

Tidal Enterprise Scheduler: Java Stack Dump mit "jstack" abrufen

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konventionen](#)

[Problem](#)

[Lösung](#)

[Zugehörige Informationen](#)

Einführung

In diesem Dokument wird beschrieben, wie Sie einen Java-Stack-Dump mithilfe des Java-Befehls **jstack** abrufen. Dieser Prozess ist unter Windows, Linux und UNIX sowie auf allen Tidal-Systemen gültig, die Oracle Sun Java Version 1.6+ verwenden.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über Java-Kenntnisse und Kenntnisse des Betriebssystems zu verfügen, auf dem Java ausgeführt wird. Außerdem muss das Java JDK (Java Development Kit) Version 1.6+ auf dem System installiert sein.

Verwendete Komponenten

- Sun Java JRE 1,6+
- Sun Java JDK 1,6+

Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie unter [Cisco Technical Tips Conventions](#) (Technische Tipps zu Konventionen von Cisco).

Problem

Gelegentlich muss der Support Threaddaten auf niedriger Ebene erfassen, die sich nicht in den

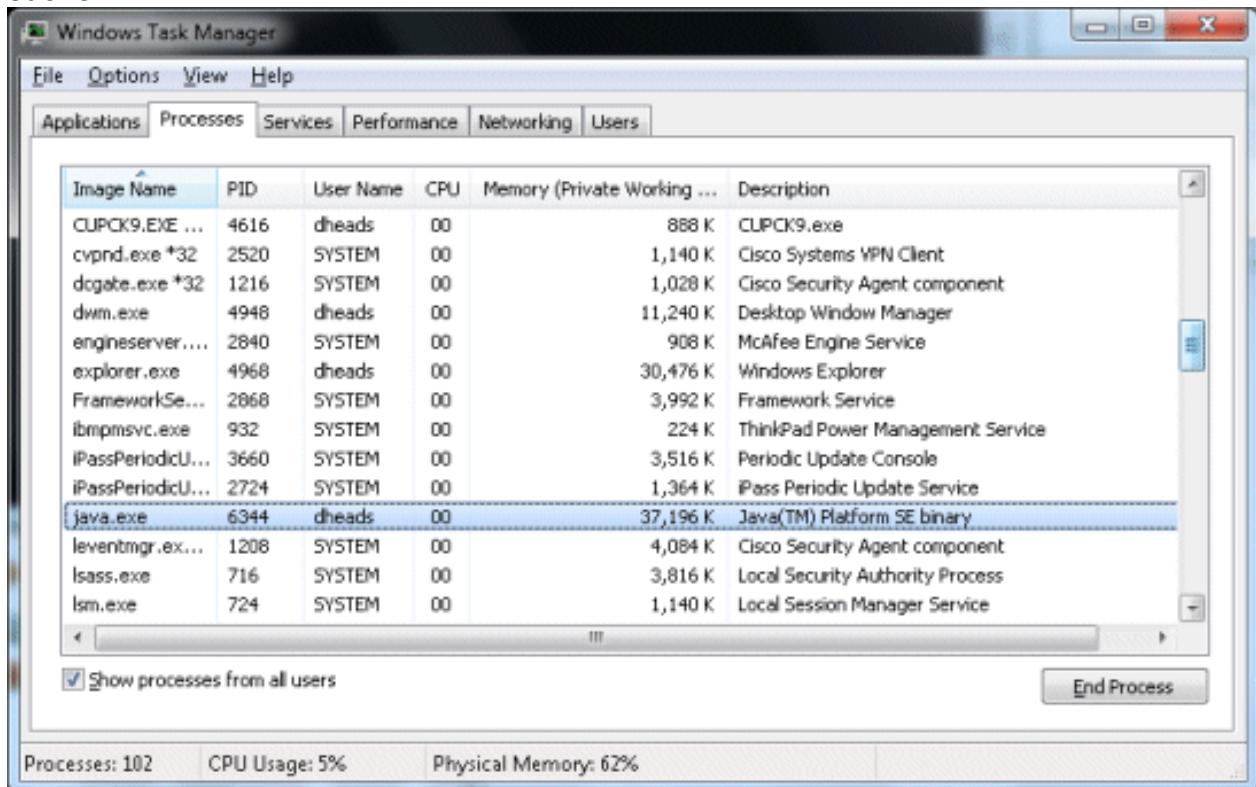
Master-Protokollen befinden.

