

# Probleemoplossing bij CVP VXML-servergeheugen met Jconsole

## Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Jconsole-hulpprogramma](#)

## Inleiding

In dit document wordt beschreven hoe u Java Console (console)-gereedschap kunt gebruiken om problemen met uw geheugen op te lossen bij Cisco Unified Customer Voice Portal (CVP).

## Voorwaarden

### Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Cisco Unified Customer Voice Portal (CVP)
- Java-console

### Gebruikte componenten

De informatie in dit document is gebaseerd op CVP versie 12.5.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk levend is, zorg er dan voor dat u de mogelijke impact van om het even welke opdracht begrijpt.

## Jconsole-hulpprogramma

Voor problemen met de Call Server van de Probleemoplossing is het lekken van het geheugen van de VXML server of prestatiegerelateerde kwesties vaak nodig om java heap dummy in jconsole voorziening op te zetten. Dit wordt meestal gedaan nadat u een resource probleem hebt verkleind naar een specifieke service of services in de server via Windows Event Viewer, Automation Manager en/of doorlopende loggereedschappen. De THEE-software voor nutsbedrijven is een sporenbestand van laag niveau en wordt aanbevolen om bovenop het Tracleer-niveau van de CVP-probleemoplossing te worden geactiveerd.

De standaardinstelling is dat Jconsole zich in het pad **C:\Cisco\CVP\jre\bin** van de CVP Server

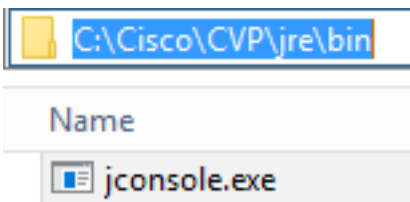
bevindt. De poortgegevens zijn al standaard ingesteld in **jmx\_callserver.conf**, **jmx\_vxml.conf**, **jmx\_oamp.conf** en **jmx\_wsm.conf** configuratiebestanden in **C:\Cisco\CVP\conf** voor elke respectievelijke CVP-server.

- JMX-poort 2098 voor Call Server
- VXML Server JMX poort 9696
- JMX-poort op OAMP Server 1001
- WSM JMX poort 2099

