

Como ligar o server com eliminação de erros do jvm, mas do demônio do monitor.

Índice

[Introdução](#)

Q. [Como ligar o server com eliminação de erros do jvm, mas do demônio do monitor.](#)

Introdução

Este documento descreve modos operacionais especiais de serviços CI.

Q. Como ligar o server com eliminação de erros do jvm, mas do demônio do monitor.

Em alguns casos limitados e somente na recomendação do apoio de Cisco DV, você pode precisar de executar o server em um jvm debuga o modo por um período de tempo prolongado. Este detalhe debuga o modo é esse onde o server é ligado deste comando:

```
# ./composite_server.sh debuga
```

Isto liga o server de informação de Cisco como um processo do primeiro plano dentro debugar o modo (o padrão debuga o port= de escuta 8000). Cisco monitora o processo de demônio não é começado neste caso. Uma vez neste modo, Cisco apoia quererá anexar um depurador de programas a sua porta 8000.

Nota: Isto debuga o modo não é o mesmo que o tipo mais convencional de debug tracing, onde você é pedido para permitir vários níveis de debug do arquivo de **/conf/server/log4j.properties**, ou para permitir a eliminação de erros da configuração UI do estúdio. Execute somente isto quando instruído pelo apoio de Cisco.

Uma edição a considerar neste caso:

Ao ligar o server como isto, o processo de servidor está sendo executado no primeiro plano e poderia ser terminado se você fecha acidentalmente a sessão ou cancela o processo com um CNTL-C. Se você precisa de deixar o server que é executado neste modo por períodos estendido, pode ser arriscado deixar sua sessão aberta. Também, sem o corredor do demônio do monitor, não esta presente para reiniciar automaticamente seu server, se para por qualquer razão.

Instruções para evitar o modo do primeiro plano e para ligar normalmente o server:

1) em **/conf/server/server.properties**, edita esta linha (depende do ósmio que você se está usando. aqui, o linux é usado como um exemplo e está em **corajoso**)

```
linux.vmargs=" - server -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) Introduzem esta linha (mostrada em corajoso)

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

nos vmargs.

Neste exemplo, é introduzido imediatamente antes das dobro-citações terminais.

```
linux.vmargs=” - server -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
```

Isto permite agora que o server rode-se debuga o modo e fá-lo igualmente escutar na porta 8000 nossa ferramenta para debug que você usará mais tarde.

O resultado do server será o mesmo que quando você executa **/bin/composite_server.sh debuga**, a não ser que agora, com esta alteração server.properties, você possa enfiar os serviços normalmente invocando nosso demônio do monitor, em vez de executar o **arquivo de composite_server.sh**.

3) Ligue o server normalmente, como este **começo do monitor** /bin/composite.sh

4) Verifique que o server está sendo executado (pode entrar com estúdio para testar)

5) Permita que o apoio de Cisco use uma ferramenta para debug para anexar a seu server.