



Guía de supervisión de Sun Java Enterprise System 5



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Referencia: 820-1588-10
Marzo de 2007

Sun Microsystems, Inc. posee derechos de propiedad intelectual en relación con la tecnología incluida en el producto descrito en este documento. De forma específica y, sin limitación, entre estos derechos de propiedad intelectual se incluyen una o varias patentes en los EE.UU. o aplicaciones pendientes de patente en los EE.UU. y otros países.

Derechos del gobierno de Estados Unidos: Software comercial. Los usuarios gubernamentales están sujetos al acuerdo de licencia estándar de Sun Microsystems, Inc. y a las disposiciones aplicables de la regulación FAR y sus suplementos.

Esta distribución puede incluir materiales desarrollados por terceras partes.

Algunas partes del producto pueden proceder de los sistemas Berkeley BSD, con licencia de la Universidad de California. UNIX es una marca comercial registrada en EE.UU. y en otros países, cuya licencia se otorga exclusivamente a través de X/Open Company, Ltd.

Sun, Sun Microsystems, el logotipo de Sun, el logotipo de Solaris, el logotipo de la taza de café de Java, docs.sun.com, JavaScript, Java, JavaServer Pages, JSP y Solaris son marcas comerciales o marcas comerciales registradas de Sun Microsystems, Inc. en EE.UU. y en otros países. Todas las marcas comerciales de SPARC se utilizan bajo licencia y son marcas comerciales o marcas registradas de SPARC International, Inc. en EE.UU. y en otros países. Los productos con marcas registradas de SPARC se basan en una arquitectura desarrollada por Sun Microsystems, Inc.

La interfaz gráfica de usuario de OPEN LOOK y SunTM fue desarrollada por Sun Microsystems, Inc. para sus usuarios y licenciatarios. Sun reconoce los esfuerzos pioneros de Xerox en la investigación y desarrollo del concepto de las interfaces gráficas o visuales de usuario para el sector de la informática. Sun cuenta con una licencia no exclusiva de Xerox para la interfaz gráfica de usuario de Xerox, que cubre también a los licenciatarios de Sun que implementan GUIs de OPEN LOOK y que cumplen con los acuerdos de licencia por escrito de Sun.

Los productos descritos y abordados en esta publicación están sometidos a la legislación de control de exportaciones de los EE.UU. y pueden estar sujetos a leyes de importación o exportación de otros países. Queda terminantemente prohibido el uso final (directo o indirecto) de esta documentación para el desarrollo de armas nucleares, químicas, biológicas, de uso marítimo nuclear o misiles. Se prohíbe estrictamente la exportación o reexportación a países bajo el embargo de los EE.UU. o a entidades incluidas en la lista de exclusión de exportación de los EE.UU., incluidas, pero no limitándose a, las personas rechazadas y a las listas nacionales designadas específicamente.

ESTA DOCUMENTACIÓN SE PROPORCIONA "TAL COMO ESTÁ" Y NO SE ASUMIRÁ LA RESPONSABILIDAD DE NINGUNA CONDICIÓN EXPRESA O IMPLÍCITA, NI DE REPRESENTACIONES NI GARANTÍAS, INCLUIDA CUALQUIER GARANTÍA IMPLÍCITA DE APROVECHAMIENTO, IDONEIDAD PARA UN FIN PARTICULAR O NO INFRACCIÓN, EXCEPTO EN EL CASO EN QUE TALES RENUNCIAS DE RESPONSABILIDAD NO SE CONSIDEREN LEGALMENTE VÁLIDAS.

Contenido

- Prefacio7**

- 1 Descripción general de la supervisión de Java ES 15**
 - Componentes de Monitoring Framework y Consola de supervisión 15
 - Funcionamiento de la supervisión en Java ES 16
 - Common Monitoring Model (CMM) 17
 - Instrumentación de CMM 17
 - Agentes de nodo 18
 - Agente principal 19
 - Secuencia de instalación recomendada 20

- 2 Habilitación y configuración de Monitoring Framework 21**
 - Diseño de los directorios instalados 22
 - Utilización de Monitoring Framework con Access Manager 23
 - ▼ Para habilitar la supervisión en Access Manager 23
 - Utilización de Monitoring Framework con Application Server 24
 - ▼ Para habilitar la supervisión en Application Server 24
 - Utilización de Monitoring Framework con Calendar Server 25
 - ▼ Para habilitar la supervisión en Calendar Server 25
 - Utilización de Monitoring Framework con Directory Server 25
 - ▼ Para habilitar la supervisión en Directory Server 26
 - Utilización de Monitoring Framework con Instant Messaging 26
 - ▼ Para habilitar la supervisión con Instant Messaging 26
 - Utilización de Monitoring Framework con Messaging Server 27
 - ▼ Para habilitar la supervisión en Messaging Server 27
 - Utilización de Monitoring Framework con Portal Server 27
 - ▼ Para habilitar la supervisión en Portal Server 27

Utilización de Monitoring Framework con Web Server	28
▼ Para habilitar la supervisión en Web Server	28
Configuración de Common Agent Container	28
▼ Para habilitar la supervisión del Common Agent Container	29
Solución de problemas de Monitoring Framework	29
Utilización de Monitoring Framework en plataformas HP-UX	29
Utilización de Monitoring Framework en Microsoft Windows	30
▼ Para reiniciar un agente de nodo	30
Comando mfwkadm	30
Resumen	31
Descripción	32
Opciones	33
Subcomandos	33
Ejemplos	44
Estado de salida	50
Atributos	50
Consulte también	50
3 Instalación y utilización de Consola de supervisión	51
Instalación de Consola de supervisión	51
▼ Para instalar la Consola de supervisión con el instalador de Java ES	52
▼ Para instalar la Consola de supervisión en una zona Solaris	53
▼ Para configurar la Consola de supervisión	54
▼ Para desconfigurar la Consola de supervisión	54
Diseño del directorio instalado	54
Inicio de la Consola de supervisión	55
▼ Para iniciar la Consola de supervisión	55
▼ Para conectarse a sus agentes de nodo	56
Utilización de la Consola de supervisión	59
▼ Para inhabilitar y rehabilitar selectivamente la supervisión	59
▼ Para crear una regla de supervisión	60
Solución de problemas de la Consola de supervisión	68
A Referencia de objetos CMM	71
Descripción general de los objetos CMM	71

B Objetos supervisados expuestos por cada componente 73

Instrumentación de Common Agent Container 73

Instrumentación de Access Manager 73

Instrumentación de Application Server 73

Instrumentación de Calendar Server 73

Instrumentación de Directory Server 74

Instrumentación de Instant Messaging 74

Instrumentación de Messaging Server 74

Instrumentación de Portal Server 74

Instrumentación de Web Server 74

Índice75

Prefacio

En esta guía se describe la nueva función de supervisión existente en Sun Java™ Enterprise System 5 (Java ES). La supervisión se implementa mediante Sun Java System Monitoring Framework 2.0 y Sun Java System Consola de supervisión 1.0.

Los procedimientos descritos en esta guía muestran cómo configurar y habilitar Monitoring Framework para cada uno de los componentes instalados y, a continuación, cómo visualizar todos los datos supervisados en Consola de supervisión. En esta guía no se documentan archivos de registro ni otros mecanismos de supervisión de componentes individuales fuera de esta estructura.

Usuarios a los que está destinada esta guía

Esta guía está destinada a los siguientes usuarios:

- Arquitectos de software que deban diseñar un plan de mantenimiento para implementaciones de Java ES.
- Administradores de sistema que realicen la instalación y configuración de Java ES.
- Administradores de sistema y técnicos que supervisen y mantengan implementaciones de Java ES.

Antes de leer esta guía

Debe familiarizarse con los documentos del conjunto de documentación de Java ES descrito en la siguiente sección. Debe familiarizarse también con el diseño y el funcionamiento de los componentes de Java ES que desee supervisar.

Además, si va a instalar y configurar los componentes de supervisión, deberá finalizar primero la instalación de los demás componentes. Antes de realizar cualquier instalación o configuración, consulte el documento *Notas de la versión Sun Java Enterprise System 5 para UNIX*.

Conjunto de documentación de Java ES

El conjunto de documentación de Java ES describe el procedimiento de planificación de la instalación y el de instalación del sistema. La dirección URL para la documentación del sistema es <http://docs.sun.com/coll/1286.2>. Para ver una introducción a Java ES, consulte los manuales en el orden en que figuran en la siguiente tabla.

TABLA P-1 Documentación de Java Enterprise System

Título del documento	Contenido
<i>Notas de la versión Sun Java Enterprise System 5 para UNIX</i>	Contiene la información más actualizada acerca de Java ES, incluidos los problemas conocidos. Además, los componentes tienen sus propias notas de la versión enumeradas en la recopilación de notas de versión (http://docs.sun.com/coll/1315.2).
<i>Sun Java Enterprise System 5 Release Notes for Microsoft Windows</i>	
<i>Descripción general técnica de Sun Java Enterprise System 5</i>	Presenta los fundamentos técnicos y conceptuales de Java ES. Describe los componentes, la arquitectura, los procesos y las funciones.
<i>Sun Java Enterprise System Deployment Planning Guide</i>	Proporciona una introducción a la planificación y el diseño de soluciones de implementación empresarial basadas en Java ES. Presenta conceptos básicos y principios relacionados con el diseño y la planificación de implementaciones, muestra el ciclo de vida de la solución y proporciona ejemplos generales y estrategias que se pueden usar a la hora de planificar soluciones basadas en Java ES.
<i>Guía de planificación de la instalación de Sun Java Enterprise System 5</i>	Le ayuda a desarrollar las especificaciones de implementación para el hardware, el sistema operativo y la red de su implementación de Java ES. Describe problemas a los que deberá hacer frente al planificar la configuración y la instalación como, por ejemplo, las dependencias existentes entre los componentes.
<i>Guía de instalación de Sun Java Enterprise System 5 para UNIX</i>	Guía al usuario en el proceso de instalación de Java ES. También muestra cómo se configuran los componentes después de instalarlos y cómo se comprueba si funcionan correctamente.
<i>Sun Java Enterprise System 5 Installation Guide for Microsoft Windows</i>	
<i>Guía de referencia de instalación de Sun Java Enterprise System para UNIX</i>	Ofrece información adicional sobre parámetros de configuración, proporciona hojas de cálculo para utilizarlas en la planificación de la configuración e indica material de referencia tal como números de puerto y directorios predeterminados en el sistema operativo Solaris y en el entorno operativo Linux.
<i>Guía de actualización de Sun Java Enterprise System 5 para UNIX</i>	Proporciona instrucciones para realizar la actualización a Java ES 5 desde versiones anteriormente instaladas.
<i>Sun Java Enterprise System 5 Upgrade Guide for Microsoft Windows</i>	

TABLA P-1 Documentación de Java Enterprise System (Continuación)

Título del documento	Contenido
<i>Guía de supervisión de Sun Java Enterprise System 5</i>	Proporciona instrucciones de configuración de Monitoring Framework para cada componente del producto y de utilización de la Consola de supervisión para visualizar datos en tiempo real y crear reglas de supervisión.
<i>Sun Java Enterprise System Glossary</i>	Define los términos que se utilizan en la documentación de Java ES.

Rutas predeterminadas y nombres de archivo

En la siguiente tabla se describen las rutas predeterminadas y los nombres de archivo de los componentes de Java ES que implementan la supervisión.

TABLA P-2 Rutas predeterminadas y nombres de archivo

Convenciones de	Descripción	Valor predeterminado
<i>base-de-mfwk</i>	Representa el directorio en el que el componente compartido de Monitoring Framework se instala automáticamente. Esta ruta se utiliza también como parte del directorio de configuración.	Sistemas Solaris: /opt/SUNWmfwk Sistemas Linux: /opt/sun/mfwk
<i>base-de-MConsole</i>	Representa el directorio de instalación elegido para Consola de supervisión.	Sistemas Solaris: /opt/SUNWjesmc Sistemas Linux: /opt/sun/jesmc
<i>base-de-WebConsole</i>	Representa el directorio en el que el componente compartido de Web Console se instala automáticamente.	Sistemas Solaris: /etc/webconsole/console Sistemas Linux: /etc/opt/webconsole/console
<i>base-de-AccessMgr</i>	Representa el directorio de instalación elegido para Sun Java System Access Manager.	Sistemas Solaris: /opt/SUNWam Sistemas Linux: /opt/sun/identity
<i>base-de-AppServer</i>	Representa el directorio de instalación elegido para Sun Java System Application Server.	Sistemas Solaris: /opt/SUNWappserver/appserver Sistemas Linux: /opt/sun/appserver
<i>base-de-CalServ</i>	Representa el directorio de instalación elegido para Sun Java System Calendar Server.	Sistemas Solaris: /opt/SUNWics5 Sistemas Linux: /opt/sun/calendar
<i>base-de-DirServ</i>	Representa el directorio de instalación elegido para Sun Java System Directory Server.	Sistemas Solaris: /opt/SUNWdsee/ds6 Sistemas Linux: /opt/sun/ds6

TABLA P-2 Rutas predeterminadas y nombres de archivo (Continuación)		
Convenciones de	Descripción	Valor predeterminado
<i>base-de-IM</i>	Representa el directorio de instalación elegido para Sun Java System Instant Messaging.	Sistemas Solaris: /opt/SUNWim Sistemas Linux: /opt/sun/im
<i>base-de-MsgServ</i>	Representa el directorio de instalación elegido para Sun Java System Messaging Server .	Sistemas Solaris: /opt/SUNWmsgsr Sistemas Linux: /opt/sun/messaging
<i>base-de-Portal</i>	Representa el directorio de instalación elegido para Sun Java System Portal Server.	Sistemas Solaris: /opt/SUNWportal Sistemas Linux: /opt/sun/portal
<i>base-de-WebServer</i>	Representa el directorio de instalación elegido para Sun Java SystemWeb Server.	Sistemas Solaris: /opt/SUNWwbsvr7 Sistemas Linux: /opt/sun/webserver

Convenciones tipográficas

La tabla siguiente describe los cambios tipográficos que se utilizan en esta guía.

TABLA P-3 Convenciones tipográficas

Tipos de letra	Significado	Ejemplo
AaBbCc123	Los nombres de comandos, archivos y directorios, y salida del equipo en pantalla	Edite el archivo <code>.login</code> . Utilice el comando <code>ls -a</code> para enumerar todos los archivos. <code>machine_name% you have mail.</code>
AaBbCc123	Lo que escribe el usuario, en contraste con lo que el equipo muestra en pantalla.	<code>machine_name% su</code> <code>Password:</code>
<i>AaBbCc123</i>	Un marcador de posición que se debe sustituir por un valor o un nombre real.	El comando para eliminar un archivo es <code>rm nombre-de-archivo</code> .
<i>AaBbCc123</i>	Títulos de manuales, términos nuevos, y términos que se deben destacar (tenga en cuenta que algunos elementos destacados aparecerán en negrita en línea)	Consulte el capítulo 6 de la <i>Guía del usuario</i> . Un elemento <i>almacenado en caché</i> es una copia que se almacena localmente. <i>No</i> guarde el archivo.

Ejemplos de comandos en el indicador del intérprete de comandos

En la siguiente tabla se muestran los indicadores predeterminados del sistema y los indicadores del superusuario.

TABLA P-4 Indicadores de intérprete de comandos

Intérprete de comandos	Indicador
Intérprete de comandos C en sistemas UNIX y Linux	machine_name%
Superusuario del intérprete de comandos C en sistemas UNIX y Linux	machine_name#
Intérpretes de comandos Bourne y Korn en sistemas UNIX y Linux	\$
Superusuario de los intérpretes de comandos Bourne y Korn en sistemas UNIX y Linux	#
Línea de comandos de Microsoft Windows	C:\

Convenciones de símbolos

En la siguiente tabla se explican los símbolos que pueden encontrarse en este manual.

TABLA P-5 Convenciones de símbolos

Símbolo	Descripción	Ejemplo	Significado
[]	Contiene argumentos optativos y opciones de comandos.	ls [-l]	La opción -l no es obligatoria.
{ }	Incluye un conjunto de selecciones posibles para una opción de comando obligatoria.	-d {y n}	La opción -d requiere que se utilice el argumento y o el argumento n.
\${ }	Indica una referencia variable.	\${com.sun.javaRoot}	Hace referencia al valor de la variable com.sun.javaRoot.
-	Combina varias pulsaciones de teclas simultáneas.	Control-A	Pulse al mismo tiempo la tecla Control y la tecla A.
+	J	Ctrl+A+N	Pulse la tecla Control, suéltela y, a continuación, pulse las teclas siguientes.

