

Sammeln von Heap Dump und Thread Dump vom CVP VXML-Server

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Problem](#)

[Fehlerbehebung](#)

[Schritte zum Abrufen von Heap-Dump](#)

[Schritte zum Abrufen von Thread-Dump](#)

Einführung

In diesem Dokument wird beschrieben, wie Sie Heap Dump und Thread Dump für Tomcat vom Cisco Customer Voice Portal (CVP) Voice eXtensible Markup Language Server (VXML) sammeln.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über Kenntnisse in folgenden Bereichen zu verfügen:

- CVP
- CVP VXML-Server
- CVP VXML-Anwendungen

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Software- und Hardwareversionen:

- CVP-Version 11.5
- CVP VXML-Server 11.5

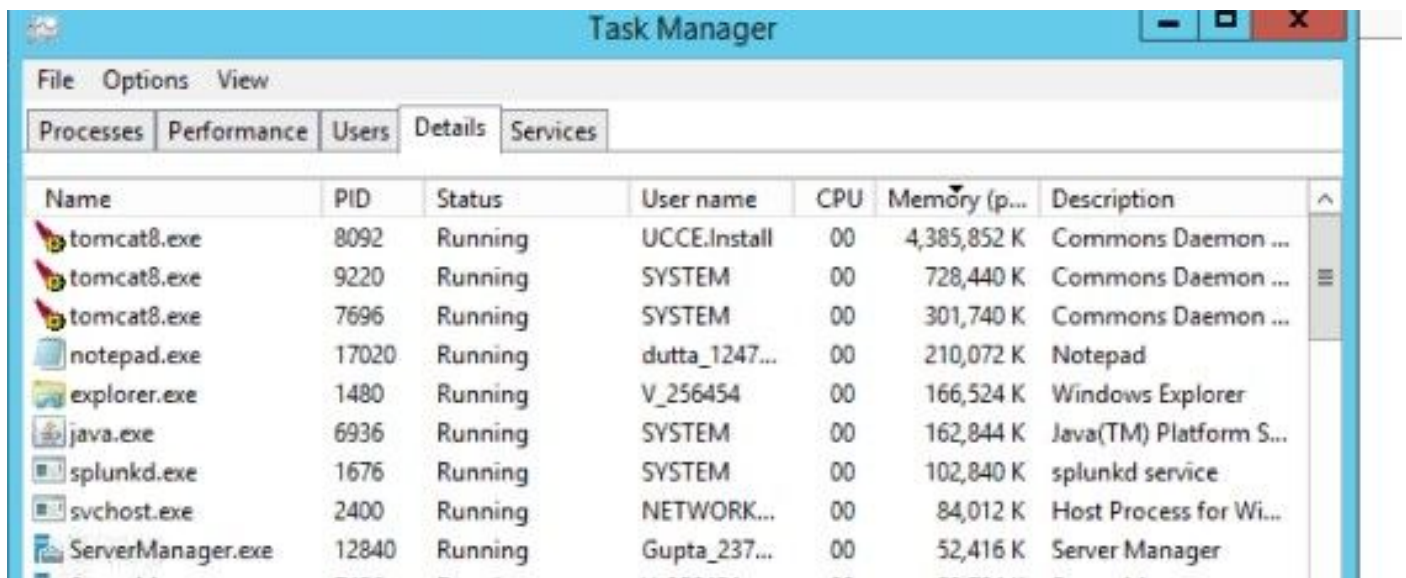
Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Problem

Szenario 1. Während Sie die Leistung des VXML-Servers überwachen, haben Sie festgestellt, dass dieser 4 GB Speicher verwendet. Sie möchten die Ursache des Speicherlecks kennen, bevor er abstürzt.

Szenario 2. Während Sie den VXML-Server überwachen, haben Sie entdeckt, dass der VXML-Server hohe Threads (~ 500) verwendet, was ziemlich ungewöhnlich ist. Sie möchten wissen, wie Sie die Threadverwendung analysieren und den Threadzustand und die Komponente/Anwendung ermitteln, die mehr Threads beansprucht.

Beispiel: VXML Tomcat verbraucht 4,5 GB Arbeitsspeicher, wie im Bild gezeigt.