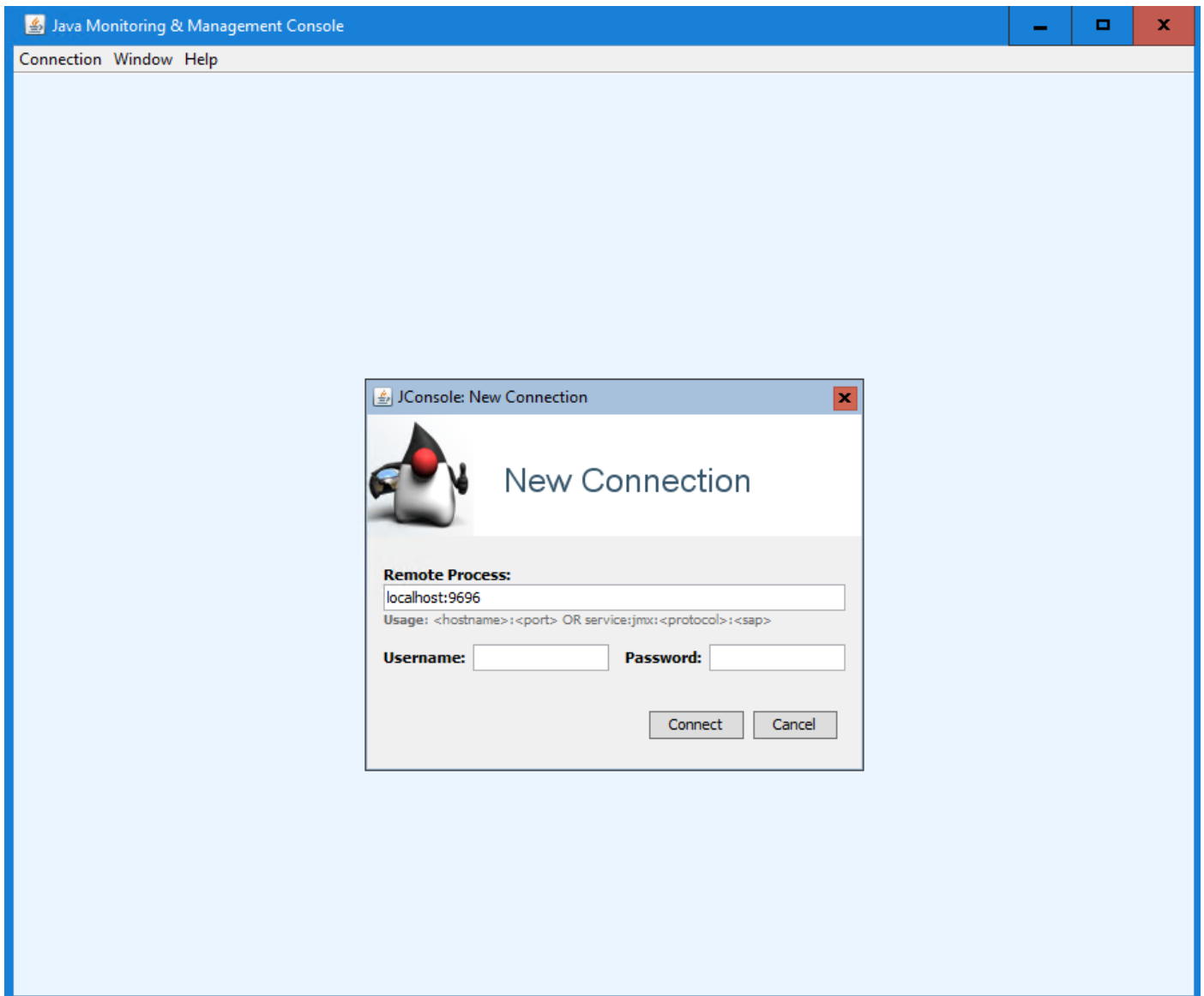
<pre>jmx_callserver.conf - Notepad File Edit Format View Help com.sun.management.jmxremote.ssl.need.client.auth = false com.sun.management.jmxremote.authenticate = false com.sun.management.jmxremote.port = 2098 com.sun.management.jmxremote.ssl = false #com.sun.management.jmxremote.ssl.config.file= com.sun.management.jmxremote.rmi.port = 2097</pre>	<pre>jmx_vxml.conf - Notepad File Edit Format View Help 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= com.sun.management.jmxremote.rmi.port = 9697</pre>
<pre>jmx_oamp.conf - Notepad File Edit Format View Help 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.rmi.port = 10000</pre>	<pre>jmx_wsm.conf - Notepad File Edit Format View Help com.sun.management.jmxremote.ssl.need.client.auth = false com.sun.management.jmxremote.authenticate = false com.sun.management.jmxremote.port = 2099 com.sun.management.jmxremote.ssl = false #com.sun.management.jmxremote.ssl.config.file= com.sun.management.jmxremote.rmi.port = 3000</pre>

U kunt de Jconsole voorziening zoals uitgelegd in deze stappen uitvoeren:

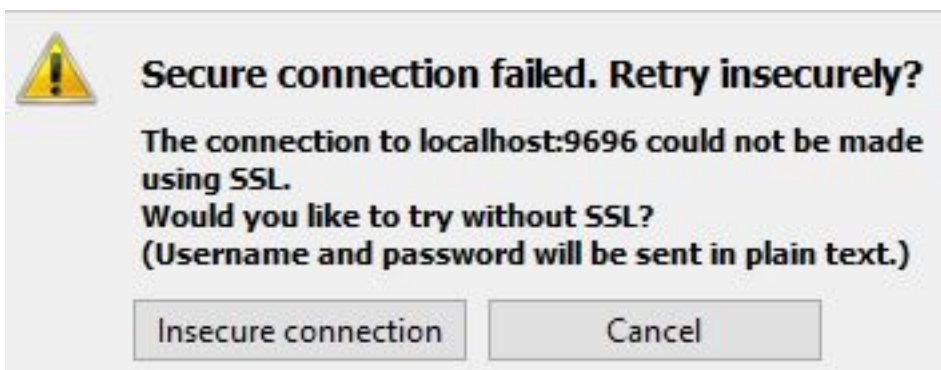
Stap 1. Navigeer naar **%CVP\_HOME%\CVP\jre\bin\jconsole.exe** en dubbelklik op **jconsole.exe**.



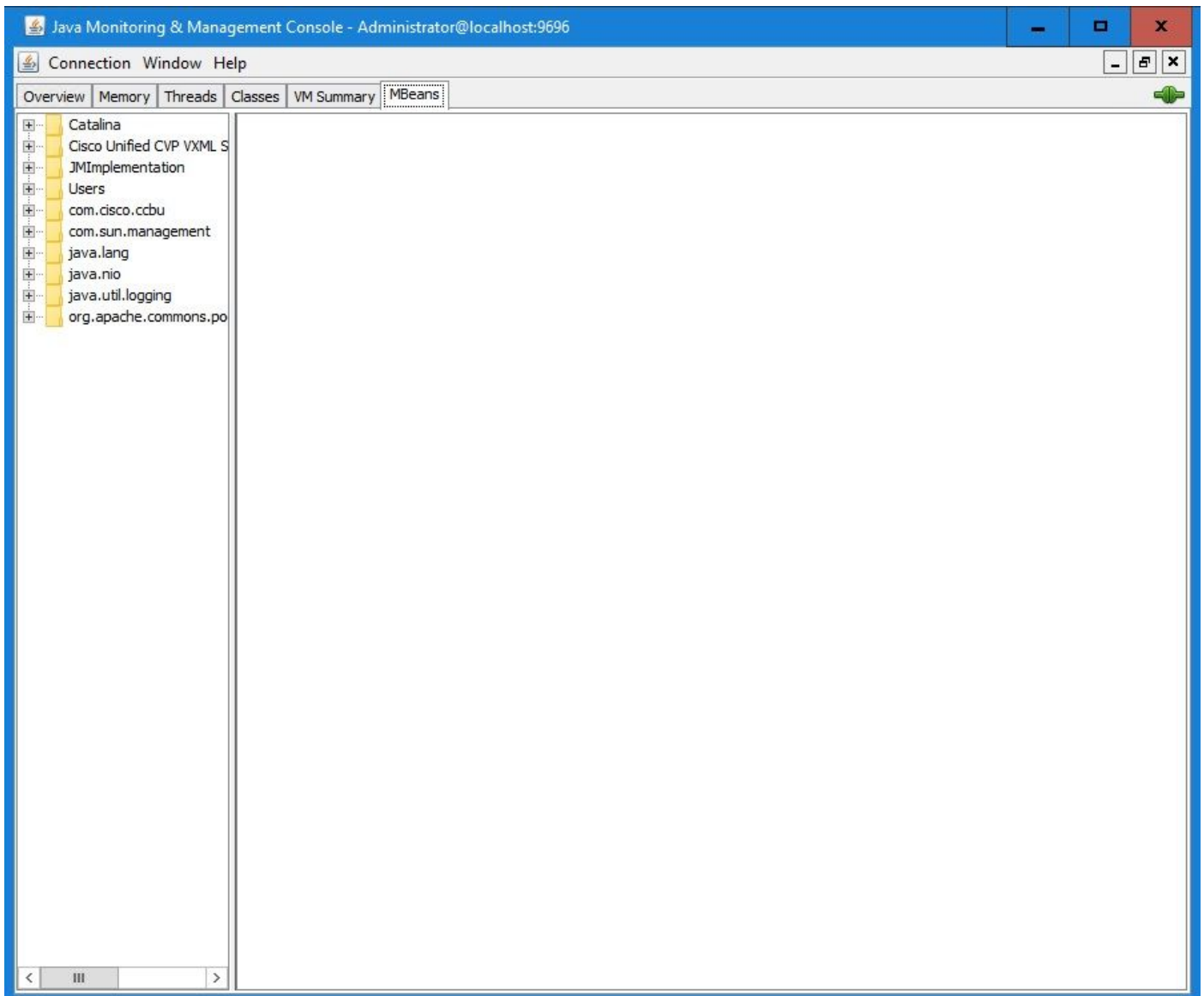
Stap 2. Sluit aan op **localhost** en specificeer het poortnummer voor de CVP-component waarmee u verbinding wilt maken, bijvoorbeeld voor CVP VXML Server gebruiken we JMX poort 9696. Laat de **gebruikersnaam** en het **wachtwoord** leeg. Klik op **Connect**.