Lösung

Führen Sie die folgenden Schritte aus, um ein Java-Stack-Dump mit dem von Oracle Sun bereitgestellten **Jstack**-Tool zu erfassen.

Hinweis: Der Java-Prozess muss Sun Java 1.6+ ausführen, um den **jstack**-Befehl auszuführen, und das JDK muss auf dem System installiert sein. Laden Sie das Oracle Sun JDK [hier](#) herunter.

1. Suchen Sie die Prozesskennnummer auf dem Rechner, auf dem der Java-Prozess ausgeführt wird. Auf einem Windows-Computer können Sie mithilfe des Task-Managers die Prozesskennnummer suchen.



Die **PID** ist die Prozesskennnummer. Wenn diese Spalte nicht angezeigt wird, wählen Sie **Ansicht > Spalten > PID (Prozesskennung)**, um sie hinzuzufügen. Unter UNIX/Linux können Sie das **ps -ef** ausgeben. | **grep UnixMaster**-Befehl, um die Prozess-ID abzurufen:

```
root@sunts06:/# ps -ef | grep UnixMaster
root 5452 5423 0 14:43:21 pts/14 0:00 grep UnixMaster
root 1868 1 0 Jun 09 ? 31:49 /usr/bin/java -DN=UnixMaster -DTI
DAL_HOME=/opt/TIDAL/master/bin/.. -Xms256m -Xm
root@sunts06:/#
```

2. Sobald Sie die Prozesskennnummer haben, rufen Sie das **BIN**-Verzeichnis auf, in dem das Sun JDK installiert ist, und geben Sie den Befehl **jstack {PID}** aus, wobei **{PID}** die **Prozesskennnummer** ist. Dadurch werden die Java-Stack-Informationen auf den Bildschirm gedruckt.

```
C:\Program Files\Java\jdk1.6.0_25\bin>jstack 3724
2011-06-30 20:07:37
Full thread dump Java HotSpot(TM) 64-Bit Server VM (20.0-b11 mixed mode):
```

```
"Swing-Shell" daemon prio=6 tid=0x0000000006cce800 nid=0x1bd8 waiting on conditi
```

```
on [0x000000000843f000]
  java.lang.Thread.State:WAITING (parking)
    at sun.misc.Unsafe.park(Native Method)
      - parking to wait for <0x00000000fb38ace8> (a java.util.concurrent.lock
s.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(Unknown Source)
    at java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject
.await(Unknown Source)
    at java.util.concurrent.LinkedBlockingQueue.take(Unknown Source)
    at java.util.concurrent.ThreadPoolExecutor.getTask(Unknown Source)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
    at sun.awt.shell.Win32ShellFolderManager2$ComInvoker$3.run(Unknown
Source)
    at java.lang.Thread.run(Unknown Source)

"Thread-2" prio=6 tid=0x0000000006cd0800 nid=0xe6c waiting on condition [0x00000
0000833f000]
  java.lang.Thread.State: TIMED_WAITING (sleeping)
    at java.lang.Thread.sleep(Native Method)
```

3. Führen Sie zum Umleiten der Stapelinformationen in eine Datei denselben Befehl mit der Option zum Erstellen von Dateien aus, die sowohl in Windows als auch in Linux/UNIX funktioniert:

```
jstack {PID} > [filename.out]
```

[Zugehörige Informationen](#)

- [Technischer Support und Dokumentation - Cisco Systems](#)