Name	PID	Status	User name	CPU	Memory (p...	Description
tomcat8.exe	8092	Running	UCCE.Install	00	4,385,852 K	Commons Daemon ...
tomcat8.exe	9220	Running	SYSTEM	00	728,440 K	Commons Daemon ...
tomcat8.exe	7696	Running	SYSTEM	00	301,740 K	Commons Daemon ...
notepad.exe	17020	Running	dutta_1247...	00	210,072 K	Notepad
explorer.exe	1480	Running	V_256454	00	166,524 K	Windows Explorer
java.exe	6936	Running	SYSTEM	00	162,844 K	Java(TM) Platform S...
splunkd.exe	1676	Running	SYSTEM	00	102,840 K	splunkd service
svchost.exe	2400	Running	NETWORK...	00	84,012 K	Host Process for Wi...
ServerManager.exe	12840	Running	Gupta_237...	00	52,416 K	Server Manager

VXML-Speicher bei 4,5 GB

Fehlerbehebung

Schritte zum Abrufen von Heap-Dump

Schritt 1: JDK-Version ähnlich der Version herunterladen, die in CVP installiert ist

```
C:\Cisco\CVP\jre\bin>java -version
java version "1.7.0_51"
Java(TM) SE Runtime Environment (build 1.7.0_51-b13)
Java HotSpot(TM) Client VM (build 24.51-b03, mixed mode)
```

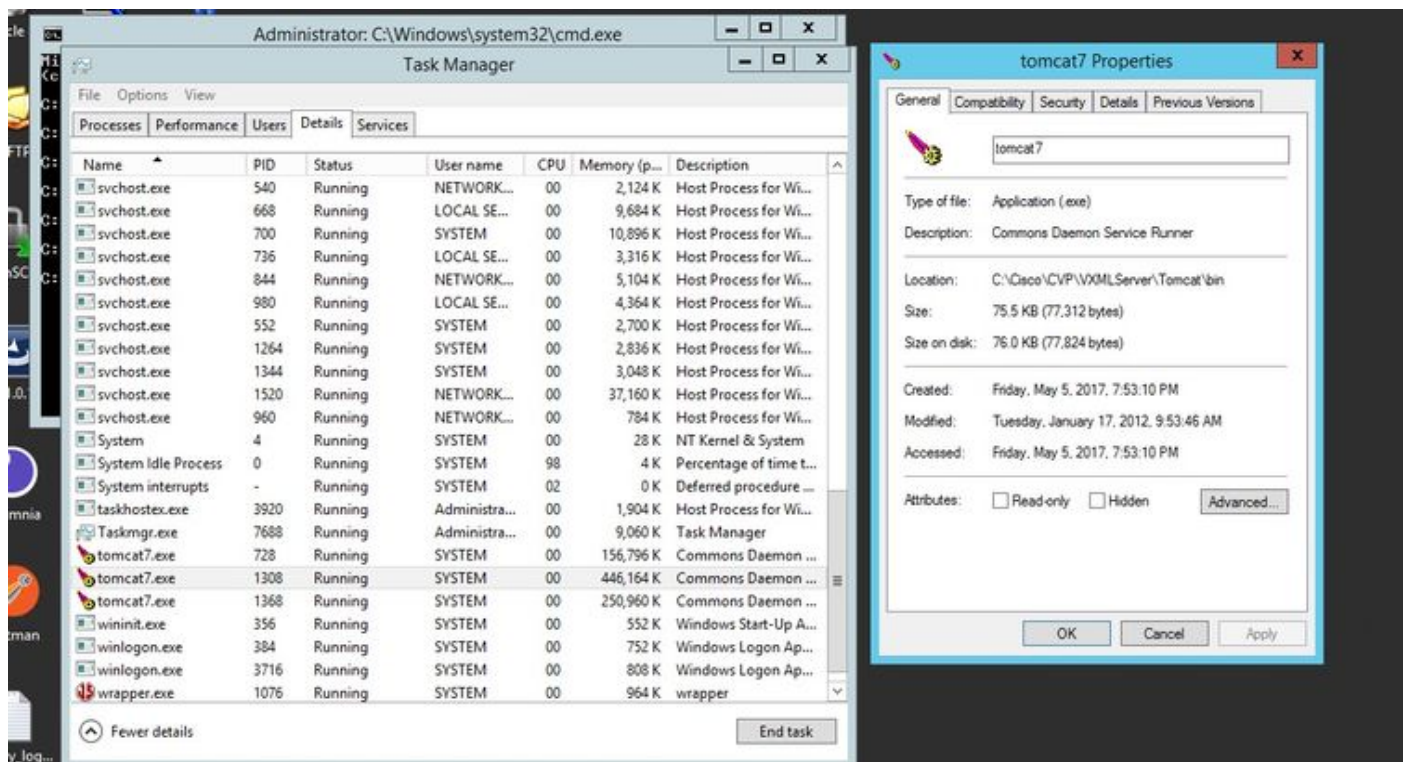
Schritt 2: Kopieren Sie JDK vom Desktop auf den CVP-Server.

```
Download windows JDK exe
Open with 7-Zip
Dump contents into a directory %JDK-EXE%
cmd: cd %JDK-EXE%.rsrc\1033\JAVA_CAB10
cmd: extrac32 111
Now have a tools.zip in directory, open it in 7-Zip
Extract contents into a new directory %JDK-VERSION%
cmd: cd %JDK-VERSION%
cmd: for /r %x in (*.pack) do .\bin\unpack200 -r "%x" "%~dx%-px%~nx.jar"
```