Stap 3. Klik op **Onveilige verbinding**.



Stap 4. Selecteer het tabblad **MB**.



Stap 5. Vergroot `com.sun.management` > `HotSpotDiagnostic` > `Operations` en klik op `dumpenHead`.

Java Monitoring & Management Console - Administrator@localhost:9696

Connection Window Help

Overview Memory Threads Classes VM Summary MBeans

Catalina  
 Cisco Unified CVP VXML S  
 JImplementation  
 Users  
 com.cisco.ccbu  
 com.sun.management  
 DiagnosticCommand  
 HotSpotDiagnostic  
 Attributes  
 Operations  
 dumpHeap  
 getVMOption  
 setVMOption  
 java.lang  
 java.nio  
 java.util.logging  
 org.apache.commons.po

Operation invocation  
 void dumpHeap ( p0 String , p1 true )

MBeanOperationInfo

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

Descriptor

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

Stap 6. Voer in **p0** de bestandsnaam in voor het stort met de bestandsextensie **.hprof**, bijvoorbeeld **vxmiDump.hprof**. Laat **p1** als **waar**.

The screenshot shows the Java Monitoring & Management Console interface. The left-hand tree view shows the hierarchy of MBeans, with 'dumpHeap' selected under 'HotSpotDiagnostic'. The main panel displays the 'MBeanOperationInfo' for the 'dumpHeap' operation, including its name, description, impact, return type, and parameters. The 'Operation invocation' section shows the command: `void dumpHeap ( p0 vxmlDump.hprof , p1 true )`. The 'Descriptor' section provides details for each parameter, including their original types.

