

Troubleshooting da memória CVP com Jconsole

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Utilidade de Jconsole](#)

Introdução

Este documento descreve como usar a ferramenta do console de Java (jconsole) para pesquisar defeitos questões de escape de memória do Portal Cisco Unified Customer Voice (CVP).

Pré-requisitos

Requisitos

Cisco recomenda que você tem o conhecimento destes assuntos

- Portal Cisco Unified Customer Voice (CVP)
- Utilidade do console de Java

Componentes Utilizados

A informação neste documento é baseada na versão 9.0(X) e mais recente CVP.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

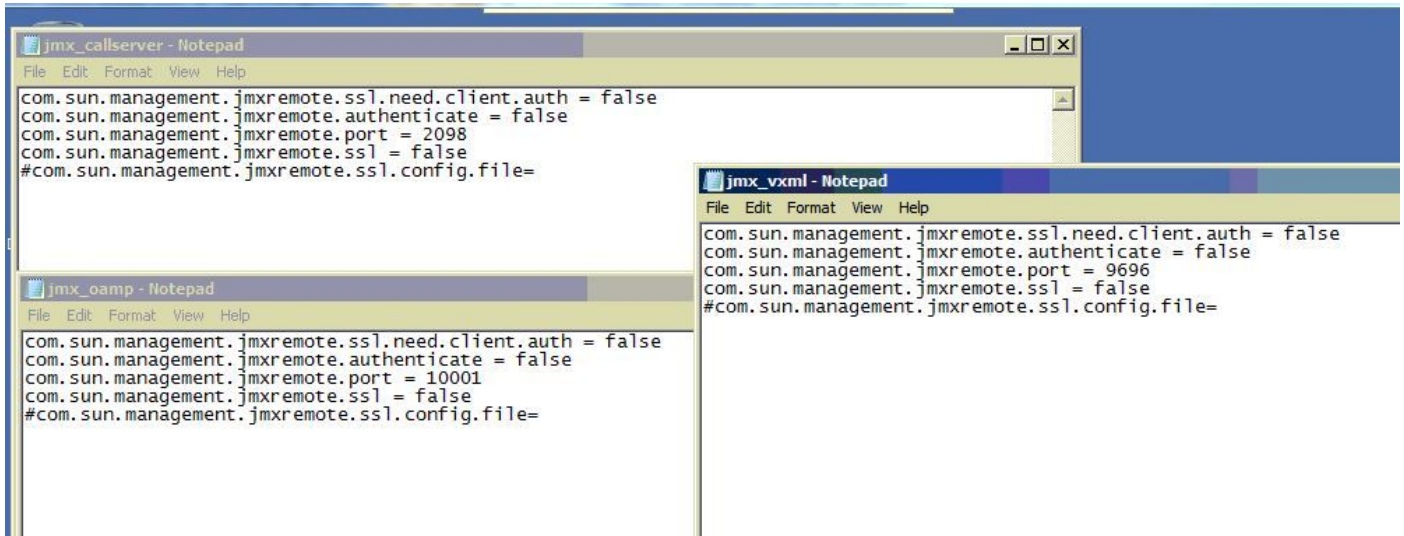
Utilidade de Jconsole

Para pesquisar defeitos o server do atendimento, escape de memória ou questões relacionadas ao desempenho do server VXML frequentemente é necessário girar acima do traço da descarga do montão das Javas na utilidade do jconsole. Isto é feito geralmente após o redução abaixo de um problema de recurso a um serviço ou a uns serviços específicos no server através do visualizador de eventos, do gerenciador de tarefa, e/ou das ferramentas dos logs do perfmon. O montão da descarga das utilidades é um arquivo de rastreamento de baixo nível e é recomendado ser girado acima sobre o CVP pesquisa defeitos o nível de rastreamento.

Jconsole é ficado situado à revelia no trajeto "C:\Cisco\CVP\jre\bin" de server do server, VXML, e OAMP do atendimento e para entrar nos arquivos de configuração do console que você deve primeiramente configurar os detalhes de porta em jmx_callserver.conf, jmx_vxml.conf, e

jmx_oamp.conf destes server respectivamente

- Porta 2098 do server JMX do atendimento
- Porta 9696 do server JMX VXML
- Porta 10001 do server 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 text:

```
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=
```