Hinweis: Laden Sie einfach JDK von

<http://www.oracle.com/technetwork/java/javaee/downloads/index.html> herunter und installieren Sie EXE im lokalen Ordner und kopieren Sie das JDK von Ihrem lokalen Rechner nach CVP .

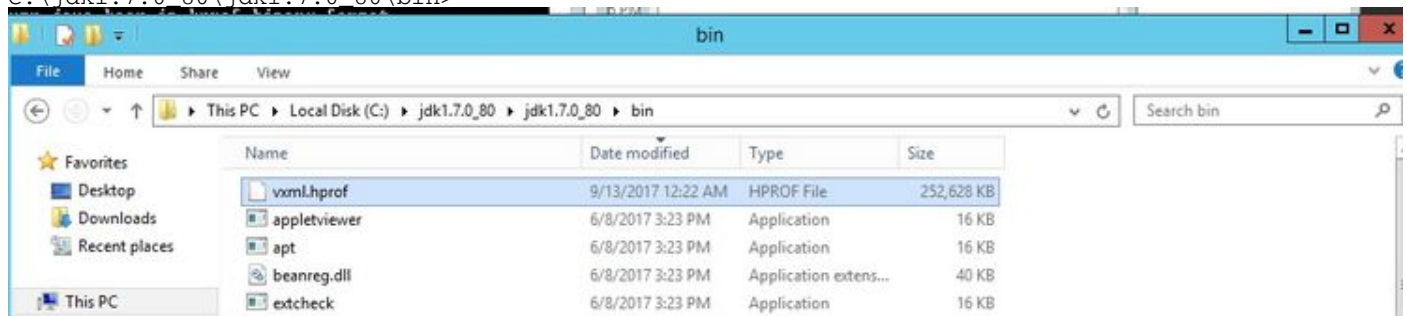
Schritt 3: Erfassen Sie die Prozess-ID (PID) des VXML-Servers vom **Task-Manager**.



Schritt 4: Führen Sie diesen Befehl aus, um HeapDump zu sammeln. (Bsp.: jmap -dump:file=vxml.hprof <PID DER TOMCAT-Instanz>).

```
C:\jdk1.7.0_80\jdk1.7.0_80\bin>jmap -dump:file=vxml.hprof 1308
Dumping heap to C:\jdk1.7.0_80\jdk1.7.0_80\bin\vxml.hprof ...
Heap dump file created
```

```
C:\jdk1.7.0_80\jdk1.7.0_80\bin>
```



