

Dépannage de mémoire CVP avec Jconsole

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Utilitaire de Jconsole](#)

Introduction

Ce document décrit comment utiliser l'outil de console Java (jconsole) pour dépanner des problèmes de fuite de mémoire du Portail Cisco Unified Customer Voice (CVP).

Conditions préalables

Exigences

Cisco recommande que vous ayez la connaissance de ces thèmes

- Portail Cisco Unified Customer Voice (CVP)
- Utilitaire de console Java

[Composants utilisés](#)

Les informations dans ce document sont basées sur la version 9.0(X) et ultérieures CVP.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

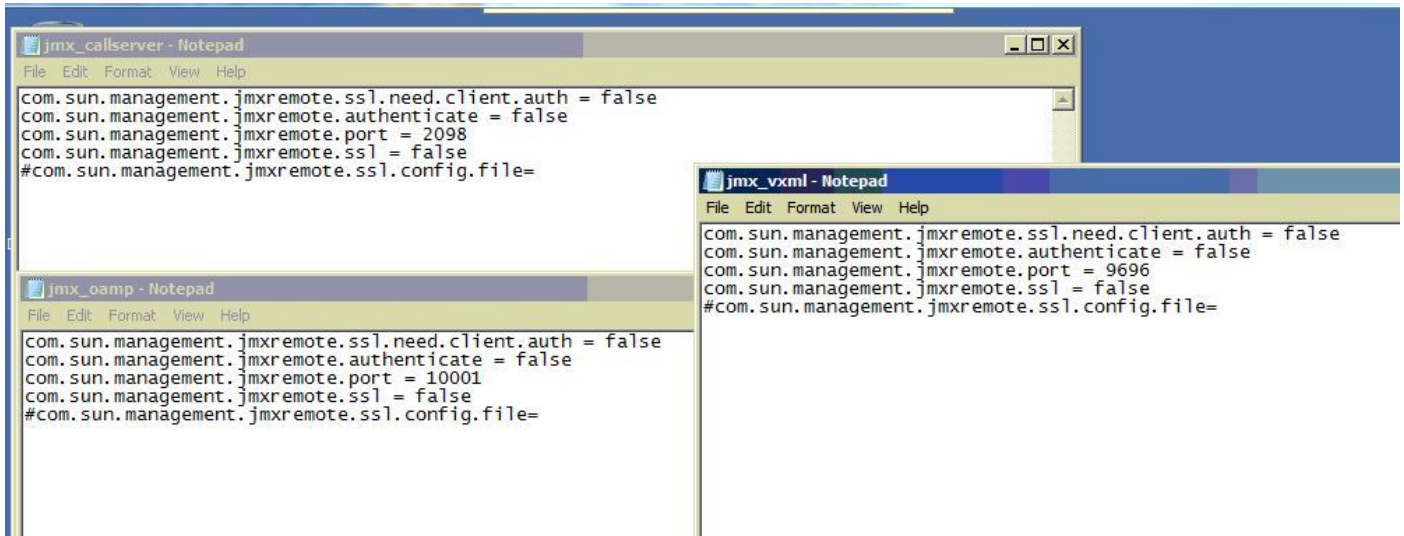
Utilitaire de Jconsole

Pour dépannage du serveur d'appel, fuite de mémoire ou problèmes relatifs aux performances de serveur VXML souvent il est nécessaire d'indiquer le suivi de vidage mémoire de segment de mémoire de Javas dans l'utilitaire de jconsole. Ceci est habituellement fait après le rétrécissement en bas d'un problème de ressource à un service spécifique ou les services dans le serveur par l'intermédiaire du visualisateur d'événements, du gestionnaire de tâches, et/ou des outils de logs de perfmon. Le segment de mémoire de vidage mémoire d'utilitaires est un fichier de suivi inférieur et est recommandé d'être tourné sur CVP dépannant le niveau de suivi.

Jconsole par défaut se trouve dans le chemin « C:\Cisco\CVP\jre\bin » des serveurs de serveur, VXML, et OAMP d'appel et pour ouvrir une session dans les fichiers de configuration de console que vous devez d'abord configurer les détails de port dans jmx_callserver.conf, jmx_vxml.conf, et

jmx_oamp.conf de ces serveurs respectivement

- Port 2098 du serveur JMX d'appel
- Port 9696 du serveur JMX VXML
- Port 10001 du serveur JMX OAMP



The image shows three Notepad windows displaying configuration files for JMX services. The windows are titled 'jmx_callserver - Notepad', 'jmx_oamp - Notepad', and 'jmx_vxml - Notepad'. Each window contains the following configuration lines:

```
com.sun.management.jmxremote.ssl.need.client.auth = false
com.sun.management.jmxremote.authenticate = false
com.sun.management.jmxremote.port = 2098
com.sun.management.jmxremote.ssl = false
#com.sun.management.jmxremote.ssl.config.file=
```

```
com.sun.management.jmxremote.ssl.need.client.auth = false
com.sun.management.jmxremote.authenticate = false
com.sun.management.jmxremote.port = 10001
com.sun.management.jmxremote.ssl = false
#com.sun.management.jmxremote.ssl.config.file=
```

