

Tidal Enterprise Scheduler:"jstack"을 사용하여 Java 스택 덤프 가져오기

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[표기 규칙](#)

[문제](#)

[솔루션](#)

[관련 정보](#)

[소개](#)

이 문서에서는 Java jstack 명령을 사용하여 Java 스택 덤프를 가져오는 방법에 대해 **설명합니다**.이 프로세스는 Windows, Linux 및 UNIX 및 Oracle Sun Java 버전 1.6 이상을 사용하는 모든 Tidal 시스템에서 유효합니다.

[사전 요구 사항](#)

[요구 사항](#)

Java가 실행되는 운영 체제에 대한 Java 지식 및 지식이 있는 것이 좋습니다.또한 Java JDK(Java Development Kit) 버전 1.6 이상을 시스템에 설치해야 합니다.

[사용되는 구성 요소](#)

- Sun Java JRE 1.6 이상
- Sun Java JDK 1.6+

[표기 규칙](#)

문서 규칙에 대한 자세한 내용은 [Cisco 기술 팁 표기 규칙을 참고하십시오](#).

[문제](#)

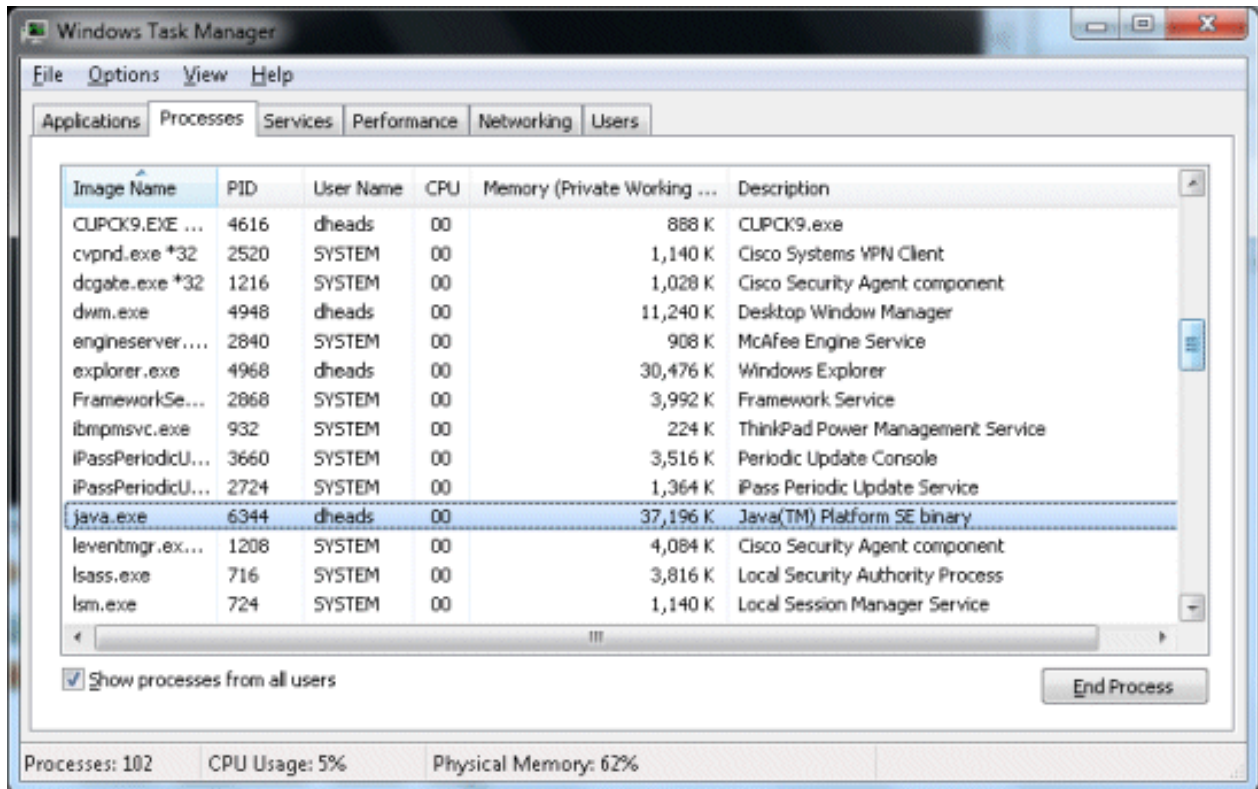
경우에 따라 지원 시 마스터 로그에 없는 하위 수준 스레드 데이터를 캡처해야 합니다.

솔루션

Oracle Sun에서 제공하는 **jstack** 툴을 사용하여 Java 스택 덤프를 캡처하려면 다음 단계를 완료하십시오.

참고: jstack 명령을 실행하려면 Java 프로세스가 Sun Java 1.6 이상을 실행해야 하며 JDK가 시스템에 설치되어 있어야 합니다. [여기](#)에서 Oracle Sun JDK를 [다운로드](#)합니다 .

1. Java 프로세스를 실행하는 시스템에서 프로세스 식별 번호를 찾습니다.Windows 시스템에서 작업 관리자를 사용하여 프로세스 식별 번호를 찾을 수 있습니다



특히 PID는 프로세스 식별 번호입니다.이 열이 표시되지 않으면 **View > Columns > PID(프로세스 식별자)**를 선택하여 추가합니다.UNIX/Linux에서는 **ps -ef**를 실행할 수 있습니다. | **grep UnixMaster** 명령 - 프로세스 ID를 가져옵니다.

```
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. 필요한 프로세스 식별 번호가 있으면 Sun JDK가 설치된 **BIN** 디렉터리로 이동하여 **jstack {PID}** 명령을 실행합니다. 여기서 **{PID}**은 **프로세스 식별 번호**입니다.그러면 Java 스택 정보가 화면에 인쇄됩니다.

```
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 condition [0x000000000843f000]
  java.lang.Thread.State:WAITING (parking)
    at sun.misc.Unsafe.park(Native Method)
      - parking to wait for <0x00000000fb38ace8> (a java.util.concurrent.locks.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. 스택 정보를 파일로 리디렉션하려면 Windows 및 Linux/UNIX 모두에서 작동하는 파일 생성 옵션과 동일한 명령을 실행합니다.

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

관련 정보

- [기술 지원 및 문서 - Cisco Systems](#)