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

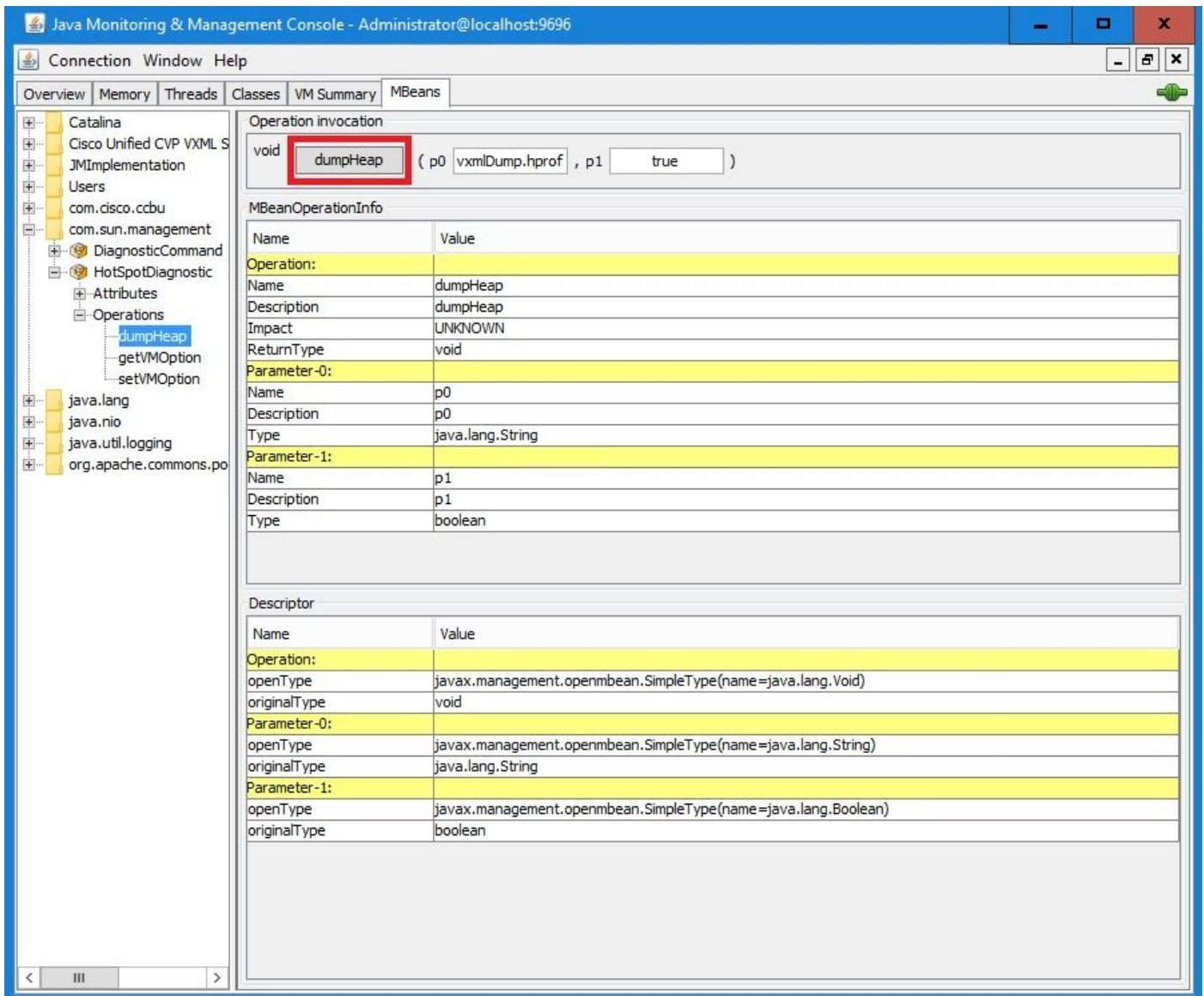
  

Name	Value
<b>Operation:</b>	
openType	javax.management.openmbean.SimpleType(name=java.lang.Void)
originalType	void
<b>Parameter-0:</b>	
openType	javax.management.openmbean.SimpleType(name=java.lang.String)
originalType	java.lang.String
<b>Parameter-1:</b>	
openType	javax.management.openmbean.SimpleType(name=java.lang.Boolean)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