TABLA P-5 Convenciones de símbolos (Continuación)

Símbolo	Descripción	Ejemplo	Significado
→	Indica las selecciones de elementos de menú en una interfaz gráfica de usuario.	Archivo → Nuevo → Plantillas	En el menú Archivo, seleccione Nuevo. En el submenú Nuevo, elija Plantillas.

Documentación, soporte y formación

En el sitio web de Sun se proporciona información sobre los siguientes recursos adicionales:

- Documentación (<http://www.sun.com/documentation/>)
- Soporte (<http://www.sun.com/support/>)
- Formación (<http://www.sun.com/training/>)

Búsqueda de documentación de productos de Sun

Además de buscar documentación de productos de Sun en el sitio web docs.sun.comSM, puede utilizar un motor de búsqueda introduciendo la siguiente sintaxis en el campo de búsqueda:

término-de-búsqueda site:docs.sun.com

Por ejemplo, para buscar broker, introduzca lo siguiente:

broker site:docs.sun.com

Para incluir otros sitios web de Sun en la búsqueda (por ejemplo, java.sun.com, www.sun.com y developers.sun.com), utilice `sun.com` en lugar de `docs.sun.com` en el campo de búsqueda.

Referencias a sitios web de terceros relacionados

Se hace referencia a las direcciones URL de terceras partes para proporcionar información adicional relacionada.

Nota – Sun no se hace responsable de la disponibilidad de los sitios Web de terceras partes que se mencionan en este documento. Sun no garantiza ni se hace responsable de ningún contenido, anuncio, producto ni de cualquier otro material disponible en dichos sitios o recursos. Sun no se responsabiliza de ningún daño, real o supuesto, ni de posibles pérdidas que se pudieran derivar del uso de los contenidos, bienes o servicios que estén disponibles en dichos sitios o recursos.

Sun valora sus comentarios

Sun tiene interés en mejorar su documentación y valora sus comentarios y sugerencias. Si desea compartir algún comentario, vaya a <http://docs.sun.com> y haga clic en Enviar comentarios (Send Comments). Se mostrará un formulario en línea en el que deberá indicar el título completo del documento y el número de referencia. El número de referencia consta de 7 o 9 dígitos y se puede encontrar en la página de título del manual o en la dirección URL del documento. Por ejemplo, el número de referencia de este manual es 819-5081 .

Descripción general de la supervisión de Java ES

En esta guía se describen los componentes Monitoring Framework 2.0 y Consola de supervisión 1.0 de Sun Java™ Enterprise System (Java ES). Juntos, estos componentes implementan la nueva función de supervisión introducida en la versión 5.

Los procedimientos descritos en esta guía muestran cómo habilitar Monitoring Framework para cada uno de los componentes instalados y, a continuación, cómo visualizar todos los datos supervisados en Consola de supervisión. En esta guía no se documentan archivos de registro, mensajes de error ni otros mecanismos de supervisión que componentes individuales pueden implementar fuera de la estructura. Ni Monitoring Framework ni Consola de supervisión proporcionan capacidades de administración o de gestión para componentes supervisados. Para obtener información sobre la administración de un componente, consulte la propia documentación del producto correspondiente.

En este capítulo se introducen conceptos de supervisión y se presenta la arquitectura de Monitoring Framework.

Este capítulo contiene las siguientes secciones:

- “Componentes de Monitoring Framework y Consola de supervisión” en la página 15
- “Funcionamiento de la supervisión en Java ES” en la página 16
- “Secuencia de instalación recomendada” en la página 20

Componentes de Monitoring Framework y Consola de supervisión

Sun Java System Monitoring Framework proporciona la infraestructura para instrumentar componentes y exponer sus atributos para su observación. Define una jerarquía de objetos supervisados llamada *Common Monitoring Model (CMM)*, que se basa en la especificación de Common Information Model (CIM) estándar del sector. Cada componente de producto expone los objetos que representan sus atributos observables, y el *agente de nodo* agrega una

vista de varios componentes en un host. Monitoring Framework proporciona también el mecanismo para recopilar estadísticas de funcionamiento y definir alarmas en función de los umbrales definidos por el usuario.

Sun Java System Consola de supervisión es la interfaz gráfica utilizada para supervisar componentes de Java ES. Incluye un *agente principal* que se conecta a todos los agentes de nodo de una implementación de Java ES. Consola de supervisión es una aplicación basada en web que se basa en Sun Java SystemWeb Console para poder acceder a ella en cualquier ubicación a través de HTTP. En la pantalla principal, proporciona un estado resumido de todos los componentes habilitados, incluyendo las alarmas que se hayan activado. A continuación, se puede acceder a la jerarquía de los objetos supervisados en cada componente y ver el estado detallado y los valores en tiempo real de todos los atributos supervisados. La interfaz de Consola de supervisión permite visualizar los detalles de cualquier alarma, aceptarla o crear reglas de supervisión basadas en cualquier atributo.

Funcionamiento de la supervisión en Java ES

La supervisión es el proceso completo de recopilación de datos de tiempo de ejecución, de exposición de dichos datos y de cálculo de la calidad de criterios de servicio para que el administrador del sistema pueda evaluar el rendimiento y recibir notificación de alarmas. Durante el funcionamiento en el tiempo de ejecución, los administradores sólo deben interactuar con Consola de supervisión para ver estadísticas de rendimiento, crear reglas para supervisar de forma automática y aceptar alarmas. No obstante, para la configuración, solución de problemas y supervisión avanzada, ayuda a comprender la arquitectura de Monitoring Framework y cómo se conecta a Consola de supervisión.

La supervisión en Java ES se basa en los siguientes conceptos:

- Common Monitoring Model (CMM) asegura que todos los componentes de Java ES expongan valores y objetos uniformes correspondientes a atributos comparables.
- Los objetos Java definidos mediante las interfaces de CMM proporcionan una instrumentación estandarizada para los componentes del producto.
- Un agente de nodo expone todos los objetos supervisados de todos los componentes instalados en un sistema, y administra las estadísticas, reglas y alarmas de dichos objetos.
- Un agente principal ubicado en un host independiente agrega todos los objetos supervisados de todos los agentes de nodo y pone los datos a disposición de Consola de supervisión.

En las siguientes secciones se describe con más detalle cada uno de estos conceptos de la arquitectura de supervisión.

Common Monitoring Model (CMM)

La base de un mecanismo de supervisión estandarizado es definir qué objetos se supervisan y la adopción de estos objetos en todos los componentes supervisados. Para este fin, la arquitectura de supervisión define Common Monitoring Model (CMM) como una ampliación de Common Information Model (CIM) mantenido por DMTF (Distributed Management Task Force). CMM es tanto un modelo de información que especifica objetos supervisados tales como un equipo, una aplicación, etc., como un modelo de datos que especifica valores uniformes tales como los valores de estado de funcionamiento. Como parte del modelo de información, CMM define también los atributos de un objeto (por ejemplo, el número de solicitudes gestionadas por un servicio) y las relaciones entre objetos, tales como el hecho de que un servicio se encuentre alojado en un determinado equipo.

Gracias a CMM, conceptos tales como aplicaciones, servicios, puntos de acceso, etc., son los mismos para todos los componentes del producto, aunque la implementación subyacente sea distinta. Por ejemplo, Web Server podría exponer un servicio que gestiona solicitudes HTTP, mientras que Directory Server podría exponer un servicio que gestiona solicitudes LDAP. No obstante, el objeto estándar capturará lo que es común a estas dos funciones, por ejemplo, la capacidad de medir el número de solicitudes gestionadas, el tiempo medio de respuesta a una solicitud en un determinado periodo de tiempo, etc.

Además, ciertos valores de datos se estandarizan para que su significado sea uniforme en todo el sistema. Por ejemplo, el estado de funcionamiento `DEGRADED` significa siempre que un servicio está aún disponible pero que el rendimiento se ha reducido de forma significativa, independientemente del componente de producto que se esté supervisando.

La especificación de CMM está incluida en las clases e interfaces Java utilizadas para la instrumentación, las cuales se describen en el [Apéndice A](#).

Instrumentación de CMM

En Monitoring Framework, la instrumentación es un conjunto de clases e interfaces Java que implementan las definiciones de CMM. Para la nueva función de supervisión incluida en Java ES, los componentes del producto han instrumentado su código para instanciar los objetos de CMM y exponer los valores de tiempo de ejecución mediante los atributos de los objetos supervisados. Los objetos de CMM que cada componente implementa determinan qué se puede supervisar y, por este motivo, algunos componentes exponen menos atributos que otros. La lista de objetos y atributos que cada componente del producto expone para la supervisión se encuentra en el [Apéndice B](#).

Agentes de nodo

En terminología de supervisión, un nodo es un host lógico identificado por una dirección IP o un nombre de dominio exclusivos totalmente cualificados. Un nodo puede ser un sistema completo o una zona Solaris configurada como sistema virtual. El agente de nodo se comunica con todos los componentes instrumentados de ese host y expone todos sus objetos supervisados. El agente de nodo administra también toda la lógica para recopilar estadísticas de rendimiento, supervisar umbrales definidos en reglas y generar alarmas para los objetos supervisados que contiene.

En el siguiente diagrama se representa el contenido de un agente de nodo en un único host que tiene instancias de tres componentes del producto Java ES. También se muestra la forma en que la instrumentación se instancia en el agente de nodo para exponer los valores que proporcionan los componentes del producto.

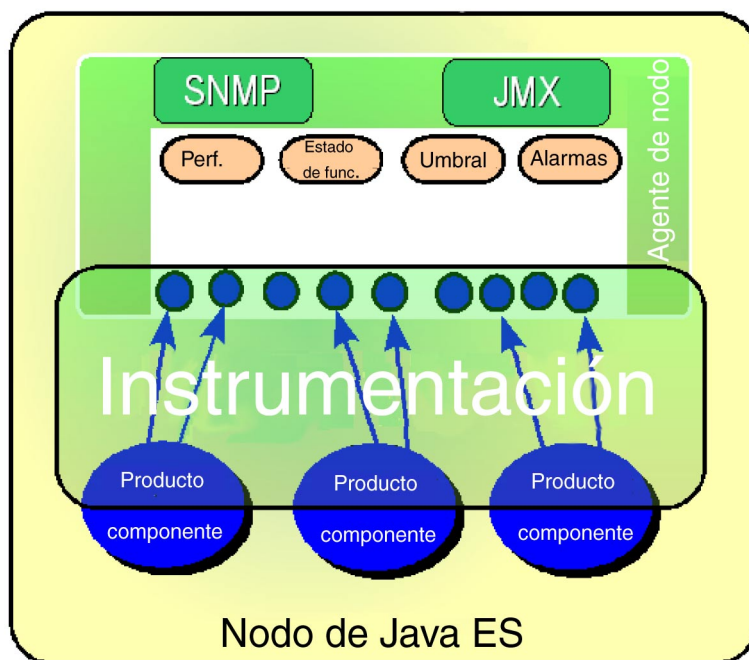


FIGURA 1-1 Diagrama de un agente de nodo

El agente de nodo se implementa como un módulo cargado en Common Agent Container, que es en sí una máquina virtual de Java. La implementación del agente de nodo se basa en JMX (Java Management Extension), que es la extensión estándar de Java para la supervisión y la administración remota. Las aplicaciones de supervisión habilitadas con JMX que entienden el modelo CMM pueden acceder a los objetos supervisados del agente de nodo. Utilizando la

funcionalidad JMX, el agente de nodo puede exponer también ciertos objetos supervisados mediante el protocolo de supervisión de red simple (SNMP).

Agente principal

El agente principal se implementa en un equipo independiente como parte de la instalación de Consola de supervisión. El agente principal se configura con el nombre o dirección de todos los nodos para que pueda agregar los objetos supervisados de todos los agentes de nodo. El agente principal se basa también en JMX, que lo utiliza para comunicarse con los agentes de nodo, e igualmente se carga en su Common Agent Container local.

El siguiente diagrama representa un agente principal conectado a dos nodos. La Consola de supervisión se conecta al agente principal para supervisar los tres componentes en cada nodo. Si desea utilizar SNMP para supervisar, deberá realizar la conexión a cada uno de los nodos por separado, ya que el agente principal no agrega atributos de SNMP. El agente principal está diseñado para utilizarse con la Consola de supervisión solamente y no se puede acceder a él mediante otras aplicaciones de supervisión.

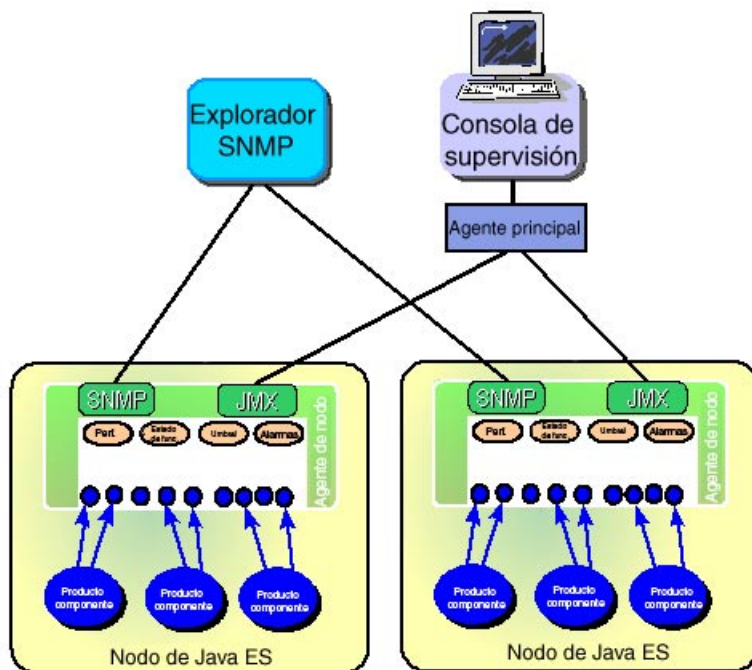


FIGURA 1-2 Diagrama de la arquitectura de supervisión general

Secuencia de instalación recomendada

Si va a evaluar o implementar la función de supervisión de Java ES, le resultará más fácil realizar la instalación en el siguiente orden:

1. Instale y configure todos los componentes de la implementación según las recomendaciones e instrucciones indicadas en el documento *Guía de instalación de Sun Java Enterprise System 5 para UNIX*.
2. Habilite y configure Monitoring Framework para todos los componentes supervisados, como se describe en el [Capítulo 2](#).
3. Instale Consola de supervisión en un host independiente, inicie el agente principal y, a continuación, inicie el servidor web, como se describe en el [Capítulo 3](#). Todos los componentes supervisados deben ser visibles entonces y supervisarse activamente en Consola de supervisión.

Nota – Debido a una incompatibilidad del agente de nodo y del agente principal en esta versión, Consola de supervisión debe instalarse en un host que no contenga ningún otro componente de Java ES. Consulte [“Solución de problemas de la Consola de supervisión” en la página 68](#) para obtener más información.

Si modifica los componentes implementados después de habilitar la supervisión, deberá reiniciar el contenedor del agente principal y el servidor web de Consola de supervisión como se describe en la sección [“Solución de problemas de Monitoring Framework” en la página 29](#).

Habilitación y configuración de Monitoring Framework

Como se describe en [“Funcionamiento de la supervisión en Java ES” en la página 16](#), Monitoring Framework proporciona la instrumentación y el agente de nodo que necesitan todos los componentes supervisados. Por tanto, Monitoring Framework es un componente compartido que se instala de forma automática cuando se instala un componente supervisado mediante el instalador de Java Enterprise System.

No obstante, muchos de los componentes supervisados no tienen la función de supervisión habilitada de manera predeterminada, y algunos requieren una configuración adicional para que aparezcan en el agente de nodo. Realice los procedimientos indicados en este capítulo para cada uno de los componentes del producto que haya instalado.

Nota – Se recomienda instalar y configurar todos los componentes del producto que vaya a ejecutar en un determinado host antes de realizar cualquiera de los procedimientos de este capítulo. Antes de realizar cualquier instalación o configuración, consulte el documento *Notas de la versión Sun Java Enterprise System 5 para UNIX*.

