint-v3.xsd`
- `combined-v3.xsd`: esquema XML `LDM_interface`
- `event-v3.xsd`: esquema XML `LDM_Event`
- `ldmd-binding.xsd`: esquema XML `Binding_Type`
- `ldmd-property.xsd`: esquema XML `GenericProperty`
- `ovf-core.xsd`:esquema `ovf-core.xsd`
- `ovf-envelope.xsd`: esquema `ovf-envelope.xsd`
- `ovf-section.xsd`:esquema `ovf-section.xsd`
- `ovf-strings.xsd`:esquema `ovf-strings.xsd`
- `ovfenv-core.xsd`:esquema `ovfenv-core.xsd`
- `ovfenv-section.xsd`: esquema `ovfenv-section.xsd`



# ◆◆◆ 4 C A P Í T U L O 4

## Descubrimiento de Dominios lógicos Manager

---

En este capítulo se proporciona información sobre el descubrimiento de Dominios lógicos Manager en ejecución en sistemas de una subred.

### Descubrimiento de sistemas que ejecutan el Dominios lógicos Manager

Dominios lógicos Managers pueden ser descubiertos en una subred mediante mensajes multidifusión. El daemon `ldmd` puede escuchar en una red para un paquete multidifusión específico. Si el mensaje multidifusión es de un determinado tipo, `ldmd` responde al llamador. Esto permite que `ldmd` sea descubierto en sistemas que ejecutan el Oracle VM Server for SPARC.

### Comunicación multidifusión

El mecanismo de descubrimiento usa la misma red multidifusión usada por el daemon `ldmd` para detectar colisiones cuando se asignan direcciones MAC automáticamente. Para configurar un punto de conexión multidifusión, debe suministrar la siguiente información:

```
#define MAC_MULTI_PORT 64535
#define MAC_MULTI_GROUP "239.129.9.27"
```

De manera predeterminada, *solo* pueden enviarse paquetes multidifusión en la subred a la que está asociado el equipo. Puede cambiar el comportamiento configurando la propiedad SMF `ldmd/hops` para el daemon `ldmd`.

### Formato del mensaje

Los mensajes de descubrimiento deben marcarse claramente de manera que no se confundan con otros mensajes. El siguiente formato de los mensajes multidifusión asegura que

los mensajes de descubrimiento pueden ser distinguidos por el proceso de escucha de descubrimiento:

```
#include <netdb.h> /* Used for MAXHOSTNAMELEN definition */
#define MAC_MULTI_MAGIC_NO 92792004
#define MAC_MULTI_VERSION 1

enum {
    SEND_MSG = 0,
    RESPONSE_MSG,
    LDMD_DISC_SEND,
    LDMD_DISC_RESP,
};

typedef struct {
    uint32_t version_no;
    uint32_t magic_no;
    uint32_t msg_type;
    uint32_t resv;
    union {
        mac_lookup_t Mac_lookup;
        ldmd_discovery_t Ldmd_discovery;
    } payload;
#define lookup payload.Mac_lookup
#define discovery payload.Ldmd_discovery
} multicast_msg_t;

#define LDMD_VERSION_LEN 32

typedef struct {
    uint64_t mac_addr;
    char source_ip[INET_ADDRSTRLEN];
} mac_lookup_t;

typedef struct {
    char ldmd_version[LDMD_VERSION_LEN];
    char hostname[MAXHOSTNAMELEN];
    struct in_addr ip_address;
    int port_no;
} ldmd_discovery_t;
```

## ▼ Cómo detectar la ejecución de Dominios lógicos Managers en la subred

### 1. Abra un punto de conexión multidifusión.

Asegúrese de que usa el puerto y la información de grupo especificada en [“Comunicación multidifusión” \[53\]](#).