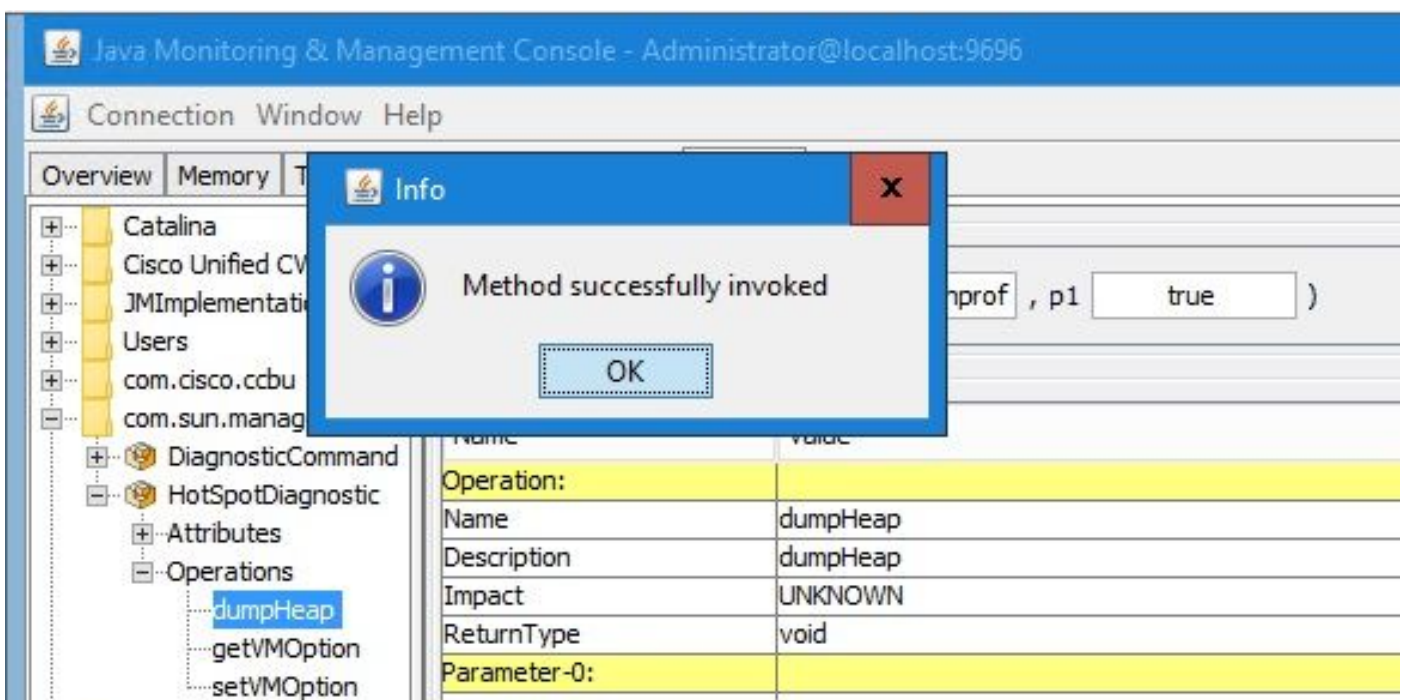
**Opmerking:** Cisco raadt niet aan om de zeep tijdens bedrijfsuren te dumpen aangezien het de dienst van de VXML Server kan veroorzaken om voor een ogenblik tijdens het proces te bevriezen. Cisco raadt aan deze activiteit uit te voeren tijdens niet-werkdagen.

Stap 7. Klik op **dumpenHeap**.



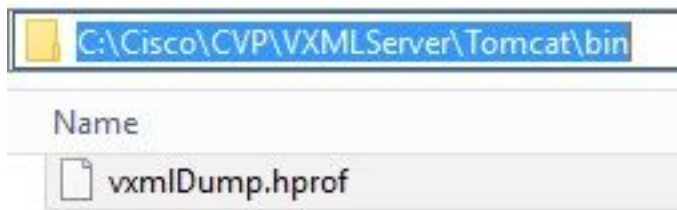


Stap 8. U moet de **aangehaalde** berichtmethode zien. Klik op **OK**.



Stap 9. Verzamel het gegenereerde dummybestand. Het standaard pad voor VXML Server-

dumpen is C:\Cisco\CVP\VXMLServer\Tomcat\bin.



**Opmerking:** dit gereedschap is bedoeld om problemen op te lossen bij het lekken van VXML-servergeheugen. Zodra de toepassing die het lek veroorzaakt is geïdentificeerd is, moet dit gereedschap ingeschakeld zijn, moet de gewenste informatie worden verzameld en moet het, nadat het probleem is herkend, worden uitgeschakeld. Jconsole is niet ontworpen als een monitor tool en moet niet voor onbepaalde tijd worden ingeschakeld.