

Cómo encender el servidor con el debugging del jvm, pero de la daemon del monitor.

Contenido

[Introducción](#)

Q. [Cómo encender el servidor con el debugging del jvm, pero de la daemon del monitor.](#)

Introducción

Este documento describe a los modos de operación especiales de servicios CIS.

Q. Cómo encender el servidor con el debugging del jvm, pero de la daemon del monitor.

En algunos casos limitados y solamente en la recomendación del soporte de Cisco DV, usted puede necesitar funcionar con el servidor en un modo del debug del jvm durante un largo período de tiempo. Este modo determinado del debug es el donde el servidor se enciende de este comando:

```
# debug de ./composite_server.sh
```

Esto enciende el servidor de información de Cisco como proceso del primero plano en el modo del debug (port= que escucha 8000 del debug predeterminado). El proceso de daemon del monitor de Cisco no se comienza en este caso. Una vez en este modo, el soporte de Cisco querrá asociar un depurador a su puerto 8000.

Nota: Este modo del debug no es lo mismo que el tipo más convencional de seguimiento de debug, donde le piden habilitar los diversos niveles de debug del archivo de `/conf/server/log4j.properties`, o habilitar el debugging de la configuración UI del estudio. Ejecute solamente esto cuando es dado instrucciones por el soporte de Cisco.

Un problema a considerar en este caso:

Al encender el servidor como esto, el proceso del servidor se está ejecutando en el primero plano y podría ser terminado si usted cierra accidentalmente la sesión o cancela el proceso con un CNTL-C. Si usted necesita dejar el servidor que se ejecuta en este modo por los períodos ampliados, puede ser aventurado dejar su sesión abierta. También, sin el funcionamiento de la daemon del monitor, no estará presente recomenzar su servidor automáticamente, si para por cualquier motivo.

Instrucciones de evitar el modo del primero plano y de encender el servidor normalmente:

1) en `/conf/server/server.properties`, edita esta línea (depende del OS que usted está utilizando. aquí, el linux se utiliza como un ejemplo y está en **intrépido**)

```
linux.vmargs=" - servidor -XX:NewRatio=6 - XX: - UseGCOverheadLimit - XX:  
+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/opt/Cisco_Systems/CIS_7.0.3_brad/logs  
-XX:PermSize=64m -XX:MaxPermSize=256m - XX: - ReduceInitialCardMarks - XX:  
+ExplicitGCInvokesConcurrent - XX: +UseConcMarkSweepGC"
```

2) Insertan esta línea (mostrada en intrépido)

```
>- Xdebug - Xnoagent - Djava.compiler=NONE -  
Xrunjdpw:transport=dt_socket,server=y,suspend=n,address=8000
```

en los vmargs.

En este ejemplo, se inserta momentos antes de la doble-cita terminal.

```
linux.vmargs=" - servidor -XX:NewRatio=6 - XX: - UseGCOverheadLimit - XX:  
+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/opt/Cisco_Systems/CIS_7.0.3_brad/logs  
-XX:PermSize=64m -XX:MaxPermSize=256m - XX: - ReduceInitialCardMarks - XX:  
+ExplicitGCInvokesConcurrent - XX: +UseConcMarkSweepGC - Xdebug-Xnoagent -  
Djava.compiler=NONE -Xrunjdpw:transport=dt_socket,server=y,suspend=n,address=8000
```

Esto ahora permite el servidor se ejecute en el modo del debug y haga que también escucha en el puerto 8000 nuestra herramienta de debug que usted utilice más adelante.

El resultado del servidor será lo mismo que cuando usted ejecuta el **debug de /bin/composite_server.sh**, a menos que ahora, con esta modificación server.properties, usted pueda comenzar los servicios normalmente invocando nuestra daemon del monitor, en vez de funcionar con el **archivo de composite_server.sh**.

3) enciende el servidor normalmente, como este **comienzo del monitor /bin/composite.sh**

4) verifica el servidor se está ejecutando (puede iniciar sesión con el estudio para probar)

5) permite que el soporte de Cisco utilice una herramienta de debugging para asociar a su servidor.