Estos procedimientos utilizan el comando `mfwksetup` que normalmente no es necesario y que, por tanto, no se documenta.

Este capítulo contiene las siguientes secciones:

- [“Diseño de los directorios instalados” en la página 22](#)
- [“Utilización de Monitoring Framework con Access Manager” en la página 23](#)
- [“Utilización de Monitoring Framework con Application Server” en la página 24](#)
- [“Utilización de Monitoring Framework con Calendar Server” en la página 25](#)
- [“Utilización de Monitoring Framework con Directory Server” en la página 25](#)
- [“Utilización de Monitoring Framework con Instant Messaging” en la página 26](#)
- [“Utilización de Monitoring Framework con Messaging Server” en la página 27](#)
- [“Utilización de Monitoring Framework con Portal Server” en la página 27](#)
- [“Utilización de Monitoring Framework con Web Server” en la página 28](#)
- [“Configuración de Common Agent Container” en la página 28](#)

- “Solución de problemas de Monitoring Framework” en la página 29
- “Comando mfwkadm” en la página 30

Diseño de los directorios instalados

Como componente compartido, Monitoring Framework se instala de forma automática cuando es necesario. Con respecto al nombre del paquete instalado en el sistema operativo, consulte el Capítulo 5, “Listado de paquetes instalables” de *Guía de referencia de instalación de Sun Java Enterprise System para UNIX*. En la siguiente tabla se describen los directorios del paquete de Monitoring Framework. El directorio de instalación predeterminado *base-de-mfwk* tiene el siguiente significado, como se describe en la sección “[Rutas predeterminadas y nombres de archivo](#)” en la página 9:

- Sistemas Solaris: /opt/SUNWmfwk
- Sistemas Linux: /opt/sun/mfwk

TABLA 2-1 Directorios que utiliza Monitoring Framework

Ruta	Descripción del contenido
<i>base-de-mfwk</i> /config	Plantilla de un archivo de configuración
Sistemas Solaris: <i>base-de-mfwk</i> /lib	Archivos de Java Archive (.jar)
Sistemas Linux: <i>base-de-mfwk</i> /share/lib	
Sistemas Solaris: <i>base-de-mfwk</i> /lib	Archivos de bibliotecas de tiempo de ejecución de 32-bits (.so)
Sistemas Linux: <i>base-de-mfwk</i> /share/lib	
Sistemas Solaris SPARC®: <i>base-de-mfwk</i> /lib/sparcv9	Archivos de bibliotecas de tiempo de ejecución de 64-bits (.so)
Sistemas Solaris x86: <i>base-de-mfwk</i> /amd64	
Sistemas Linux: <i>base-de-mfwk</i> /lib64	
<i>base-de-mfwk</i> /bin	Secuencias de comandos públicas y valores binarios privados
<i>base-de-mfwk</i> /mib	Versiones de texto de MIB SNMP compatibles con Monitoring Framework
<i>base-de-mfwk</i> /xml	Plantillas de descriptores de Common Agent Container para agente y agente principal (implementados por el comando mfwksetup)
<i>base-de-mfwk</i> /dtd	Archivos DTD para la funcionalidad OSS/J.
/etc/ <i>base-de-mfwk</i> /config	Archivos de configuración, incluyendo los relacionados con la seguridad

TABLA 2-1 Directorios que utiliza Monitoring Framework (Continuación)

Ruta	Descripción del contenido
/etc/base-de-mfwk/xml	Descriptores de Common Agent Container para agentes y ejemplos
/var/base-de-mfwk/logs	Archivos de registro de Monitoring Framework
/var/base-de-mfwk/reports	Directorio base para informes de reglas de supervisión
/var/base-de-mfwk/alarms	Repositorio para archivos de alarma

Utilización de Monitoring Framework con Access Manager

De manera predeterminada, la supervisión está habilitada en Access Manager, pero una limitación impide que los objetos supervisados aparezcan en la Consola de supervisión.

Consulte la sección “Instrumentación de Access Manager” en la página 73 para obtener la lista de objetos y atributos que pueden supervisarse.

▼ Para habilitar la supervisión en Access Manager

- 1 Inhabilite temporalmente la supervisión en Access Manager con los siguientes comandos:

```
cacaoadm unregister-module com.sun.cmm.am.xml
cacaoadm restart
```

- 2 Abra el archivo de descriptor XML de Access Manager para editar:

```
vi /etc/base-de-AccessMgr/config/com.sun.cmm.am.xml
```

- 3 Localice las líneas que contengan lo siguiente:

```
<param-name>Product Name</param-name>
<param-value>Access Manager</param-value>
```

y modifique la segunda línea a:

```
<param-value>Java ES Access Manager</param-value>
```

Guarde el archivo y salga del editor.

- 4 Registre el módulo XML modificado:

```
base-de-mfwk/bin/mfwksetup -u /etc/base-de-AccessMgr/config/com.sun.cmm.am.xml
base-de-mfwk/bin/mfwksetup -r /etc/base-de-AccessMgr/config/com.sun.cmm.am.xml
```

- 5 Reinicie Common Agent Container:

```
cacaoadm restart
```

Errores más frecuentes

Al no haberse comprobado el comportamiento con contenedores web de otros fabricantes, la supervisión se inhabilita de manera predeterminada cuando Access Manager se implementa en Websphere o Weblogic. Puede habilitar la supervisión según se describe en [“Para inhabilitar y rehabilitar selectivamente la supervisión” en la página 59](#), aunque esta configuración no está admitida.

Utilización de Monitoring Framework con Application Server

Consulte la sección [“Instrumentación de Application Server” en la página 73](#) para obtener la lista de objetos y atributos que pueden supervisarse.

▼ Para habilitar la supervisión en Application Server

- 1 **Edite el archivo `/var/base-de-AppServer/domains/domain1/config/domain.xml` y cambie todas las configuraciones de `module-monitoring-level` de OFF a HIGH. Alternativamente:**

- a. **Inicie una sesión en la consola de administración de Application Server en la dirección `https:// nombre-de-host:4849`.**
- b. **Seleccione Configurations (Configuraciones) y, a continuación, server-config (Admin Config).**
- c. **Defina el valor de Monitoring (Supervisión) como HIGH.**
- d. **Defina los demás valores como HIGH.**

- 2 **Reinicie Application Server con los siguientes comandos:**

```
cd base-de-AppServer/appserv/bin
asadmin stop-domain domain1
asadmin start-domain user myUser domain1
```

Introduzca la contraseña para `myUser` cuando se le solicite.

- 3 **Si ha implementado y supervisado una instancia de Portal Server con Application Server, el proceso de reinicio de Application Server interferirá con la supervisión de Portal Server. Para que la instancia de Portal Server aparezca en la Consola de supervisión, debe visitar una página de portal en un explorador. Por ejemplo, cargue la página `http://portalserv.example.com:8080/portal` para posibilitar la supervisión de `portalserv.example.com`.**

Errores más frecuentes

Debido a una limitación, los objetos supervisados para Application Server se eliminan de Monitoring Framework cuando Application Server se para o está inactivo. Cuando esto ocurre, Application Server desaparece de la Consola de supervisión y no se puede seguir supervisando.

Utilización de Monitoring Framework con Calendar Server

Consulte la sección “[Instrumentación de Calendar Server](#)” en la [página 73](#) para obtener la lista de objetos y atributos que pueden supervisarse.

▼ Para habilitar la supervisión en Calendar Server

1 Edite el archivo `ics.conf`:

```
vi base-de-CalServ/cal/config/ics.conf
```

2 Añada la línea:

```
local.mfagent.enable="yes"
```

3 Registre el módulo XML de Calendar Server:

```
base-de-mfwk/bin/mfwksetup -r /opt/SUNWics5/cal/lib/com.sun.cmm.cs.xml
```

4 Defina la variable de entorno `LD_LIBRARY_PATH` como se indica a continuación:

```
LD_LIBRARY_PATH=base-de-mfwk/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

5 Reinicie Calendar Server:

```
cd base-de-CalServ/cal/sbin/
./stop-cal
./start-cal
```

6 Reinicie Common Agent Container:

```
cacaoadm restart
```

Utilización de Monitoring Framework con Directory Server

Consulte la sección “[Instrumentación de Directory Server](#)” en la [página 74](#) para obtener la lista de objetos y atributos que pueden supervisarse.

▼ Para habilitar la supervisión en Directory Server

- 1 Cree un archivo de contraseña temporal:

```
echo -n contraseña > /tmp/pwd
```

- 2 Habilite el complemento de supervisión con el siguiente comando:

```
base-de-DirServ/ds6/bin/dscfg enable-plugin -e -p 389 -w /tmp/pwd "Monitoring Plugin"
```

- 3 Reinicie Directory Server:

```
cd base-de-DirServ/ds6/bin  
./dsadm restart /var/base-de-DirServ/instanciaDS/
```

Utilización de Monitoring Framework con Instant Messaging

Consulte la sección “[Instrumentación de Instant Messaging](#)” en la [página 74](#) para obtener la lista de objetos y atributos que pueden supervisarse.

▼ Para habilitar la supervisión con Instant Messaging

- 1 Abra el archivo de descriptor XML de Instant Messaging para editar:

```
vi /etc/base-de-IM/default/com.sun.cmm.im.xml
```

- 2 Cambie la ubicación de instalación de *base-de-IM* a */etc/base-de-IM/default*.

- 3 Registre el descriptor XML modificado de Instant Messaging:

```
base-de-mfwk/bin/mfwksetup -r /etc/base-de-IM/default/com.sun.cmm.im.xml
```

- 4 Habilite la instrumentación añadiendo la siguiente línea al archivo

```
base-de-IM/config/iim.conf:  
iim_server.monitor.enable = true
```

- 5 Reinicie Instant Messaging con los siguientes comandos:

```
cd base-de-IM/sbin  
./imadmin stop  
./imadmin start
```

- 6 Reinicie Common Agent Container:

```
cacaoadm restart
```

Utilización de Monitoring Framework con Messaging Server

Consulte la sección “[Instrumentación de Messaging Server](#)” en la [página 74](#) para obtener la lista de objetos y atributos que pueden supervisarse.

▼ Para habilitar la supervisión en Messaging Server

- 1 **Habilite la instrumentación con el siguiente comando:**

```
base-de-MsgServ/sbin/configutil -o local.mfagent.enable -v 1
```

- 2 **Registre el módulo XML de Messaging Server :**

```
base-de-mfwk/bin/mfwksetup -r base-de-MsgServ/lib/com.sun.cmm.ms.xml
```

- 3 **Reinicie Messaging Server :**

```
cd base-de-MsgServ/sbin
./stop-msg
./start-msg
```

- 4 **Reinicie Common Agent Container:**

```
cacaoadm restart
```

Utilización de Monitoring Framework con Portal Server

Consulte la sección “[Instrumentación de Portal Server](#)” en la [página 74](#) para obtener la lista de objetos y atributos que pueden supervisarse.

▼ Para habilitar la supervisión en Portal Server

- **Para habilitar Portal Server, el usuario debe iniciar una sesión en**

```
http://nombredehostcompleto:8080/portal/dt
```

De esta forma se compilará el portal JSP, que crea la instancia de portal preparada para la supervisión.

Errores más frecuentes

Cuando se reinicie el Application Server que alberga el Portal Server, deberá rehabilitar manualmente la supervisión con este procedimiento.

Utilización de Monitoring Framework con Web Server

Consulte la sección “[Instrumentación de Web Server](#)” en la [página 74](#) para obtener la lista de objetos y atributos que pueden supervisarse.

▼ Para habilitar la supervisión en Web Server

- 1 **Inicie Web Server con el siguiente comando:**

```
cd /var/base-de-WebServer/https-nombredelhostcompleto/bin
./startserv
```

- 2 **Inicie el servidor de administración:**

```
cd /var/base-de-WebServer/admin-server/bin
./startserv
```

Configuración de Common Agent Container

Common Agent Container es otro componente compartido y del que Monitoring Framework depende para ejecutar el agente de nodo. En función de la secuencia de instalación, es posible que Common Agent Container se detenga y que sea necesario reiniciarlo. Además, Common Agent Container se ha instrumentado y también se puede supervisar. Para obtener una descripción de los objetos supervisados, consulte “[Instrumentación de Common Agent Container](#)” en la [página 73](#).

Para comprobar si Common Agent Container ya se ha iniciado y, por extensión, también el agente de nodo, ejecute el siguiente comando:

```
cacaoadm status
```

Si aparece un mensaje similar al siguiente, significa que el agente de nodo se está ejecutando:

```
default instance is DISABLED at system startup.
Smf monitoring process:
26996
Uptime: 0 day(s), 0:57
```

Si aparece un mensaje similar al siguiente, significa que el agente de nodo no se está ejecutando:

```
default instance is DISABLED at system startup.
default instance is not running.
```

▼ Para habilitar la supervisión del Common Agent Container

Common Agent Container es un componente compartido que cuenta con instrumentación para posibilitar la supervisión. Como se describe en “[Agentes de nodo](#)” en la [página 18](#), todos los componentes de Java ES existentes en un host o en una zona comparten Common Agent Container y el agente de nodo. Realice esta tarea como root en todos los hosts lógicos de la implementación donde desee supervisar Common Agent Container.

- 1 Si Common Agent Container está ejecutándose, deténgalo con el siguiente comando:

```
cacaoadm stop
```

- 2 Habilite la instrumentación del contenedor en sí:

```
cacaoadm set-param enable-instrumentation=true
```

- 3 Compruebe el valor del parámetro que acaba de definir y reinicie Common Agent Container:

```
cacaoadm get-param enable-instrumentation
cacaoadm start
```

- 4 Cree una contraseña clave:

```
echo -n contraseña > /etc/base-de-mfwk/config/security/password.cacao
```

- 5 Genere la clave:

```
base-de-mfwk/bin/cpgenkey -n cacao -p /etc/base-de-mfwk/config/security/password.cacao
```

- 6 Registre los propios módulos de supervisión de Common Agent Container:

```
cacaoadm register-module /usr/lib/cacao/ext/instrum/config/com.sun.cacao.instrum.xml
cacaoadm register-module /usr/lib/cacao/ext/instrum_jesmf/config/com.sun.cacao.instrum.jesmf.xml
cacaoadm register-module /usr/lib/cacao/ext/instrum_jesmf/config/com.sun.cacao.cmm.xml
```

Solución de problemas de Monitoring Framework

Consulte también los problemas conocidos indicados en las *Notas de la versión Sun Java Enterprise System 5 para UNIX*.

Utilización de Monitoring Framework en plataformas HP-UX

Java Virtual Machine (JVM) en HP-UX no se ajusta de forma predeterminada para el procesamiento intensivo de tareas que Monitoring Framework necesita, y esto puede provocar

excepciones de OutOfMemory . Para configurar JVM, descargue y ejecute la herramienta HPjconfig desde la siguiente ubicación:http://h21007.www2.hp.com/dspp/tech/tech_TechDocumentDetailPage_IDX/1,1701,1620,00.html.

Utilización de Monitoring Framework en Microsoft Windows

La supervisión de componentes de Java ES en la plataforma Windows a través de Monitoring Framework es totalmente compatible, aunque existen algunas diferencias. Por ejemplo, debe realizar la actualización a Java 1.5 o una versión posterior para evitar ciertos problemas conocidos. Con respecto a otros problemas conocidos, consulte el documento *Sun Java Enterprise System 5 Release Notes for Microsoft Windows* .

▼ Para reiniciar un agente de nodo

Si necesita reiniciar el Common Agent Container que alberga un agente de nodo, los componentes supervisados mediante ese agente no se visualizarán en la Consola de supervisión hasta que realice este procedimiento:

1 Reinicie el Common Agent Container que alberga el agente de *nodo*:

```
cacaoadm restart
```

2 Reinicie el Common Agent Container que alberga el agente *principal*: El agente principal se ejecuta en Monitoring Framework en el host o en la zona en que haya instalado la Consola de supervisión.