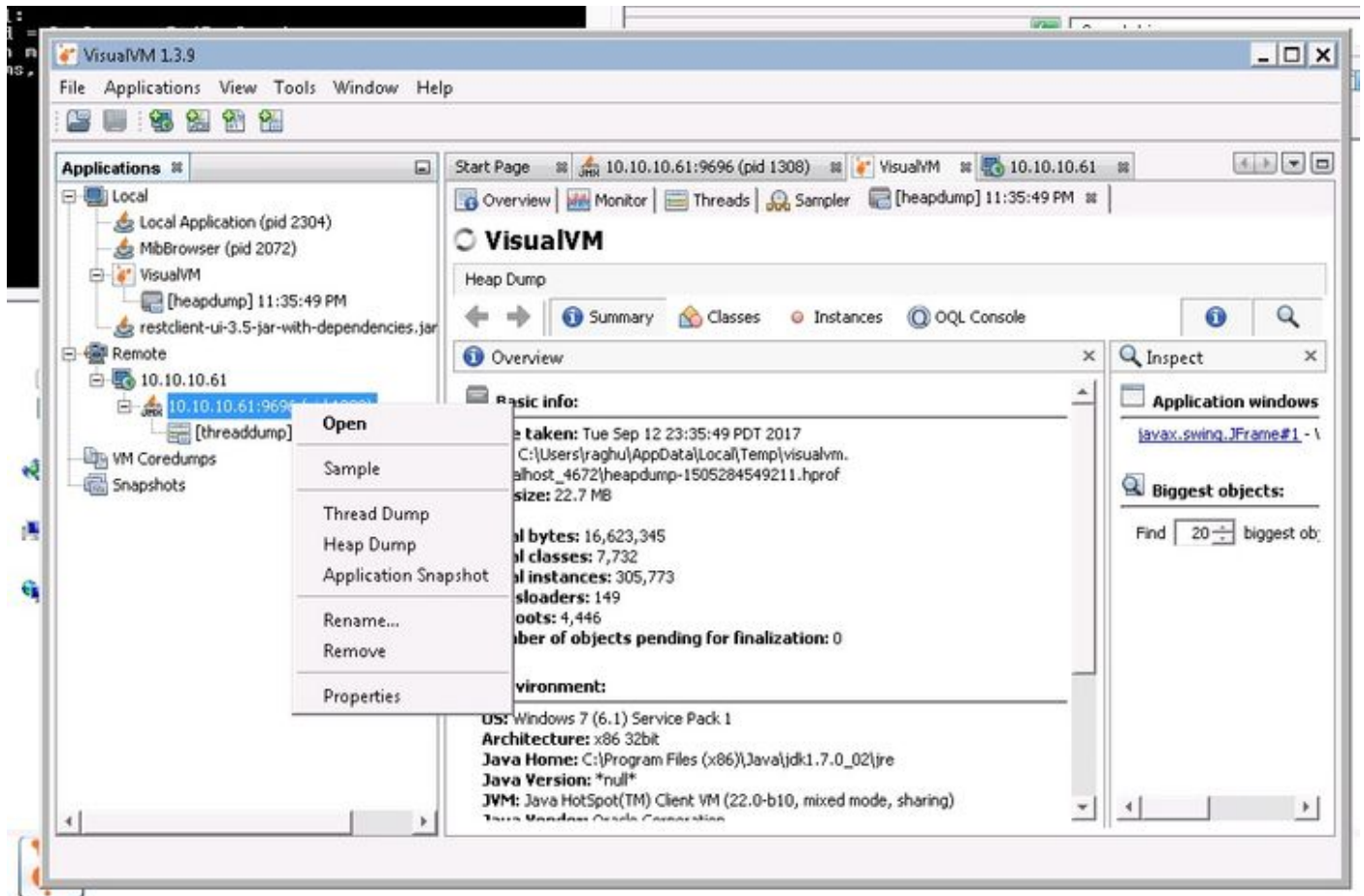
Hinweis: Ihr hProf wird erstellt, Sie können auf das lokale System kopieren und offline überwachen.

Schritte zum Abrufen von Thread-Dump

Thread-Dump-Sammlung ist relativ einfach im Vergleich zu Heap-Dump.

Schritt 1: Herstellen einer Verbindung zum CVP VXML-Server auf <CVP VXML-Server>:9696 (9696 ist der Standard-JMX-Port für VXML-Server), der jVisualVM verwendet.

Schritt 2: Klicken Sie mit der rechten Maustaste auf **JXM Connection** und **Collect Thread Dump**.



Schritt 3: Hier wird Thread-Dump im Remote-Server abgelegt, wo er als Datei gespeichert und für weitere Analysen verwendet werden kann.