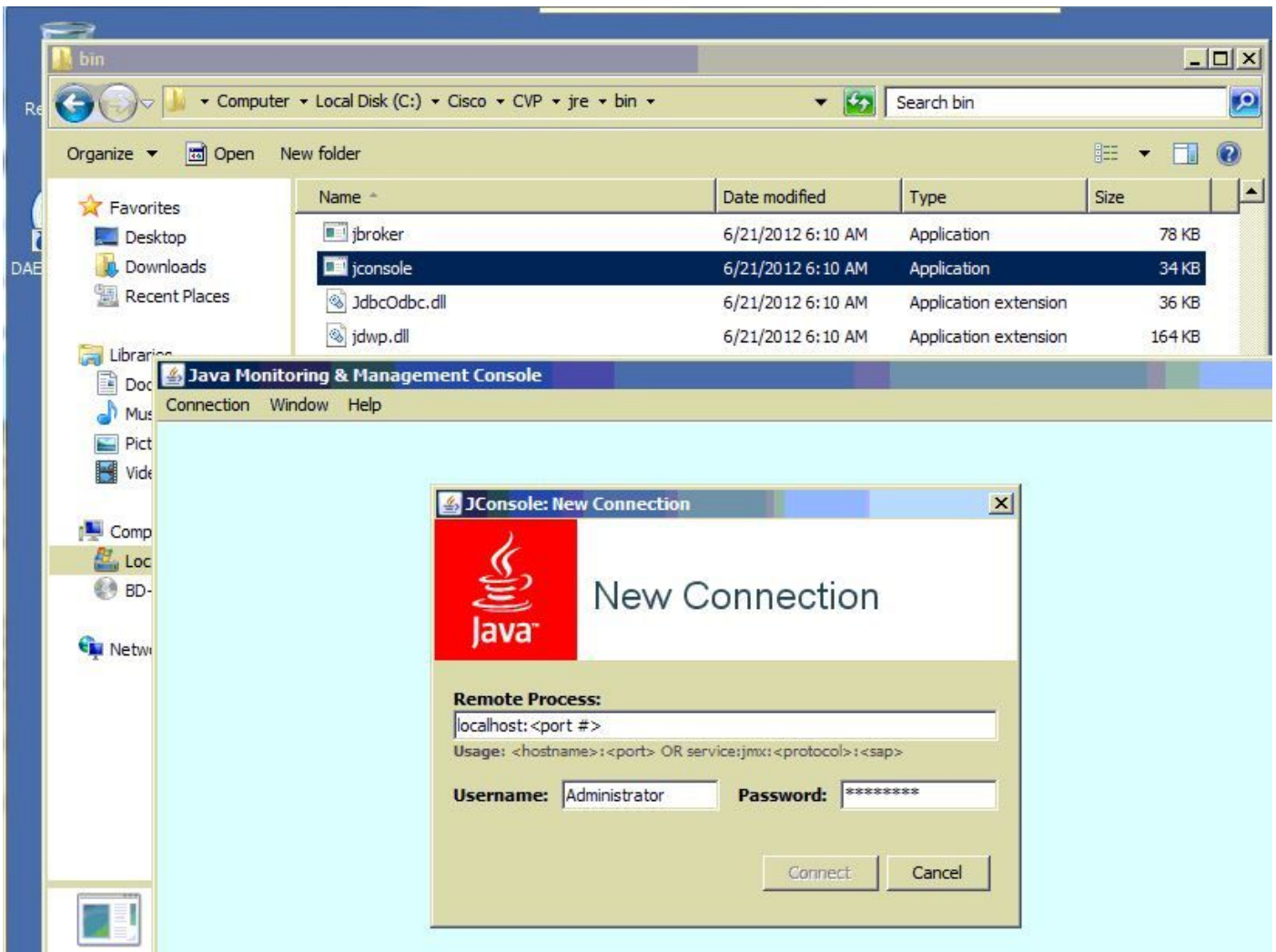
```
com.sun.management.jmxremote.ssl.need.client.auth = false
com.sun.management.jmxremote.authenticate = false
com.sun.management.jmxremote.port = 9696
com.sun.management.jmxremote.ssl = false
#com.sun.management.jmxremote.ssl.config.file=
```

Depois que o arquivo de configuração é atualizado, execute o jconsole de serviço público como explicado nestas etapas

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

Step2. Conecte ao host local, e especifique o número de porta que corresponde à porta do server JMX de Callserver, VXML ou OAMP.

Step3. Entre com as credenciais do início de uma sessão do oamp CVP ou as credenciais Admin.



Step4. Abra a aba de MBeans.

Step5. Abra com.sun.management - HotSpotDiagnostics - Operações - dumpheap.

Step6. Em p0 entre em um nome de arquivo e em um trajeto a que você quer dirigir a descarga. Deixe p1 como verdadeiro. Clique sobre o botão do dumpHeap e recolha o arquivo gerado.

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

Operation invocation

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

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