```
cacaoadm restart
```

El agente principal se volverá a conectar de forma automática con todos los agentes de nodo que supervisaba anteriormente.

3 Reinicie el servidor web que alberga la Consola de supervisión:

```
/usr/sbin/smcwebserver restart
```

Comando mfwkadm

En esta sección se reproduce la página del manual (man) correspondiente al comando mfwkadm, un comando de mantenimiento incluido en la página del manual, sección 1M. Utilice este comando para administrar el contenido de un agente de nodo, incluyendo todos los módulos de componentes en supervisión y las reglas de supervisión (conocidas también como trabajos) que haya definido en este nodo. Algunos de los términos y descripciones de la página del manual (man) se han modificado aquí para que coincidan con los utilizados en este documento.

Resumen

```
mfwkadm --help
mfwkadm start
mfwkadm stop
mfwkadm restart
mfwkadm list-params
mfwkadm list-modules
mfwkadm info Instanciaenejecución
```

Supervisión del rendimiento

```
mfwkadm pm-job observable-classes
mfwkadm pm-job observable-objects [class=Clasedeobjeto] [domain= Dominiodeobjeto]
mfwkadm pm-job observable-attributes class=Clasedeobjeto
mfwkadm pm-job list
mfwkadm pm-job info Nombredetrabajo
mfwkadm pm-job create Nombredetrabajo granularity=Valordenúmeroentero
object=Nombredeobjeto [object=Nombredeobjeto ...]
mfwkadm pm-job delete Nombredetrabajo
mfwkadm pm-job suspend Nombredetrabajo
mfwkadm pm-job resume Nombredetrabajo
```

Supervisión del estado de funcionamiento

```
mfwkadm opstat-job observable-classes
mfwkadm opstat-job observable-objects
[class=Clasedeobjeto] [domain= Dominiodeobjeto]
mfwkadm opstat-job observable-attributes class= Clasedeobjeto
mfwkadm opstat-job list
mfwkadm opstat-job info Nombredetrabajo
mfwkadm opstat-job create Nombredetrabajo granularity=Valordenúmeroentero
object=Nombredeobjeto [object=Nombredeobjeto ...]
mfwkadm opstat-job delete Nombredetrabajo
```

mfwkadm opstat-job suspend *Nombredetrabajo*

mfwkadm opstat-job resume *Nombredetrabajo*

Supervisión del valor de umbral

mfwkadm thrsh-job observable-classes

mfwkadm thrsh-job observable-objects

[class=*Clasedeobjeto*] [domain= *Dominiodeobjeto*]

mfwkadm thrsh-job observable-attributes class=*Clasedeobjeto*

mfwkadm thrsh-job list

mfwkadm thrsh-job info *Nombredetrabajo*

mfwkadm thrsh-job create *Nombredetrabajo* granularity=*Valordenúmeroentero*

attributeName=*Nombredeatributo* attributeType=*Tipodeatributo*

thresholdValue=*Valordeumbral* thresholdOffset=*Valordedesplazamiento*

thresholdDirection=[RISING | FALLING] object=*Nombredeobjeto*

mfwkadm thrsh-job delete *Nombredetrabajo*

mfwkadm thrsh-job suspend *Nombredetrabajo*

mfwkadm thrsh-job resume *Nombredetrabajo*

Descripción

La utilidad mfwkadm es la interfaz de línea de comandos para administrar el agente de Monitoring Framework, denominado también agente de nodo. El agente de nodo se ejecuta dentro de Common Agent Container. La utilidad mfwkadm se puede utilizar para detener y reiniciar el agente de nodo y para administrar los trabajos de supervisión que realiza. Este comando se debe ejecutar desde el mismo host en el que se ejecuta el agente de nodo. Se debe respetar el orden de los argumentos de este comando que se indica aquí.

Para cambiar el idioma de los mensajes de salida, defina la variable de entorno LC_MESSAGE según su *configuración regional*. El comando mfwkadm utilizará los mensajes incluidos en el archivo llamado JesmfMessages_ *configuración regional*.pm existente en el directorio lib/resources. Si la configuración regional no tiene ningún archivo de mensajes correspondiente, o si no se especifica ninguna configuración regional, el comando mfwkadm utilizará el conjunto de mensajes predeterminado del archivo JesmfMessages .pm.

La utilidad `mfwkadm` tiene los siguientes subcomandos. Los marcados con un asterisco (*) requieren que Common Agent Container esté ejecutándose y que el agente de nodo esté cargado:

- `start`
- `stop`
- `restart`
- `list-params (*)`
- `list-modules (*)`
- `info (*)`
- `pm-job (*)`
- `opstat-job (*)`
- `thrsh-job (*)`

Dependiendo del número de módulos de Common Agent Container que se cargue, se producirá un retardo de unos segundos a unos minutos entre el inicio del agente de nodo y la disponibilidad de la utilidad `mfwkadm`. Durante este periodo de tiempo, los comandos fallan mostrando un mensaje explícito.

Opciones

Se admiten las siguientes opciones.

`--help` Muestra el resumen de utilización.

Subcomandos

`start`

Inicia el agente de nodo de Monitoring Framework y sus módulos de producto componente asociados sin detener Common Agent Container.

Esta acción implementa en Common Agent Container el agente de nodo primero y, a continuación, los módulos de producto componente asociados. Esta utilidad es un programa de ajuste además de los subcomandos `lock` y `undeploy` de la utilidad `cacaoadm`.

El subcomando `start` sólo inicia el agente de nodo y los módulos del componente de Java ES asociados a Monitoring Framework. Los módulos del componente tienen el prefijo `com.sun.cmm`.

Seguridad: el subcomando `start` sólo lo puede ejecutar el usuario que inicie Common Agent Container. En caso contrario, aparecerá un mensaje de error similar al mostrado a continuación:

```
Error occured in mfwkadm
Problem running /usr/sbin/cacaoadm unlock com.sun.mfwk 2>&1.
Stdout/Stderr: This command must be run by user: [root].
```

stop

Detiene el agente de nodo de Monitoring Framework y sus módulos del componente asociados de Java ES en Common Agent Container.

Esta acción detiene primero los módulos del componente de Java ES implementados en Common Agent Container y, a continuación, detiene el agente de nodo. Esta utilidad es un programa de ajuste además de los subcomandos `lock` y `unlock` de la utilidad `cacaoadm`.

El subcomando `stop` detiene sólo los módulos del componente de Java ES asociados a Monitoring Framework y, a continuación, el agente de nodo en sí. Los módulos del componente tienen el prefijo `com.sun.cmm`.

Seguridad: el subcomando `stop` sólo lo puede ejecutar el usuario que inicie Common Agent Container. En caso contrario, aparecerá un mensaje de error similar al mostrado a continuación:

```
Error occured in mfwkadm
Problem running /usr/sbin/cacaoadm unlock com.sun.mfwk 2>&1.
Stdout/Stderr: This command must be run by user: [root].
```

restart

Reinicia el agente de nodo de Monitoring Framework y sus módulos del componente asociados de Java ES en Common Agent Container.

Esta acción intentará detener y, a continuación, iniciar el agente de nodo y sus módulos asociados en Common Agent Container de la misma manera que los subcomandos `stop` y `start`.

Seguridad: el subcomando `restart` sólo lo puede ejecutar el usuario que inicie Common Agent Container. En caso contrario, aparecerá un mensaje de error similar al mostrado a continuación:

```
Error occured in mfwkadm
Problem running //usr/sbin/cacaoadm unlock com.sun.mfwk 2>&1.
Stdout/Stderr: This command must be run by user: [root].
```

list-params

Enumera todos los parámetros de configuración relacionados con el agente de nodo de Monitoring Framework.

Seguridad: no existe ninguna restricción de usuario para este comando.

list-modules

Muestra una lista de los módulos de producto componente que implementan el Common Monitoring Model (CMM) y que se cargan en Common Agent Container. Este subcomando

enumera también todas las instancias en ejecución de cada componente de Java ES instalado. Cada componente puede tener una, varias o ninguna instancia en ejecución.

Seguridad: para los usuarios que no sean el que ha iniciado Common Agent Container, la lista de componentes de Java ES instalados no incluye instancias de componente.

info Instanciaenejecución

Muestra información sobre la *Instanciaenejecución* nombrada. La *Instanciaenejecución* debe coincidir con una instancia en ejecución enumerada en la salida del subcomando `list-modules`. La información mostrada incluye lo siguiente:

- Para cada tipo de trabajo de supervisión, todos los objetos observables asociados a la instancia en ejecución, ordenados según su nombre de clase. Los objetos observables son aquellos en los que puede crear un trabajo de supervisión del rendimiento, un trabajo de estado de funcionamiento o un trabajo de supervisión del umbral utilizando los subcomandos `pm-job`, `opstat-job` o `thrsh-job`, respectivamente.
- Para cada clase de objetos observables, todos sus atributos observables, incluyendo el nombre y tipo de cada uno de ellos.

Seguridad: para los usuarios que no sean el que ha iniciado Common Agent Container, no aparecerá ninguna información.

Supervisión del rendimiento

`pm-job observable-classes`

Muestra la lista de todas las clases de objetos actualmente observables para los que se pueden crear trabajos de supervisión del rendimiento.

`pm-job observable-objects [class= Clasedeobjeto] [domain= Dominiodeobjeto]`

Muestra la lista de todos los objetos actualmente observables para los que se pueden crear trabajos de supervisión del rendimiento. De manera predeterminada, se enumerarán todos los objetos de todas las clases observables y en todos los dominios. La lista de objetos se ordena según su nombre de clase.

class=Clasedeobjeto

Al especificar la *Clasedeobjeto* opcional, la salida se limitará a los objetos observables de esa clase específica. La *Clasedeobjeto* debe ser una de las clases enumeradas por el subcomando `pm-job observable-classes`.

domain=Dominiodeobjeto

Al especificar el *Dominiodeobjeto* opcional, la salida se limitará a los objetos observables de ese dominio. El dominio de un objeto es la cadena que precede a los dos puntos (:) en el nombre de un objeto.

`pm- job observable-attributes class=Clasedeobjeto`

Muestra la lista de todos los atributos observables de la *Clasedeobjeto* especificada. Los atributos aparecen con su nombre y tipo. La *Clasedeobjeto* debe ser una de las clases que admite trabajos de supervisión del rendimiento, según se enumeran mediante el subcomando `pm- job observable-classes`.

`pm- job list`

Muestra la lista de todos los trabajos de supervisión del rendimiento actualmente definidos. Los trabajos se enumeran para cada objeto que tenga un trabajo de rendimiento definido, y los objetos se ordenan según su nombre de clase. La información mostrada para cada trabajo es la misma que la mostrada por el subcomando `pm- job info`.

Seguridad: para los usuarios que no sean el que ha iniciado Common Agent Container, no aparecerá ningún trabajo.

`pm- job info Nombredetrabajo`

Muestra información detallada sobre un trabajo de supervisión del rendimiento llamado *Nombredetrabajo*. El *Nombredetrabajo* debe ser uno de los mostrados por el subcomando `pm- job list`. Este subcomando muestra la siguiente información:

- Nombre del trabajo de supervisión del rendimiento.
- Tipo de trabajo de supervisión del rendimiento, ya sea "por objeto o por clase". Los trabajos por objeto supervisan una o varias instancias de objetos con nombre, mientras que los trabajos por clase supervisan todas las instancias de una clase de objeto. Tenga en cuenta que no es posible crear trabajos por clase con la utilidad `mfwkadm`.
- Estado del trabajo de supervisión del rendimiento: activo en servicio, activo fuera de servicio o suspendido. Un trabajo activo en servicio está programado actualmente para ejecutarse y está recopilando datos. Un trabajo activo fuera de servicio está ejecutándose pero no recopila datos, ya que la hora actual está fuera de su programación de funcionamiento. Un trabajo suspendido no está ejecutándose ni recopilando datos. Utilice los subcomandos `pm- job suspend` y `pm- job resume` para cambiar el estado de ejecución de un trabajo de supervisión del rendimiento.
- La granularidad en segundos del trabajo de supervisión del rendimiento. Se trata del intervalo para la recopilación de datos por parte de este trabajo.
- El periodo de informe del trabajo de supervisión. El periodo de informe de la granularidad equivale a la frecuencia de notificación. Por ejemplo, si el periodo de granularidad es de 10 segundos y el periodo de informe es de 6, un trabajo que informe por evento recopilará datos cada 10 segundos y enviará una notificación que incluye 6 informes cada 60 segundos (10*6). Si el trabajo informa también por archivo, enviará un evento cada 60 segundos que contiene las ubicaciones de los 6 archivos generados.
- Si el trabajo de supervisión del rendimiento informa por evento. Esto significa que los resultados del trabajo de supervisión del rendimiento se envían como notificaciones a un cliente registrado.

- Si el trabajo de supervisión del rendimiento informa por archivo. Esto significa que los informes del trabajo de supervisión del rendimiento se escriben en archivos locales y notificaciones que contienen los nombres de archivo que se envían a clientes registrados.
- El formato del informe del trabajo de supervisión del rendimiento, que es siempre XML.
- La programación del trabajo de supervisión del rendimiento. La programación especifica los días y horas en que el trabajo está activo en servicio o activo fuera de servicio (recopilando datos o no, respectivamente). A continuación, para un trabajo por objeto:
- La lista de objetos observados, ordenados por nombre.
- Si sólo se especifica un subconjunto de atributos observables, los atributos observados de los objetos observados se enumerarán por nombre y tipo. Y para un trabajo por clase:
- La lista de clases observadas, ordenadas por nombre.
- Si sólo se especifica un subconjunto de atributos observables, los atributos observados de las clases observadas se enumerarán por nombre y tipo. Estos atributos son comunes a todas las clases.

Seguridad: para los usuarios que no sean el que ha iniciado Common Agent Container, no aparecerá ninguna información.

`pm-job create Nombredetrabajo granularity= Valordenúmeroentero object=Nombredeobjeto [object= Nombredeobjeto ...]`

Crea un trabajo de supervisión del rendimiento en uno o varios objetos. El comando `mfwkadm` no puede crear trabajos por clase. Al crear trabajos de supervisión del rendimiento, se pueden definir los siguientes parámetros:

Nombredetrabajo

Cadena que identifica exclusivamente el trabajo de supervisión del rendimiento. El *Nombredetrabajo* no puede estar utilizándose ya en ningún otro trabajo de supervisión del rendimiento.

granularity=Valordenúmeroentero

Tiempo especificado en segundos entre el inicio de dos recopilaciones sucesivas de datos de medición, mientras el trabajo está activo en servicio. Ejemplos de periodo de granularidad pueden ser 300 segundos (5 minutos), 900 segundos (15 minutos), 1.800 segundos (cada media hora) y 3.600 segundos (cada hora). Un periodo de granularidad de 300 segundos es suficiente en la mayoría de los casos. Para algunas mediciones, puede ser más significativo recopilar datos con periodos mayores de granularidad.

object=Nombredeobjeto [object= Nombredeobjeto ...]

Uno o varios objetos observables a partir de los que el trabajo de supervisión del rendimiento recopilará datos y sobre los que informará. El *Nombredeobjeto* debe ser uno de los mostrados por los subcomandos `pm-job list` o `pm-job observable-objects`. Al especificar parámetros `multiple object= Nombredeobjeto`, se creará un único trabajo de supervisión del rendimiento que supervisará varios objetos.

Seguridad: este subcomando sólo lo puede ejecutar el usuario que inicie Common Agent Container.

`pm-job delete Nombredetrabajo`

Elimina un trabajo de supervisión del rendimiento llamado *Nombredetrabajo*. El *Nombredetrabajo* debe ser uno de los mostrados por el subcomando `pm-job list`.

Seguridad: este subcomando sólo lo puede ejecutar el usuario que inicie Common Agent Container.

`pm-job suspend Nombredetrabajo`

Suspende un trabajo de supervisión del rendimiento llamado *Nombredetrabajo*. Un trabajo suspendido no está activo y no recopilará datos, independientemente de su programación. No obstante, el trabajo permanece definido y se puede activar de nuevo con el subcomando `pm-job resume`. El *Nombredetrabajo* debe ser uno de los mostrados por el subcomando `pm-job list`.

Seguridad: este subcomando sólo lo puede ejecutar el usuario que inicie Common Agent Container.

`pm-job resume Nombredetrabajo`

Reanuda un trabajo de supervisión del rendimiento llamado *Nombredetrabajo*. Un trabajo reanudado comenzará a recopilar datos y a enviar informes según su programación. El *Nombredetrabajo* debe ser uno de los mostrados por el subcomando `pm-job list`. Es el equivalente al subcomando `pm-job suspend`.

Seguridad: este subcomando sólo lo puede ejecutar el usuario que inicie Common Agent Container.

Supervisión del estado de funcionamiento

`opstat-job observable-classes`

Muestra la lista de todas las clases de objetos actualmente observables para los que se pueden crear trabajos de supervisión del estado de funcionamiento.

`opstat-job observable-objects [class= Clasedeobjeto] [domain= Dominiodeobjeto]`

Muestra la lista de todos los objetos actualmente observables para los que se pueden crear trabajos de supervisión del estado de funcionamiento. De manera predeterminada, se enumerarán todos los objetos de todas las clases observables y en todos los dominios. La lista de objetos se ordena según su nombre de clase.

`class= Clasedeobjeto`

Al especificar la *Clasedeobjeto* opcional, la salida se limitará a los objetos observables de esa clase específica. La *Clasedeobjeto* debe ser una de las clases enumeradas por el subcomando `opstat-job observable-classes`.

`domain=Dominio de objeto`

Al especificar el *Dominio de objeto* opcional, la salida se limitará a los objetos observables de ese dominio. El dominio de un objeto es la cadena que precede a los dos puntos (:) en el nombre del objeto.

`opstat-job observable-attributes class= Clase de objeto`

Muestra la lista de todos los atributos observables de la *Clase de objeto* especificada. Los atributos aparecen con su nombre y tipo. La *Clase de objeto* debe ser una de las clases enumeradas por el subcomando `opstat-job observable-classes`.

`opstat-job list`

Muestra la lista de todos los trabajos de supervisión del estado de funcionamiento actualmente definidos. Los trabajos se enumeran para cada objeto que tenga un trabajo de estado de funcionamiento definido, y los objetos se ordenan según su nombre de clase. La información mostrada para cada trabajo es la misma que la mostrada por el subcomando `opstat-job info`.

Seguridad: Para los usuarios que no sean el que ha iniciado Common Agent Container, no aparecerá ningún trabajo.

`opstat-job info Nombre de trabajo`

Muestra información detallada sobre un trabajo de supervisión del estado de funcionamiento llamado *Nombre de trabajo*. El *Nombre de trabajo* debe ser uno de los mostrados por el subcomando `opstat-job list`. Este subcomando muestra la siguiente información:

- Nombre del trabajo de supervisión del estado de funcionamiento.
- Tipo de trabajo de supervisión del estado de funcionamiento, ya sea "por objeto" o "por clase". Los trabajos por objeto supervisan una instancia de objeto con nombre, mientras que los trabajos por clase supervisan todas las instancias de una clase de objeto. Tenga en cuenta que no es posible crear trabajos por clase con la utilidad `mfwkadm`.
- Estado del trabajo de supervisión del estado de funcionamiento: activo en servicio, activo fuera de servicio o suspendido. Un trabajo activo en servicio está programado actualmente para ejecutarse y está recopilando datos. Un trabajo activo fuera de servicio está ejecutándose pero no recopila datos, ya que la hora actual está fuera de su programación de funcionamiento. Un trabajo suspendido no está ejecutándose ni recopilando datos. Utilice los subcomandos `opstat-job suspend` y `opstat-job resume` para cambiar el estado de ejecución de un trabajo de supervisión del estado de funcionamiento.
- La granularidad en segundos del trabajo de supervisión del estado de funcionamiento. Se trata del intervalo para la recopilación de datos por parte de este trabajo.
- Si el trabajo de supervisión del estado de funcionamiento informa por evento. Esto significa que los resultados del trabajo de supervisión del estado de funcionamiento se envían como notificaciones a un cliente registrado.

- Si el trabajo de supervisión del estado de funcionamiento informa por archivo. Esto significa que los informes del trabajo de supervisión del estado de funcionamiento se escriben en archivos locales y notificaciones que contienen los nombres de archivo que se envían a clientes registrados.
- El formato del informe del trabajo de supervisión del estado de funcionamiento, que es siempre XML.
- La programación del trabajo de supervisión del estado de funcionamiento. La programación especifica los días y horas en que el trabajo está activo en servicio o activo fuera de servicio (recopilando datos o no, respectivamente).
- Para un trabajo por objeto, la lista de objetos observados, ordenados por nombre.
- Para un trabajo por clase, la lista de clases observadas, ordenadas por nombre.

Seguridad: para los usuarios que no sean el que ha iniciado Common Agent Container, no aparecerá ninguna información.

`opstat - job create Nombredetrabajo granularity= Valordenúmeroentero
object=Nombredeobjeto [object= Nombredeobjeto ...]`

Crea un trabajo de supervisión del estado de funcionamiento en uno o varios objetos. El comando `mfwkadm` no puede crear trabajos por clase. Al crear trabajos de supervisión del rendimiento, se pueden definir los siguientes parámetros:

Nombredetrabajo

Cadena que identifica exclusivamente el trabajo de supervisión del estado de funcionamiento. El *Nombredetrabajo* no puede estar utilizándose ya en ningún otro trabajo de supervisión del estado de funcionamiento.

`granularity= Valordenúmeroentero`

Tiempo especificado en segundos entre el inicio de dos recopilaciones sucesivas de datos de medición, mientras el trabajo está activo en servicio.

`object=Nombredeobjeto [object= Nombredeobjeto ...]`

Uno o varios objetos observables a partir de los que el trabajo de supervisión del estado de funcionamiento recopilará datos y sobre los que informará. El *Nombredeobjeto* debe ser uno de los mostrados por los subcomandos `opstat - job list` o `opstat - job observable - objects`. Al especificar parámetros multiple `object= Nombredeobjeto`, se creará un único trabajo de supervisión del estado de funcionamiento que supervisará varios objetos.

Seguridad: este subcomando sólo lo puede ejecutar el usuario que inicie Common Agent Container.

`opstat - job delete Nombredetrabajo`

Elimina un trabajo de supervisión del estado de funcionamiento llamado *Nombredetrabajo*. El *Nombredetrabajo* debe ser uno de los mostrados por el subcomando `opstat - job list`.

Seguridad: este subcomando sólo lo puede ejecutar el usuario que inicie Common Agent Container.

`opstat - job suspend Nombredetrabajo`

Suspende un trabajo de supervisión del estado de funcionamiento llamado *Nombredetrabajo*. Un trabajo suspendido no está activo y no recopilará datos, independientemente de su programación. No obstante, el trabajo permanece definido y se puede activar de nuevo con el subcomando `opstat - job resume`. El *Nombredetrabajo* debe ser uno de los mostrados por el subcomando `opstat - job list`.

Seguridad: este subcomando sólo lo puede ejecutar el usuario que inicie Common Agent Container.

`opstat - job resume Nombredetrabajo`

Reanuda un trabajo de supervisión del estado de funcionamiento llamado *Nombredetrabajo*. Un trabajo reanudado comenzará a recopilar datos y a enviar informes según su programación. El *Nombredetrabajo* debe ser uno de los mostrados por el subcomando `opstat - job list`. Es el equivalente al subcomando `opstat - job suspend`.

Seguridad: este subcomando sólo lo puede ejecutar el usuario que inicie Common Agent Container.

Supervisión del valor de umbral

`thrsh - job observable - classes`

Muestra la lista de todas las clases de objetos actualmente observables para los que se pueden crear trabajos de supervisión del umbral.

`thrsh - job observable - objects [class= Clasedeobjeto] [domain= Dominiodeobjeto]`

Muestra la lista de todos los objetos actualmente observables para los que se pueden crear trabajos de supervisión del umbral. De manera predeterminada, se enumerarán todos los objetos de todas las clases observables y en todos los dominios. La lista de objetos se ordena según su nombre de clase.

`class= Clasedeobjeto`

Al especificar la *Clasedeobjeto* opcional, la salida se limitará a los objetos observables de esa clase específica. La *Clasedeobjeto* debe ser una de las clases enumeradas por el subcomando `thrsh - job observable - classes`.

`domain= Dominiodeobjeto`

Al especificar el *Dominiodeobjeto* opcional, la salida se limitará a los objetos observables de ese dominio. El dominio de un objeto es la cadena que precede a los dos puntos (:) en el nombre del objeto.

`thrsh - job observable - attributes class= Clasedeobjeto`

Muestra la lista de todos los atributos observables de la *Clasedeobjeto* especificada. Los atributos aparecen con su nombre y tipo. La *Clasedeobjeto* debe ser una de las clases enumeradas por el subcomando `thrsh - job observable - classes`.

thrsh-job list

Muestra la lista de todos los trabajos de supervisión del umbral actualmente definidos. Los trabajos se enumeran para cada objeto que tenga un trabajo de umbral definido, y los objetos se ordenan según su nombre de clase. La información mostrada para cada trabajo es la misma que la mostrada por el subcomando thrsh-job info.

Seguridad: Para los usuarios que no sean el que ha iniciado Common Agent Container, no aparecerá ningún trabajo.

thrsh-job info *Nombredetrabajo*

Muestra información detallada sobre un trabajo de supervisión del umbral llamado *Nombredetrabajo*. El *Nombredetrabajo* debe ser uno de los mostrados por el subcomando thrsh-job list. Este subcomando muestra la siguiente información:

- Nombre del trabajo de supervisión del umbral.
- Multiplicidad del trabajo de supervisión del umbral. En esta versión, sólo se pueden realizar trabajos de umbral simples que supervisen un atributo en un objeto.
- Estado del trabajo de supervisión del umbral: activo en servicio, activo fuera de servicio o suspendido. Un trabajo activo en servicio está programado actualmente para ejecutarse y está recopilando datos. Un trabajo activo fuera de servicio está ejecutándose pero no recopila datos, ya que la hora actual está fuera de su programación de funcionamiento. Un trabajo suspendido no está ejecutándose ni recopilando datos. Utilice los subcomandos thrsh-job suspend y thrsh-job resume para cambiar el estado de ejecución de un trabajo de supervisión del umbral.
- La granularidad en segundos del trabajo de supervisión del umbral. Se trata del intervalo para la recopilación de datos por parte de este trabajo.
- La programación del trabajo de supervisión del umbral. La programación especifica los días y horas en que el trabajo está activo en servicio o activo fuera de servicio (recopilando datos o no, respectivamente).
- La configuración de la alarma del trabajo de supervisión del umbral. Se trata de la alarma que se activará cuando el valor observado del atributo supervisado cruce el valor de umbral definido. En la indicación se incluye el tipo y gravedad de la alarma.
- El objeto observado del trabajo de supervisión del umbral.
- El nombre de atributo al que se aplica el umbral.
- El valor del umbral que activará una alarma.
- La dirección del progreso del valor que activará una alarma en el umbral, ya sea RISING o FALLING.
- El desplazamiento de tolerancia del umbral. Si la dirección es RISING, la alarma no se activará de nuevo hasta que el atributo observado sea inferior al *Valordeumbral-Valordedesplazamiento*. Si la dirección es FALLING, la alarma no se activará de nuevo hasta que el atributo observado sea superior al *Valordeumbral+Valordedesplazamiento*. Este comportamiento conserva el valor "true" incluso cuando el desplazamiento es cero.

Seguridad: para los usuarios que no sean el que ha iniciado Common Agent Container, no aparecerá ninguna información.

thrsh-job create *Nombredetrabajo* object= *Nombredeobjeto*
 granularity= *Valordenúmeroentero* attributeName= *Nombredeatributo*
 attributeType= *Tipodeatributo* thresholdValue= *Valordeumbral*
 thresholdOffset= *Valordedesplazamiento* thresholdDirection= [RISING|FALLING]

Crea un trabajo de supervisión del umbral que supervisa un atributo en un solo objeto. Al crear trabajos de umbral, se pueden definir los siguientes parámetros:

Nombredetrabajo

Cadena que identifica exclusivamente el trabajo de supervisión del umbral. El *Nombredetrabajo* no puede estar utilizándose ya en ningún otro trabajo de supervisión del umbral.

object= *Nombredeobjeto*

El objeto observable en el que el trabajo de supervisión del umbral recopilará los valores de atributo para compararlos con el umbral. El *Nombredeobjeto* debe ser uno de los mostrados por los subcomandos thrsh-job list o thrsh-job observable-objects

granularity= *Valordenúmeroentero*

Tiempo especificado en segundos entre el inicio de dos observaciones sucesivas del valor de atributo, mientras el trabajo está activo en servicio.

attributeName= *Nombredeatributo*

El nombre del atributo para el que el trabajo de supervisión del umbral recopila valores y los compara con el umbral. El *Nombredeatributo* debe enumerarse mediante los subcomandos thrsh-job info o thrsh-job observable-attributes.

attributeType= *Tipodeatributo*

Tipo de atributo observable que va a supervisarse. El *Tipodeatributo* debe enumerarse mediante los subcomandos thrsh-job info o thrsh-job observable-attributes.

thresholdValue= *Valordeumbral*

El valor del atributo supervisado que provocará que este trabajo de umbral active una alarma al cruzarse en la dirección especificada por thresholdDirection.

thresholdOffset= *Valordedesplazamiento*

El *Valordedesplazamiento* determina la tolerancia del trabajo de umbral al activar alarmas sucesivas. El *Valordedesplazamiento* debe ser cero o un valor positivo. Tras activarse un evento de alarma, no se activará ningún evento de alarma nuevo hasta que el valor del atributo supervisado supere el rango definido por el *Valordedesplazamiento* y la Direccióndeumbral.

thresholdDirection= [RISING|FALLING]

Si la dirección es RISING, el evento de alarma no se activará de nuevo hasta que el valor del atributo observado sea inferior al *Valordeumbral-Valordedesplazamiento*. Si la dirección es FALLING, el evento de alarma no se activará de nuevo hasta que el valor del atributo observado sea superior al *Valordeumbral+Valordedesplazamiento*. Este comportamiento

conserva el valor "true" incluso cuando el *Valordedesplazamiento* es cero.

Seguridad: este subcomando sólo lo puede ejecutar el usuario que inicie Common Agent Container.

`thrsh-job delete Nombredetrabajo`

Elimina un trabajo de supervisión del umbral llamado *Nombredetrabajo*. El *Nombredetrabajo* debe ser uno de los mostrados por el subcomando `thrsh-job list`.

Seguridad: este subcomando sólo lo puede ejecutar el usuario que inicie Common Agent Container.

`thrsh-job suspend Nombredetrabajo`

Suspende un trabajo de supervisión del umbral llamado *Nombredetrabajo*. Un trabajo suspendido no está activo y no recopilará datos, independientemente de su programación. No obstante, el trabajo permanece definido y se puede activar de nuevo con el subcomando `thrsh-job resume`. El *Nombredetrabajo* debe ser uno de los mostrados por el subcomando `thrsh-job list`.

Seguridad: este subcomando sólo lo puede ejecutar el usuario que inicie Common Agent Container.

`thrsh-job resume Nombredetrabajo`

Reanuda un trabajo de supervisión del umbral llamado *Nombredetrabajo*. Un trabajo reanudado comenzará a recopilar datos y a enviar informes según su programación. El *Nombredetrabajo* debe ser uno de los mostrados por el subcomando `thrsh-job list`. Es el equivalente al subcomando `thrsh-job suspend`.

Seguridad: este subcomando sólo lo puede ejecutar el usuario que inicie Common Agent Container.

Ejemplos

En el siguiente escenario de ejemplo se muestra cómo emplear la utilidad `mfwkadm`, junto con sus opciones y subcomandos.

El subcomando `list-modules` muestra las instancias del componente de Java ES en el host actual y sus módulos correspondientes en Common Agent Container. En el siguiente ejemplo se enumeran dos componentes instalados: Directory Server sin instancias en ejecución y Web Server con una instancia en ejecución.

```
$ mfwkadm list-modules
```

```
Installed products and their running instances:
```

```
=====
```

```
Instances for installed product: com.sun.cmm.ds:collectionID=/opt/SUNWdsee/ds6,
name=Sun Java(TM) System Directory Server,type=CMM_InstalledProduct
-----
```

No instance.

```
Instances for installed product: com.sun.cmm.ws:collectionID=/var/opt/SUNWwbsvr7,
name=WebServer,type=CMM_InstalledProduct
-----
```

```
/wsPrefix/com.sun.cmm.ws:name=https-hostname.example.com,type=CMM_ApplicationSystem
```

El siguiente subcomando `info` muestra objetos observables en la instancia de Web Server, con sus clases y atributos para cada tipo de trabajo.