### 2. Envíe un mensaje `multicast_msg_t` por el punto de conexión.

El mensaje debe incluir los siguientes datos:

- Valor válido para `version_no`, que es 1 tal y como definido por `MAC_MULTI_VERSION`

- Valor válido para `magic_no`, que es 92792004 tal y como definido por `MAC_MULTI_MAGIC_NO`
- `msg_type` de `LDMD_DISC_SEND`

**3. Escuche en el socket de multidifusión para detectar respuestas de Dominios lógicos Managers.**

Las respuestas deben ser un mensaje `multicast_msg_t` con la siguiente información:

- Valor válido para `version_no`
- Valor válido para `magic_no`
- `msg_type` establecido a `LDMD_DISC_RESP`
- La carga debe consistir en una estructura `ldmd_discovery_t`, que contenga la siguiente información:
  - `ldmd_version` – Versión del Dominios lógicos Manager que se ejecuta en el sistema
  - `hostname` – Nombre del host del sistema
  - `ip_address` – Dirección IP del sistema
  - `port_no` – Número de puerto usado por Dominios lógicos Manager para las comunicaciones, que debe ser el puerto XMPP 6482

Cuando escucha para recibir una respuesta de Dominios lógicos Managers, asegúrese de que no se tienen en cuenta los paquetes de detección de colisión de MAC de asignación automática.



# ◆◆◆ 5 CAPÍTULO 5

## Uso de la API y del comando de información de dominio virtual

---

En este capítulo, se describe la API y el comando de información de dominio virtual.

### Uso del comando de información de dominio virtual

El comando `virtinfo` le habilita para obtener información sobre un dominio virtual en ejecución.

La siguiente lista muestra parte de la información que puede obtener sobre un dominio virtual utilizando un comando o API:

- Tipo de dominio (implementación, control, invitado, E/S, servicio, raíz)
- Nombre de dominio determinado por el administrador de dominio virtual
- Identificador único universal (UUID) del dominio
- Nombre del nodo de red del dominio de control del dominio
- Número serial de chasis en el que se está ejecutando el dominio

Para obtener más información sobre el comando `virtinfo`, consulte la página del comando `man virtinfo(1M)`.

### Uso de la API de información de dominio virtual

También puede usar la API de información de dominio virtual para crear programas que obtienen información sobre los dominios virtuales. Consulte las páginas del comando `man libv12n(3LIB)` y `v12n(3EXT)`.



# Índice

---

## D

Dominios lógicos Manager  
  esquema XML usado con, 21  
  mecanismo de descubrimiento, 53

## E

esquemas XML, 51  
  Dominios lógicos Manager usado con, 21  
etiquetas XML  
  <cmd>, 24  
  <data>, 25  
  <LDM\_interface>, 24  
eventos XML  
  dominio, 29  
  hardware, 30  
  mensajes, 27  
  progreso, 30  
  recurso, 31  
  registro y anulación de registro, 28  
  tipos, 29  
  todos, 32

## I

información de dominio virtual  
  API, 57  
  virtinfo, 57  
instalación  
  utilidades de la Plantilla de Oracle VM Server for  
  SPARC, 10

## P

protocolo XML, 22

## R

recursos XML  
  console, 50  
  cpu, 36  
  disk, 41  
  ldom\_info, 35  
  mau, 38  
  memory, 39  
  network, 43  
  physio\_device, 45  
  policy, 49  
  spconfig, 48  
  var, 44  
  vcc, 44  
  vds, 40  
  vds\_volume, 40  
  vsan, 39  
  vsw, 42

## T

tramas de transporte de XML, 21

## U

utilidades de la Plantilla de Oracle VM Server for  
SPARC  
  instalación, 10

## V

### virtinfo

- información de dominio virtual, 57

## X

### XML

- <LDM\_event> mensajes, 28
- acciones, Dominios lógicos Manager, 32
- esquemas, 51
- mensajes de respuesta, 25
- mensajes de solicitud, 23
- mensajes de solicitud y respuesta, 23
- migración de dominio, 50
- recursos y propiedades del Dominios lógicos Manager, 34
- respuesta de comando, 27
- respuesta de objeto, 27
- respuesta general, 26

### XMPP

- conexiones locales, 22
- servidor, 22