

## How\_to\_debug\_OutOfMemoryError

JVM can crash with OOM Error for multiple reasons.

- Where to check
  - ◆ Check in the application logs for OutOfMemoryError

Exception in thread ?xxx" **java.lang.OutOfMemoryError**: Java heap space

- OutOfMemoryError types
  - ◆ Java heap space
  - ◆ PermGen space
  - ◆ Out of swap space
  - ◆ unable to create new native thread
  - ◆ Requested array size exceeds VM limit
  - ◆ GC overhead limit exceeded
- How to debug different OutOfMemory errors
  - ◆ **Java heap space**
  - ◆ **GC overhead limit exceeded**
    - ◇ Happens due to a memory leak.
    - ◇ Heap dumps are automatically generated at 70/80/90/98% thresholds of the heap size.
      - file name looks like heapDump\_MIVR\_70%Threshold.hprof.
    - ◇ When the application goes OOM, JVM generates a heap dump during the crash.
      - file name looks like java\_pid2665.hprof.
      - Due to a bug CSCte90486, this file can be collected only using remote account.
    - ◇ Collect the heap dumps (present in the service log directory. i.e. MIVR directory for engine heap dumps).
    - ◇ Collect the thread dump present in JVM.log file.
    - ◇ Refer [How to analyze heap dumps](#) for the procedure to analyze a heap dump.
  - ◆ **PermGen space**
    - ◇ Happens most probably due to ClassLoader leak.
    - ◇ When the application goes OOM, JVM generates a heap dump during the crash.
      - file name looks like java\_pid2665.hprof.
      - Due to a bug CSCte90486, this file can be collected only using remote account.
    - ◇ Collect the heap dumps.
    - ◇ Collect the thread dump present in the JVM.log file.
  - ◆ **Out of swap space**
    - ◇ Happens due to a memory leak in the JNI code, or in some native applications.
    - ◇ System is low on memory.
    - ◇ Collect the Perfmon counters of all the services running in the system.
    - ◇ Plot a graph of memory usage and find which process is the culprit.
  - ◆ **unable to create new native thread**
    - ◇ Happens when OS runs out of system resources, or the application has run out of its address space due to excessive threads.
    - ◇ Collect the thread dump present in the JVM.log file.
    - ◇ Debug the perfmon logs to find the faulty application.
  - ◆ **Requested array size exceeds VM limit**

## How\_to\_debug\_OutOfMemoryError

- ◇ Happens due to an application bug or a memory leak.
- ◇ Debug application logs and heap dumps if present.