```
$ mfwkadm info /wsPrefix/com.sun.cmm.ws:name=https-hostname.example.com,\\
type=CMM_ApplicationSystem
```

```
Information about running instance: /wsPrefix/com.sun.cmm.ws:
name=https-hostname.example.com,type=CMM_ApplicationSystem
=====
```

```
Observable objects for performance jobs:
-----
```

```
+ Objects of class: com.sun.cmm.settings.CMM_ApplicationSystemSetting
```

```
      /wsPrefix/com.sun.cmm.ws:name=https-hostname.example.com-setting,
type=CMM_ApplicationSystemSetting
```

Observable attributes:

```
Caption [STRING]
ConfigurationDirectory [STRING]
CreationClassName [STRING]
Description [STRING]
DirectoryName [STRING]
ElementName [STRING]
InstanceID [STRING]
Name [STRING]
URL [STRING]
```

```
+ Objects of class: com.sun.cmm.settings.CMM_KeepAliveSetting
```

```
      /wsPrefix/com.sun.cmm.ws:name=process-1-keepalive-setting,
type=CMM_KeepAliveSetting
```

Observable attributes:

```
AllocationUnit [STRING]
Caption [STRING]
ConnectionsUpperBound [LONG]
CreationClassName [STRING]
Description [STRING]
ElementName [STRING]
InputUnit [STRING]
InstanceID [STRING]
LowerAllocationLimit [LONG]
LowerInputLimit [LONG]
LowerOutputLimit [LONG]
Name [STRING]
OtherAllocationUnit [STRING]
OtherInputUnit [STRING]
OtherLowerAllocationLimit [LONG]
OtherLowerInputLimit [LONG]
OtherLowerOutputLimit [LONG]
OtherOutputUnit [STRING]
OtherUpperAllocationLimit [LONG]
OtherUpperInputLimit [LONG]
OtherUpperOutputLimit [LONG]
OutputUnit [STRING]
QueuedUpperBound [LONG]
SecondsTimeout [LONG]
TimeoutUpperBound [LONG]
UpperAllocationLimit [LONG]
UpperInputLimit [LONG]
UpperOutputLimit [LONG]
...
```

El siguiente comando muestra la lista de trabajos de supervisión del rendimiento definidos. En este ejemplo, hay un trabajo de rendimiento llamado `myPerfJob` que supervisa un objeto:

```
$ mfwkadm pm-job list
```

```
BY_OBJECTS performance jobs:
```

```
=====
```

```
Performance job information for: myPerfJob
```

```
-----
```

```
Type:                BY_OBJECTS
State:               ACTIVE_ON_DUTY
Granularity period:  30
Reporting period:    1
By event:            EVENT_SINGLE
By file:             EVENT_SINGLE
```

```

Report format:      XML
Schedule:
    Global start time: Immediately
    Global stop time: Forever
    Weekly schedule: Everyday
    Daily schedule: All day
Observed objects:
    /wsPrefix/com.sun.cmm.ws:name=virtualServer-hostname.example.com-
webApp/-stats,type=CMM_VirtualServerWebModuleStats
Observed attributes:
    All available

BY_CLASSES performance jobs:
=====

No jobs found.

```

El siguiente comando crea un trabajo de supervisión del estado de funcionamiento relacionado con dos objetos observables que se obtienen de los subcomandos `opstat -job info` o `opstat -job observable-objects`:

```

$ mfwkadm opstat-job create myOpStatJob granularity=60 \
object=/wsPrefix/com.sun.cmm.ws:name=process-1,type=CMM_UnixProcess \
object=/wsPrefix/com.sun.cmm.ws:name=process-1-DNSCache1,type=CMM_DnsCache

```

El siguiente comando suspende el trabajo creado anteriormente:

```

$ mfwkadm opstat-job suspend myOpStatJob

```

El siguiente comando muestra las clases observables correspondientes a los trabajos de supervisión del umbral potenciales:

```

$ mfwkadm thrsh-job observable-classes

```

```

Threshold jobs observable classes:
=====

com.sun.cmm.cim.CIM_ScopedSettingData
com.sun.cmm.cim.CIM_SettingData
com.sun.cmm.cim.CIM_StatisticalData
com.sun.cmm.cim.statistics.CIM_EthernetPortStatistics
com.sun.cmm.cim.statistics.CIM_NetworkPortStatistics
com.sun.cmm.cim.statistics.j2ee.CIM_J2eeJVMStats
com.sun.cmm.cim.statistics.j2ee.CIM_J2eeStatistic
com.sun.cmm.settings.CMM_ApplicationSystemSetting
com.sun.cmm.settings.CMM_KeepAliveSetting
com.sun.cmm.settings.CMM_QueueTimeoutSetting
com.sun.cmm.settings.CMM_RFC2788ApplicationSystemSetting

```

```
com.sun.cmm.settings.CMM_ScopedSettingData
com.sun.cmm.settings.CMM_SoftwareResourceSetting
com.sun.cmm.settings.CMM_SWRBufferSetting
com.sun.cmm.settings.CMM_SWRLimitSetting
com.sun.cmm.settings.CMM_SWRQueueSetting
com.sun.cmm.settings.CMM_VirtualServerSetting
com.sun.cmm.statistics.CMM_ApplicationSystemStats
com.sun.cmm.statistics.CMM_ApplicationSystemWatchdogStats
com.sun.cmm.statistics.CMM_ConnectionQueueStats
com.sun.cmm.statistics.CMM_DnsCacheStats
com.sun.cmm.statistics.CMM_EthernetPortStats
com.sun.cmm.statistics.CMM_FileCacheStats
com.sun.cmm.statistics.CMM_HTTPResponsesStats
com.sun.cmm.statistics.CMM_JVMJSR174ExtStats
com.sun.cmm.statistics.CMM_JVMJSR174Stats
com.sun.cmm.statistics.CMM_JVMStats
com.sun.cmm.statistics.CMM_NetworkPortStats
com.sun.cmm.statistics.CMM_OperatingSystemStats
com.sun.cmm.statistics.CMM_ProcessorStats
com.sun.cmm.statistics.CMM_ProcessStats
com.sun.cmm.statistics.CMM_QueueTimeoutStats
com.sun.cmm.statistics.CMM_RFC2788ApplicationTableStats
com.sun.cmm.statistics.CMM_ServiceStats
com.sun.cmm.statistics.CMM_SoftwareResourceStats
com.sun.cmm.statistics.CMM_SolarisEthernetPortStats
com.sun.cmm.statistics.CMM_SolarisNetworkPortStats
com.sun.cmm.statistics.CMM_SolarisOperatingSystemStats
com.sun.cmm.statistics.CMM_SolarisProcessorStats
com.sun.cmm.statistics.CMM_SolarisProcessorSysinfoStats
com.sun.cmm.statistics.CMM_SolarisProcessorVmStats
com.sun.cmm.statistics.CMM_Statistic
com.sun.cmm.statistics.CMM_SWRBufferStats
com.sun.cmm.statistics.CMM_SWRCacheStats
com.sun.cmm.statistics.CMM_SWRLimitStats
com.sun.cmm.statistics.CMM_SWRQueueStats
com.sun.cmm.statistics.CMM_UnixOperatingSystemStats
com.sun.cmm.statistics.CMM_UnixProcessStats
com.sun.cmm.statistics.CMM_VirtualServerWebModuleStats
com.sun.cmm.statistics.CMM_WebModuleStats
```

El siguiente comando muestra los atributos observables de trabajos de umbral que supervisan objetos de clase `com.sun.cmm.statistics.CMM_SWRQueueStats` existentes en el ejemplo anterior:

```
$ mfwkadm thrsh-job observable-attributes \\  
class=com.sun.cmm.statistics.CMM_SWRQueueStats
```

Threshold jobs observable attributes:

```
=====
```

```
Class: com.sun.cmm.statistics.CMM_SWRQueueStats
```

```
Attributes:
```

```

BufferSize [LONG]
EntriesCount [LONG]
EntriesHighWaterMark [LONG]
EntriesLowWaterMark [LONG]
EntriesTotal [LONG]
ErrorCount [INTEGER]
FailedOperations [LONG]
LowerLimit [LONG]
OperationsCount [LONG]
OtherLowerLimit [LONG]
OtherUpperLimit [LONG]
OverflowsCount [LONG]
QueuedCount [LONG]
QueuedHighWater [LONG]
SampleInterval [LONG]
TotalQueuedCount [LONG]
UpperLimit [LONG]

```

El siguiente comando es otro ejemplo de creación de trabajos, aquí con un trabajo de umbral:

```

$ mfwkadm thrsh-job create myThreshJob granularity=30 \
object=/wsPrefix/com.sun.cmm.ws:name=process-1-threadPool-NativePool-stats,\
type=CMM_SWRQueueStats attributeName=EntriesCount attributeType=LONG \
thresholdValue=1000 thresholdOffset=10 thresholdDirection=RISING

```

En el siguiente ejemplo se muestra la salida del subcomando thrsh-job info correspondiente al trabajo de supervisión del umbral creado en el anterior ejemplo:

```
$ mfwkadm thrsh-job info myThreshJob
```

```
Threshold job information for: myThreshJob
```

```
-----
```

```

Type:                SIMPLE
State:               ACTIVE_ON_DUTY
Granularity period:  30
Schedule:
    Global start time: Immediately
    Global stop time: Forever
    Weekly schedule: Everyday
    Daily schedule: All day
Alarm configuration:

```

```

    Type: QualityOfServiceAlarm
    Severity: INDETERMINATE
Threshold definition(s):
    Object: /wsPrefix/com.sun.cmm.ws:name=process-1-threadPool-
NativePool-stats,type=CMM_SWRQueueStats
    Attribute: EntriesCount [LONG]
    Value: 1000
    Direction: RISING
    Offset: 10

```

Estado de salida

Se devuelven los siguientes valores de salida:

- 0 Finalización correcta
- 1 Se ha producido un error

Atributos

Tipo de atributo	Valor del atributo
Disponibilidad	SUNWmfwk
Estabilidad de interfaz	Contrato privado

Consulte también

cacao.5, cacaoadm.1m

Instalación y utilización de Consola de supervisión

Consola de supervisión es la aplicación basada en web que muestra todos los datos de supervisión recopilados por la instrumentación. Se basa en un agente principal para agregar todos los valores y notificaciones de alarmas de cada uno de los agentes de nodo.

Una vez instalado Consola de supervisión, se puede acceder a ella de forma segura mediante una simple ventana de explorador en cualquier host, incluso mediante Internet si el servidor de seguridad está configurado para permitirlo. Utilizando la interfaz gráfica, es posible ver a continuación los valores supervisados en tiempo real, visualizar y aceptar alarmas, y crear reglas para activar alarmas personalizadas.

Nota – Antes de realizar cualquier instalación o configuración, consulte el documento *Notas de la versión Sun Java Enterprise System 5 para UNIX*.

Este capítulo contiene las siguientes secciones:

- “Instalación de Consola de supervisión” en la página 51
- “Diseño del directorio instalado” en la página 54
- “Inicio de la Consola de supervisión” en la página 55
- “Utilización de la Consola de supervisión” en la página 59
- “Solución de problemas de la Consola de supervisión” en la página 68

Instalación de Consola de supervisión

Debido a limitaciones del agente principal de esta versión, no es posible tener un agente principal en el mismo host como agente de nodo. Por tanto, la Consola de supervisión no se puede instalar en el mismo host que cualquiera de los demás componentes supervisados de Java ES. Se debe instalar en su propio host, a menos que haya configurado zonas Solaris. Para obtener más información, consulte a continuación “[Para instalar la Consola de supervisión en una zona Solaris](#)” en la página 53

La instalación de Consola de supervisión instala también Monitoring Framework como dependencia de componente compartido. La consola requiere que la estructura y Common Agent Container carguen el agente principal, pero a diferencia del agente de nodo, el usuario no puede configurar el agente principal. Específicamente, no debe utilizar el comando `mfwkadm` en el host ni en la zona en la que instale Consola de supervisión.

▼ Para instalar la Consola de supervisión con el instalador de Java ES

Debido a una limitación de esta versión beta, debe instalar Consola de supervisión en un host o en una zona Solaris donde no haya otro componente de Java ES instalado. Como resultado, la Consola de supervisión es el único componente que instalará en este procedimiento.

En este procedimiento se utiliza la interfaz gráfica del instalador. Para obtener información sobre cómo ejecutar el instalador en otros modos, consulte el Capítulo 4, “Instalación con la interfaz basada en texto” de *Guía de instalación de Sun Java Enterprise System 5 para UNIX* y el Capítulo 5, “Instalación en el modo silencioso” de *Guía de instalación de Sun Java Enterprise System 5 para UNIX*.

- 1 **Inicie la aplicación `installer` desde el directorio que corresponda a su plataforma de la versión de Java ES. Para obtener más información, consulte la sección “Para comenzar la instalación” de *Guía de instalación de Sun Java Enterprise System 5 para UNIX*.**
- 2 **Después de aparecer la pantalla de bienvenida y de aceptar la licencia, elija Actualizar o Instalar, seleccione Instalar nuevo software y, a continuación, haga clic en Siguiente.**
- 3 **En la pantalla de selección de componentes, seleccione sólo Sun Java System Consola de supervisión para su instalación. Haga clic en Siguiente.**
- 4 **El instalador comprueba qué actualizaciones son necesarias para los componentes compartidos. Una vez realizada la operación, haga clic en Siguiente.**
- 5 **El instalador comprueba ahora los requisitos de sistema. Si el sistema operativo necesita revisiones, cancele la instalación, añada las revisiones necesarias para el sistema y reinicie este procedimiento. En caso contrario, haga clic en Siguiente.**
- 6 **En la pantalla de selección del tipo de configuración, elija Configurar ahora y, en la siguiente pantalla de configuración personalizada, haga clic en Siguiente.**
- 7 **El instalador está preparado para instalar la Consola de supervisión; haga clic en Siguiente para comenzar. Durante la instalación, puede abrir la ventana de registro del producto si aún no ha registrado la implementación de Java ES.**

- 8 Una vez realizada la instalación, puede revisar el resumen de instalación y los registros y, a continuación, hacer clic en La instalación se ha completado para salir del instalador.

Pasos siguientes Ahora debe realizar el procedimiento [“Para configurar la Consola de supervisión” en la página 54.](#)

▼ Para instalar la Consola de supervisión en una zona Solaris

Mediante el uso de zonas Solaris, puede instalar la Consola de supervisión en el mismo host físico que otros componentes de Java ES. Esos componentes estarán en la zona global, y creará una zona local root dispersa como host lógico para la Consola de supervisión. Proceda en el siguiente orden.

- 1 Instale y configure todos los componentes de Java ES en la zona global, excepto la Consola de supervisión. Después de la instalación, configure por completo los componentes seleccionados de la zona global para que se ejecuten todas las instancias del servidor.
- 2 Como parte de la instalación en la zona global, Monitoring Framework se instalará como componente compartido en la zona global. Realice todos los procedimientos indicados en el [Capítulo 2](#) que sean aplicables a los componentes instalados.
- 3 En el mismo host, cree una zona local root dispersa como host lógico para la Consola de supervisión. Al ser una zona root dispersa, Monitoring Framework instalado en *base-de-mfwk* debe ser visible (consulte [“Rutas predeterminadas y nombres de archivo” en la página 9](#)).
- 4 Instale la Consola de supervisión en la zona local root dispersa siguiendo el procedimiento [“Para instalar la Consola de supervisión con el instalador de Java ES” en la página 52](#).
- 5 Configure Monitoring Framework en la zona root dispersa con los siguientes comandos:

```
cd base-de-mfwk/bin
./mfwksetup -i
```

Utilizando los archivos de la zona global, este comando creará los archivos de configuración de Monitoring Framework necesarios en la zona local.

Pasos siguientes Ahora debe realizar el procedimiento [“Para configurar la Consola de supervisión” en la página 54.](#)

▼ Para configurar la Consola de supervisión

Este procedimiento describe cómo configurar la Consola de supervisión en un host físico independiente. Si ha instalado Consola de supervisión en un host lógico creado por una zona Solaris, los comandos serán los mismos, pero se deben ejecutar dentro del sistema de archivos de esa zona.

- 1 **Utilice Monitoring Framework para inicializar el agente principal con los siguientes comandos:**

```
cd base-de-mfwk/bin
./masetup -i
```

- 2 **Reinicie el Common Agent Container (cacao) con el siguiente comando:**