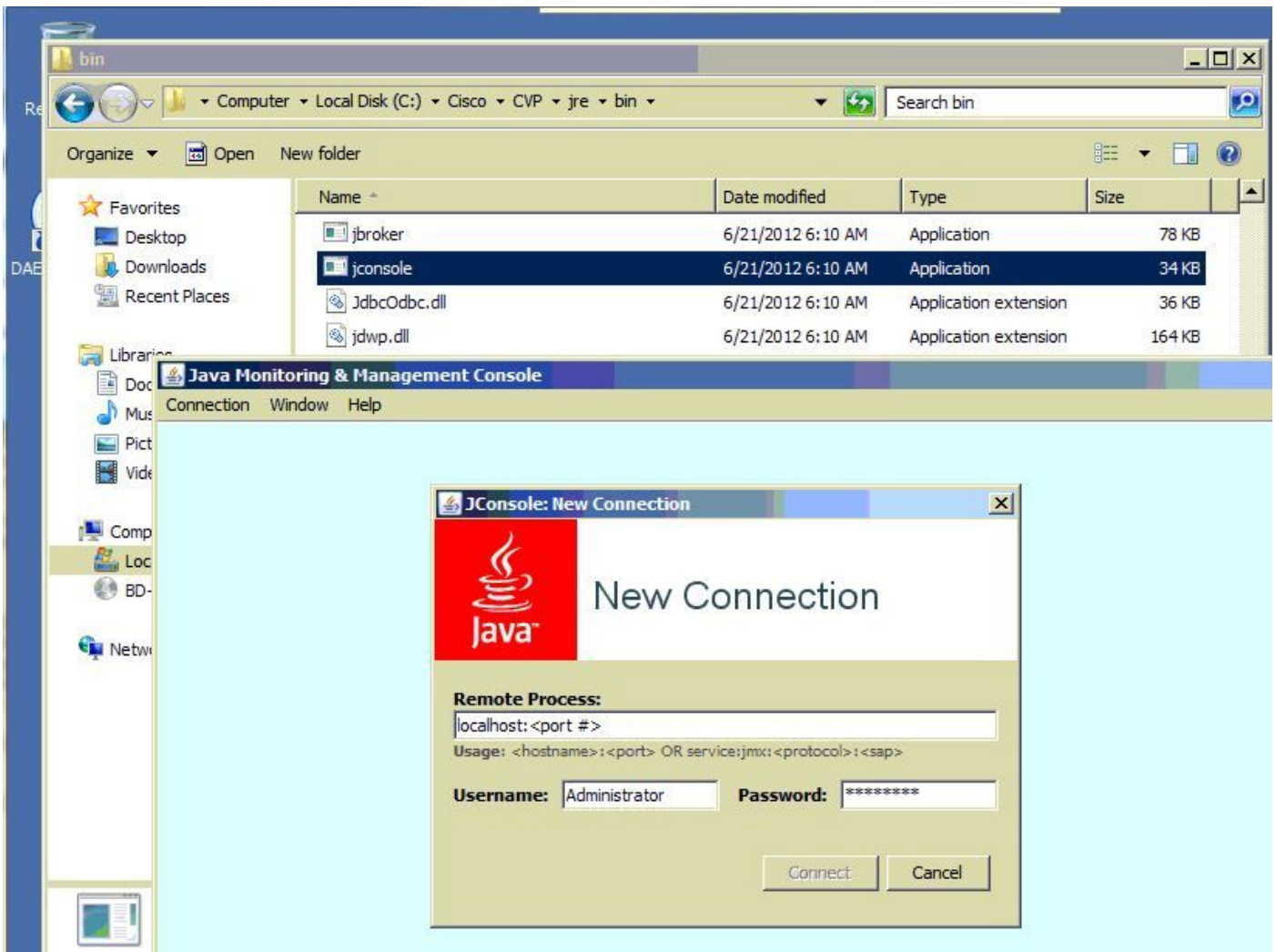
```
com.sun.management.jmxremote.ssl.need.client.auth = false
com.sun.management.jmxremote.authenticate = false
com.sun.management.jmxremote.port = 9696
com.sun.management.jmxremote.ssl = false
#com.sun.management.jmxremote.ssl.config.file=
```

Après que le fichier de configuration soit mis à jour, exécutez le jconsole de service comme expliqué dans ces étapes

Step1. Ouvrez %CVP_HOME%/CVP/jre/bin/jconsole.ex3.

Step2. Connectez au localhost, et spécifiez le numéro de port correspondant au port du serveur JMX de Callserver, VXML ou OAMP.

Step3. Ouvrez une session avec le laisser-passer de procédure de connexion d'oamp CVP ou le laisser-passer d'admin.



Step4. Onglet ouvert de MBeans.

Step5. Ouvrez com.sun.management - HotSpotDiagnostics - Exécutions - dumpheap.

Step6. Dans p0 entrez dans un nom du fichier et un chemin vers lesquels vous voulez diriger le vidage mémoire. Congé p1 comme vrai. Cliquez sur en fonction le bouton de dumpHeap et collectez le fichier généré.

- + Catalina
 - + JMImplementation
 - + Users
 - + com.sun.management
 - HotSpotDiagnostic
 - + Attributes
 - Operations
 - dumpHeap**
 - getVMOption
 - setVMOption
 - + java.lang
 - + java.util.logging

Operation invocation

```
void dumpHeap ( p0  , p1  )
```

MBeanOperationInfo

Name	Value
Operation:	
Name	dumpHeap
Description	dumpHeap
Impact	UNKNOWN
ReturnType	void
Parameter-0:	
Name	p0
Description	p0
Type	java.lang.String
Parameter-1:	
Name	p1
Description	p1
Type	boolean

Descriptor

Name	Value
Operation:	
openType	javax.management.openmbean.SimpleType(name=java.lang.Void)
originalType	void
Parameter-0:	
openType	javax.management.openmbean.SimpleType(name=java.lang.String)
originalType	java.lang.String
Parameter-1:	
openType	javax.management.openmbean.SimpleType(name=java.lang.Boolean)
originalType	boolean