```
cacoadm restart
```

▼ Para desconfigurar la Consola de supervisión

Si instala y configura Consola de supervisión en un host en el que desea instalar otros componentes, no podrá supervisar esos componentes por un conflicto existente en Monitoring Framework. Para supervisar los nuevos componentes con un agente de nodo, debe desconfigurar el agente principal de la Consola de supervisión.

- **Como root, ejecute los siguientes comandos para desconfigurar la Consola de supervisión:**

```
cacoadm stop
cacoadm unregister-module com.sun.mfwk.masteragent.xml
cacoadm register-module /etc/base-de-mfwk/xml/com.sun.mfwk.xml
cacoadm restart
```

Diseño del directorio instalado

Con respecto al nombre del paquete instalado en el sistema operativo, consulte el Capítulo 5, “Listado de paquetes instalables” de *Guía de referencia de instalación de Sun Java Enterprise System para UNIX*. En la siguiente tabla se describen los directorios del paquete de la Consola de supervisión. El directorio de instalación predeterminado *base-de-MConsole* tiene el siguiente significado, como se describe en la sección “[Rutas predeterminadas y nombres de archivo](#)” en la [página 9](#):

- Sistemas Solaris: /opt/SUNWjesmc
- Sistemas Linux: /opt/sun/jesmc

TABLA 3-1 Directorios y archivos que utiliza la Consola de supervisión

Ruta	Descripción del contenido
<i>base-de-MConsole/WEB-INF/classes</i>	Clases de servlet de aplicación web
<i>base-de-MConsole/WEB-INF/lib</i>	Dependencias JAR de aplicación web
<i>base-de-MConsole/WEB-INF/*.xml</i>	Descriptores de aplicación web
<i>base-de-MConsole/css</i>	Archivos de hojas de estilo
<i>base-de-MConsole/html</i>	Archivos HTML
<i>base-de-MConsole/images</i>	Archivos de imagen GIF utilizados en la interfaz de usuario
<i>base-de-MConsole/js</i>	Archivos JavaScript™
<i>base-de-MConsole/*.jsp</i>	Archivos JavaServer Pages™
<i>base-de-WebConsole/prereg/jesmc/*.reg</i>	Archivos de Web Console para la Consola de supervisión

Inicio de la Consola de supervisión

La Consola de supervisión es una aplicación web disponible a través de cualquier explorador que pueda conectarse al host en el que la haya instalado. Accederá a la Consola de supervisión a través de Web Console que se instala automáticamente en el mismo host. En el siguiente procedimiento se describe cómo acceder a la Consola de supervisión y visualizar los componentes supervisados.

▼ Para iniciar la Consola de supervisión

- 1 Debe reiniciar primero el servidor web correspondiente a Web Console. Ejecute este comando en el host o en la zona en que haya instalado la Consola de supervisión:

```
/usr/sbin/smcwebserver restart
```

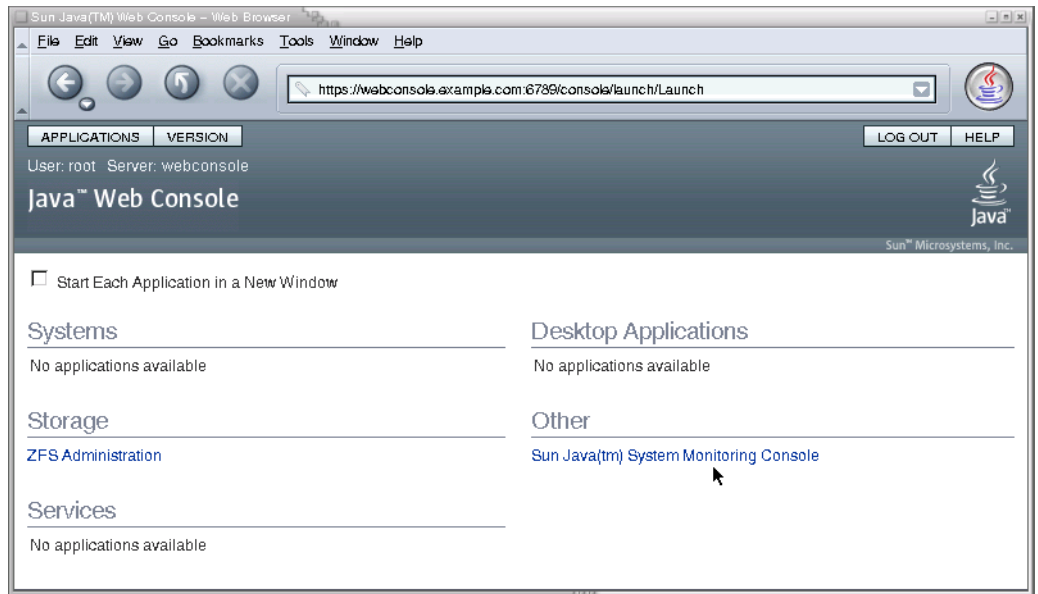
- 2 Espere a que Web Console se inicie. Utilice el siguiente comando para ver si está preparado:

```
/usr/sbin/smcwebserver status
```

Es posible que sea necesario ejecutar este comando varias veces hasta que aparezca el siguiente mensaje:

```
Sun Java(TM) Web Console is running.
```

- 3 Abra Web Console utilizando la siguiente URL desde cualquier explorador que pueda conectarse al host de la Consola de supervisión. Si ha realizado la instalación en una zona Solaris, *host-de-MC* será el nombre de host lógico que dé a esa zona:
`https://host-de-MC.dominio:6789`
- 4 Dependiendo de la configuración de su explorador, es posible que aparezca un mensaje sobre un certificado en el que no se confía. Deberá confiar en el certificado para acceder a Web Console.
- 5 Cuando se le solicite, inicie una sesión en Web Console como `root` utilizando la contraseña `root` en el host de la Consola de supervisión.
 Cuando haya iniciado la sesión, Web Console enumerará todos los servicios que proporciona.
- 6 Para abrir la ventana principal de la Consola de supervisión, haga clic en Sun Java System Consola de supervisión debajo del encabezado "Other" (Otros) como se muestra en la siguiente captura de pantalla.

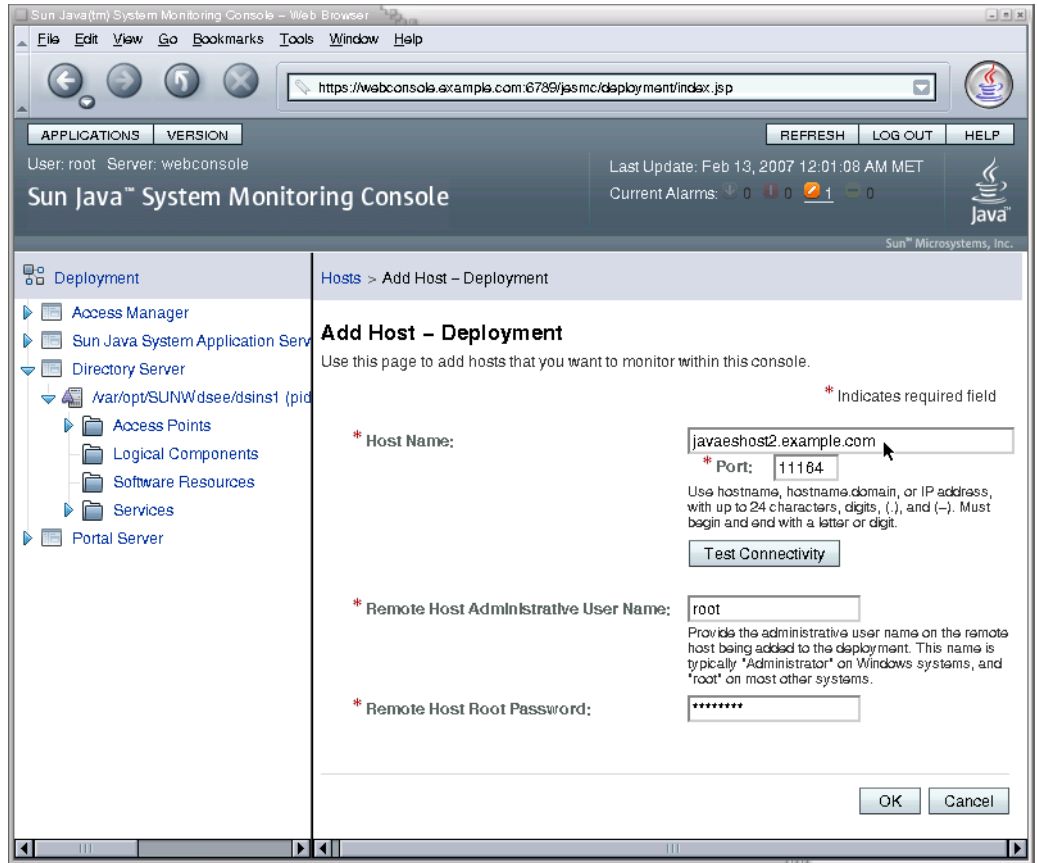


▼ Para conectarse a sus agentes de nodo

Quando la Consola de supervisión se inicia por primera vez, debe indicar dónde se alojan los componentes supervisados. Especifique la ubicación de cada agente de nodo en la implementación de Java ES y la consola mostrará automáticamente todos los componentes existentes en cada agente de nodo. Será necesario también que repita este procedimiento si más adelante añade componentes de Java ES a la implementación instalándolos en nuevos hosts.

Una vez añadido un agente de nodo, la Consola de supervisión volverá a conectarse a él cada vez que acceda a la consola hasta que lo elimine explícitamente. Si un agente de nodo anteriormente añadido no responde, siga el procedimiento [“Para reiniciar un agente de nodo” en la página 30](#).

- 1 Sincronice la fecha y la hora entre el host lógico en el que esté instalada la Consola de supervisión y el host que contenga el agente de nodo y los componentes de Java ES que desea supervisar. Independientemente de si realiza la sincronización de forma automática o manual, no debe haber una diferencia superior a 10 minutos aproximadamente en la hora de cada host.**
- 2 Si es necesario, acceda a la pantalla de nivel de implementación haciendo clic en el vínculo "Deployment" (Implementación) existente en la root de la jerarquía del lado izquierdo de la Consola de supervisión. Seleccione ahora la ficha Hosts del panel derecho y haga clic en Add (Añadir).**
- 3 En el cuadro de diálogo Add Host (Añadir host) que aparece, introduzca la información necesaria como se muestra en la siguiente captura de pantalla:**



- Nombre de host: introduzca el nombre de host totalmente cualificado de un agente de nodo en que haya configurado componentes supervisados.
 - Puerto: 11164, a menos que haya configurado Monitoring Framework en el host en que resida el agente de nodo.
 - Contraseña root del host remoto: introduzca la contraseña root del sistema en que resida el agente de nodo.
- 4 **Haga clic en Test Connectivity (Comprobar conectividad).**
Si la información de conexión es correcta y el agente de host está configurado y en ejecución, el cuadro de diálogo indicará que ya está conectado.
 - 5 **Haga clic en Aceptar para salir del cuadro de diálogo Add Host; el nombre nuevo aparecerá en la lista de hosts. Todos los componentes supervisados del agente de nodo de ese host aparecen ahora también en la columna izquierda.**

- 6 Repita este procedimiento para cada host que haya en la implementación de Java ES en que estén instalados los componentes supervisados.

Pasos siguientes Ahora puede desplazarse a los componentes enumerados en la columna izquierda para ver su estado de funcionamiento, los atributos supervisados que exponen y las alarmas que hayan activado.

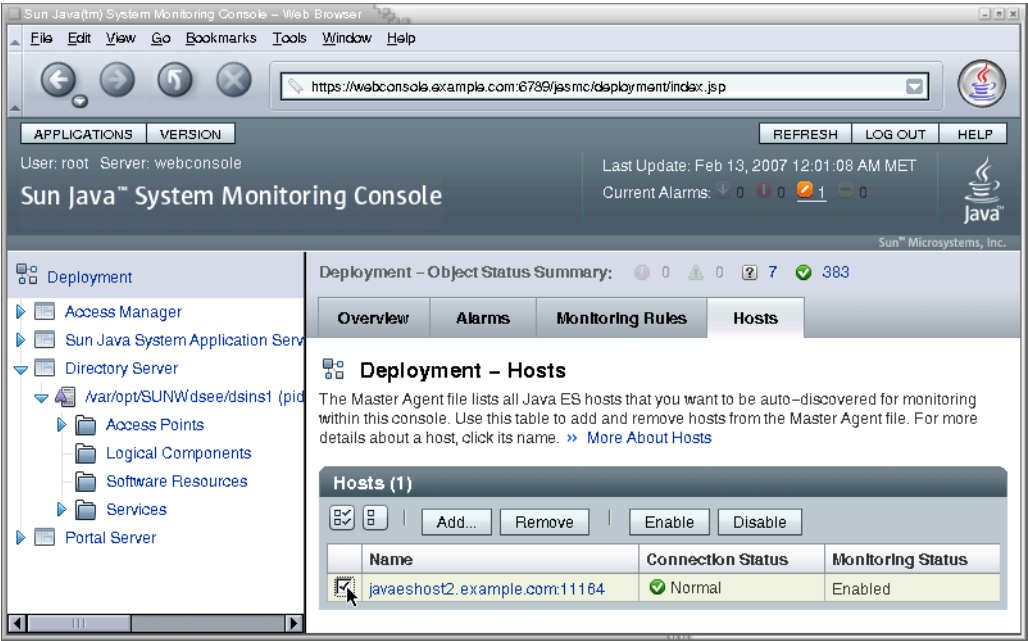
Utilización de la Consola de supervisión

En los procedimientos indicados en esta sección se describe cómo interactuar con la Consola de supervisión.

▼ Para inhabilitar y rehabilitar selectivamente la supervisión

Los mecanismos de supervisión de Java ES están diseñados para ser ligeros con el fin de no afectar al rendimiento de los sistemas de producción. No obstante, en ciertos casos es recomendable dejar de recopilar valores de supervisión para que la instrumentación no afecte apenas al rendimiento. La Consola de supervisión proporciona una forma de realizar esto host por host, como se describe en el siguiente procedimiento.

- 1 Si es necesario, acceda a la pantalla de nivel de implementación haciendo clic en el vínculo "Deployment" (Implementación) existente en la root de la jerarquía del lado izquierdo de la Consola de supervisión. A continuación, haga clic en la ficha Hosts del panel derecho.



En la tabla de la ficha Hosts se enumeran todos los hosts que contienen componentes de Java ES que supervisa la Consola de supervisión.

- 2 Utilice las casillas de verificación de la columna izquierda de la tabla para seleccionar todos los hosts que desee dejar de supervisar. Haga clic en Disable (Inhabilitar) en la parte superior de la tabla de Hosts.

Más información

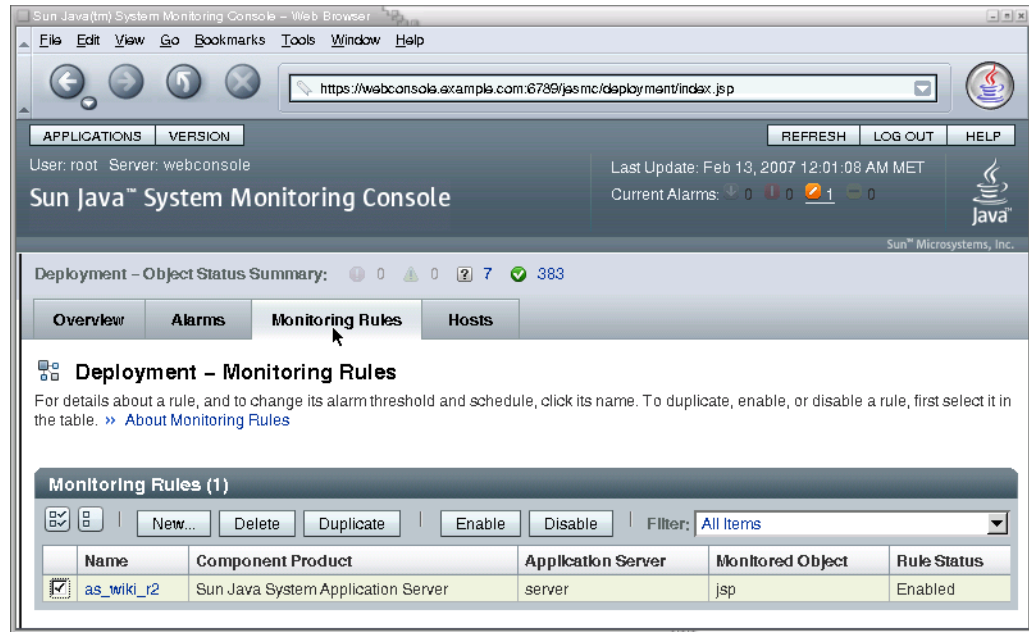
Consecuencias

Al inhabilitar la supervisión en un host, todos los objetos de supervisión de la jerarquía de ese host se inhabilitan. En el estado inhabilitado, los objetos supervisados dejan de actualizarse, aunque sigan conteniendo el último valor. Las reglas de supervisión dependientes de un objeto inhabilitado se suspenden. Para habilitar un host inhabilitado, siga este procedimiento utilizando el botón Enable (Habilitar) de la parte superior de la tabla de Hosts.

▼ **Para crear una regla de supervisión**

Una regla de supervisión, denominada también trabajo de supervisión, es un conjunto de condiciones de valores supervisados que el usuario define para activar una alarma. El asistente de reglas de supervisión de la Consola de supervisión ayuda a definir las condiciones que se desean supervisar.

- 1 Si es necesario, acceda a la pantalla de nivel de implementación haciendo clic en el vínculo "Deployment" (Implementación) existente en la root de la jerarquía del lado izquierdo de la Consola de supervisión. Seleccione ahora la ficha Rules (Reglas) del panel derecho como se muestra en la siguiente captura de pantalla, y haga clic en New (Nuevo) en la tabla de reglas de supervisión:



- 2 Asigne un nombre a la nueva regla de supervisión y elija el tipo de servidor que desea supervisar.

https://webconsole.example.com:6789 - Sun Java(tm) System Monitoring Console - Web Browser

Sun Java™ System Monitoring Console

New Monitoring Rule

Steps Help

Step 1: Define Rule Name and Select Product

Use this wizard to create a rule describing how you want a specific object attribute to be monitored. Set a threshold for that attribute and choose the type of alarm you want generated when the threshold is crossed.

* Indicates required field

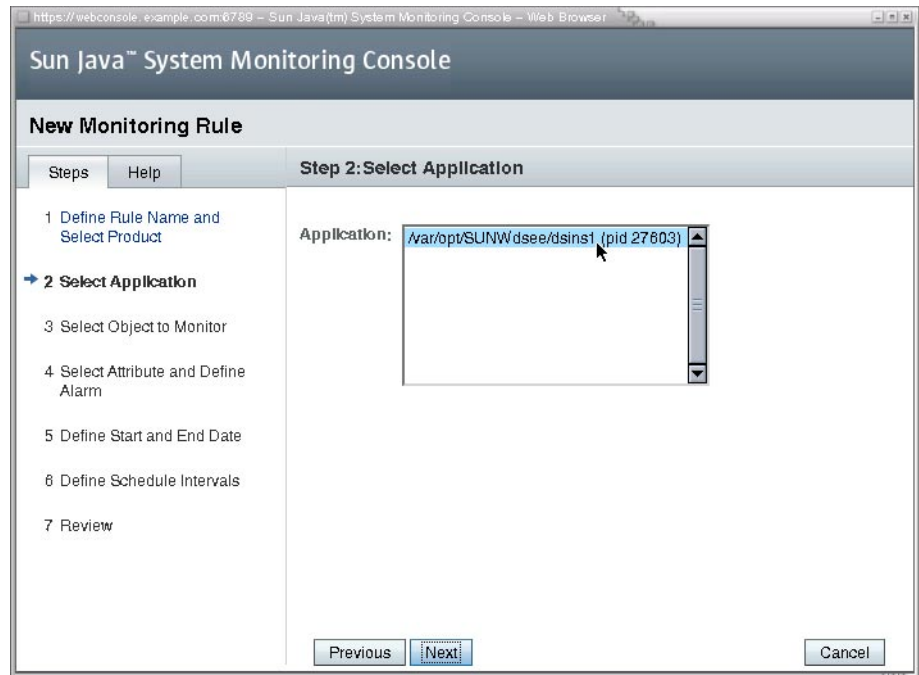
* Rule Name:

Product:

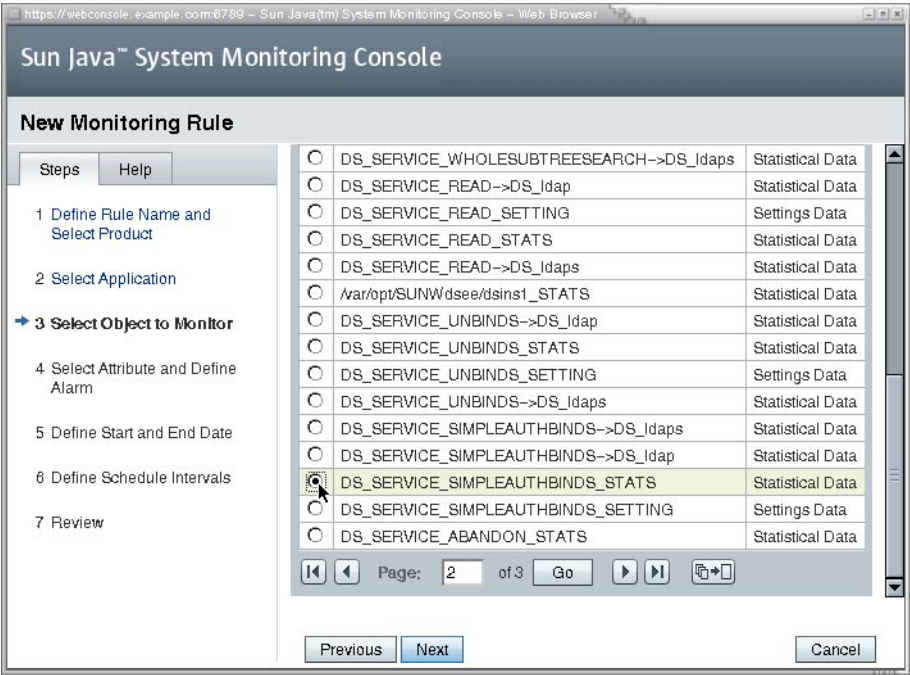
- Access Manager
- Directory Server**
- Portal Server
- Sun Java System Application Server

Previous Next Cancel

- 3 Elija la instancia del producto componente que desea supervisar. Si hay dos instancias del mismo producto instaladas en hosts separados, es posible que algunas instancias tengan nombres idénticos en esta tabla. En este caso, las instancias pueden estar en el mismo orden en que aparecen en la jerarquía del panel izquierdo, pero no hay ninguna manera segura de saberlo. Es posible que necesite crear reglas de supervisión idénticas en ambas instancias para garantizar que la regla se define.



- 4 Seleccione el objeto que contenga el atributo que desea supervisar:



- 5 Ahora, puede especificar finalmente el atributo que se supervisará junto con los valores que generarán una alarma.

The screenshot shows a web browser window titled "Sun Java™ System Monitoring Console - Web Browser" with the URL "https://webconsole.example.com:6789". The main heading is "Sun Java™ System Monitoring Console". Below it is a section titled "New Monitoring Rule" with two tabs: "Steps" and "Help". The "Steps" tab is active, showing a list of seven steps: 1. Define Rule Name and Select Product, 2. Select Application, 3. Select Object to Monitor, 4. Select Attribute and Define Alarm (highlighted with a blue arrow), 5. Define Start and End Date, 6. Define Schedule Intervals, and 7. Review.

The main content area is titled "Step 4: Select Attribute and Define Alarm" and contains the instruction: "Select the attribute to monitor and then provide information about the alarm." A red asterisk indicates required fields. The form includes the following fields:

- Attribute:** A dropdown menu with "InRequests" selected.
- * Threshold:** A text input field containing "10000". Below it, a note says "Use seconds for time, KB for size."
- Direction:** A dropdown menu with "Rising" selected.
- * Offset:** A text input field containing "100". Below it, a note says "Enter a value greater than 0. Use the same measurement units as for Threshold."
- Alarm Severity:** A dropdown menu with "Minor" selected.

At the bottom of the form are three buttons: "Previous", "Next", and "Cancel".

- 6 Introduzca las fechas de inicio y finalización para la regla. A diferencia de la programación que determina la actividad o inactividad de una regla, las fechas de inicio y finalización determinan el periodo de existencia de una regla. Si una hora de inicio está en el pasado, como ocurre siempre con el valor predeterminado, la supervisión asociada a esta regla se iniciará de forma inmediata.

The screenshot shows a web browser window with the URL `https://webconsole.example.com:6789 - Sun Java(tm) System Monitoring Console - Web Browser`. The page title is "Sun Java™ System Monitoring Console". Below the title is a section "New Monitoring Rule" with two tabs: "Steps" and "Help". The "Steps" tab is active, showing a list of seven steps: 1. Define Rule Name and Select Product, 2. Select Application, 3. Select Object to Monitor, 4. Select Attribute and Define Alarm, 5. Define Start and End Date (highlighted with a blue arrow), 6. Define Schedule Intervals, and 7. Review. The main content area is titled "Step 5: Define Start and End Date". It contains several input fields: "Start Date:" with dropdowns for "February", "13", and "2007"; "Start Time:" with dropdowns for "11" and "35", and a note "For 00:00, the start time and date are the same."; "End Date:" with dropdowns for "February", "14", and "2009"; "End Time:" with dropdowns for "00" and "00", and a note "For 00:00, the rule is active through the End Date."; and "* Granularity Period:" with a text input field containing "300" and a note "Time between data collections, in seconds". A legend at the top right states "* Indicates required field". At the bottom are "Previous", "Next", and "Cancel" buttons.

- 7 Opcionalmente, utilice los controles para crear uno o varios rangos de tiempo durante los cuales la regla estará activamente en supervisión. También puede seleccionar los días de la semana para crear una programación semanal.

https://webconsole.example.com:6789 - Sun Java(tm) System Monitoring Console - Web Browser

Sun Java™ System Monitoring Console

New Monitoring Rule

Steps

Help

1 Define Rule Name and Select Product

2 Select Application

3 Select Object to Monitor

4 Select Attribute and Define Alarm

5 Define Start and End Date

→ 6 Define Schedule Intervals

7 Review

Step 6: Define Schedule Intervals

Use this step to define one or more daily and weekly schedules. A daily schedule must be defined before a weekly schedule can be defined.

Daily Schedule Intervals (1)

☒ ☐

Add

Remove

	Start Time	End Time
☐	08 00	20 00

Days of Week: ☒ Mon ☒ Tue ☒ Wed ☒ Thu ☒ Fri ☐ Sat ☐ Sun

Previous

Next

Cancel

- 8 En este paso final del asistente de reglas, revise los datos introducidos y haga clic en Finalizar para crear la nueva regla.



Cuando finalice el asistente de reglas, debe volver a aparecer la ficha Monitoring Rules con la nueva regla incluida en la tabla de reglas.

Solución de problemas de la Consola de supervisión

Consulte también los problemas conocidos indicados en las *Notas de la versión Sun Java Enterprise System 5 para UNIX*.

Si el agente principal crea conflictos con un agente de nodo, compruebe las siguientes condiciones:

- Si utiliza zonas Solaris, compruebe que ha instalado la Consola de supervisión en una zona local root dispersa.
- Compruebe que en el host o en la zona no queda ninguna instalación anterior del componente supervisado.
- En cualquiera de los casos, deberá desinstalar la Consola de supervisión y los componentes, corregir el problema y volver a realizar la instalación de la Consola de supervisión.

Si desinstala la Consola de supervisión y vuelve a instalarla en el mismo host, no se inicializará y no aparecerá en Web Console. En este caso, ejecute el comando `masetup -i` en el host de la Consola de supervisión para inicializar el agente principal. A continuación, siga el procedimiento indicado en la sección [“Para iniciar la Consola de supervisión” en la página 55](#).

Las reglas de supervisión tienen una limitación por la que sólo se pueden inhabilitar cuando están activas. Si desea inhabilitar una regla cuya programación la desactive en el momento actual, deberá cambiar su programación para que la active brevemente, o bien deberá eliminar la regla por completo.

Debido a limitaciones existentes en la plataforma Windows, los valores de `handleCount` y `threadCount` de las estadísticas del host son siempre 0 (cero).

Referencia de objetos CMM

El Common Monitoring Model (CMM) es una ampliación del modelo de información común (CIM) implementado en el lenguaje de programación Java. El CIM está incluido en las interfaces Java de los paquetes `com.sun.cmm.cim.*`. El CMM está incluido en las interfaces de los paquetes `com.sun.cmm.*` que amplían las interfaces CIM. Los objetos supervisados se representan en los agentes de nodo mediante clases que implementan las interfaces CMM. En las siguientes tablas se muestran los atributos que pueden supervisarse para cada clase de objeto.

Descripción general de los objetos CMM

El CMM se basa en un conjunto limitado de interfaces centrales que definen los atributos que puede exponer un objeto supervisado de ese tipo. En la siguiente lista se muestran las clases que representan los tipos generales de objetos supervisados definidos por el CMM, y se proporcionan algunos de sus atributos clave:

<code>CMM_InstalledProduct</code>	Un producto componente Java ES, tomado en conjunto. Por ejemplo, Java ES Directory Server.
<code>CMM_ApplicationSystem</code>	Una instancia instalada y configurada de un producto componente Java ES. Esta instancia puede estar en ejecución o no. Los atributos habituales de este objeto serían la información de contacto del administrador, el estado de funcionamiento del sistema y la hora de inicio o de detención de la aplicación.
<code>CMM_Service</code>	Una función específica de un producto componente, por ejemplo, el servicio de autenticación de Java ES Directory Server. Un atributo habitual sería el estado de funcionamiento del servicio.

CMM_SoftwareResource	Una representación de entidades de software del entorno, tal como un conjunto de subprocesos, antememoria, etc. Un atributo habitual sería el tamaño de una antememoria.
CMM_LogicalComponent	Una entidad que se gestiona mediante un servicio y que el usuario final puede ver, pero que no representa un recurso físico real o una función de software. Por ejemplo, un conjunto de parámetros de configuración para una instancia de software en lugar de la propia instancia.
CMM_ServiceAccessURI	El punto en el que un servicio se encuentra disponible para su uso. Los atributos habituales serían un número de puerto o un identificador de recurso uniforme (URI).
CMM_RemoteServiceAccessPoint	Información de dirección y acceso para una conexión remota. Los atributos habituales serían un URI o el estado de funcionamiento de la conexión (abierta o cerrada).
CMM_Process	Una única instancia de un programa que se está ejecutando. Los atributos habituales serían la utilización de la memoria o de la CPU.
CMM_UnitaryComputerSystem	Un único host que utiliza la implementación de Java ES, por ejemplo, un equipo de escritorio o un servidor. Los atributos habituales podrían ser el número de procesadores disponibles o la cantidad de memoria física.
CMM_OperatingSystem	El software o firmware que permite utilizar el hardware de un equipo host. Un atributo habitual podría ser la cantidad de memoria virtual disponible en el sistema.
CMM_JVM	Java Virtual Machine que un servidor Java ES utiliza. Un atributo de ejemplo podría ser el número de versión de Java Virtual Machine.
CMM_DatabaseService	Una tarea realizada en nombre de una base de datos, por ejemplo, proporcionar acceso de usuario. Un atributo habitual podría ser el número máximo permitido de conexiones con la base de datos.
CMM_CommonDatabase	Propiedades comunes en un determinado tipo de base de datos. Un atributo habitual podría ser la fecha de la copia de seguridad más reciente.

Objetos supervisados expuestos por cada componente

En las secciones de este apéndice se enumeran los objetos CMM que se han instrumentado en cada componente del producto compatible con la supervisión. Cuando sólo se instrumenta un subconjunto de atributos de un objeto, también se enumeran esos atributos.

Instrumentación de Common Agent Container

No se ha documentado aún.

Instrumentación de Access Manager

No se ha documentado aún.

Instrumentación de Application Server

No se ha documentado aún.

Instrumentación de Calendar Server

No se ha documentado aún.

Instrumentación de Directory Server

No se ha documentado aún.

Instrumentación de Instant Messaging

No se ha documentado aún.

Instrumentación de Messaging Server

No se ha documentado aún.

Instrumentación de Portal Server

No se ha documentado aún.

Instrumentación de Web Server

No se ha documentado aún.

Índice

G

glosario, vínculo al, 9

