Tool -3: Thread and Monitor Dump Analyzer for Java
Contents
About TMDA ................................................................................................................................. 3
Obtain and Install the tool ............................................................................................................. 3
Opening a thread dump .................................................................................................................. 5
Analyzing the Dump ...................................................................................................................... 7
  Thread Detail ................................................................................................................................ 8
  Finding blocked thread details .................................................................................................... 9
  Thread aggregation ...................................................................................................................... 10
  Comparing threads ..................................................................................................................... 11
O! that’s nice but when to use this tool? ....................................................................................... 12
Alright, this tool is all to do with Java thread dumps. How do I get those thread dumps? ............ 13
References ..................................................................................................................................... 14
About Author .................................................................................................................................. 15
Connect With US ............................................................................................................................. 16
In this third part of this series, we will be discussing how to diagnose issues with the java threads like: Hung threads, Deadlocks, Resource starvation and server crash. IBM has developed a tool named “Thread and Monitor Dump Analyzer for Java” also known as TMDA. TDMA is a part of the ISA tool.

About TMDA
The IBM Thread and Monitor Dump Analyzer for Java (TMDA) analyzes java cores and diagnoses monitor locks and thread activities to identify the root cause of hangs, deadlocks, and resource contention or monitor bottlenecks. It compares each java core and provides process ID information for threads, garbage collection frequency, allocation failure frequency, and a list of hang suspects.

Obtain and Install the tool
- Start the ISA
- Go to update → find new → tools add-ons
- Expand the JVM based tools from the results and select [tech preview] IBM Thread Monitor and Dump analyzer for Java {TMDA} and install the tool.
- Restart the ISA

Click on Analyze problem and select TMDA and launch the tool.
Tools starting page
Opening a thread dump

- Go to file → open ThreadDumps
- Select multiple thread dump files

Selecting a file from the list, shows the details/summary about that thread dump.
You can find the following information from this page:

- **Cause of the thread dump**
- **Operating system information**
- **Java information**
- **Class path**
- **User arguments**
- **Heap details**
- **Memory segment analysis**
- **Thread status analysis**
- **Thread method analysis**
- **Garbage collection details and history**
Analyzing the Dump

Select the thread dump from the list and click on Analysis. You can do the following analysis:

- Native memory analysis
- Thread detail analysis
- Details of the thread monitor
- Compare threads from different dumps

<table>
<thead>
<tr>
<th>File</th>
<th>Analysis</th>
<th>View</th>
<th>Help</th>
<th>Native Memory Analysis</th>
<th>Thread Detail</th>
<th>Monitor Detail</th>
<th>Compare Threads</th>
<th>Compare Monitors</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>java</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>java</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>java</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Thread Detail

Select a javacore file from the list and go to Analysis → thread detail

Color codes are used to show the status of a thread status. Below is a table of the thread status codes.

<table>
<thead>
<tr>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deadlock</td>
</tr>
<tr>
<td>Runnable</td>
</tr>
<tr>
<td>Waiting on condition</td>
</tr>
<tr>
<td>Waiting on monitor</td>
</tr>
<tr>
<td>Suspended</td>
</tr>
<tr>
<td>Object. wait()</td>
</tr>
<tr>
<td>Blocked</td>
</tr>
<tr>
<td>Parked</td>
</tr>
</tbody>
</table>
Finding blocked thread details
Click on the thread that is shown as blocked from the thread list. On the right side panel, you can see the blocked thread information.

Click on the entry in Blocked by list, shows the blocked thread
**Thread aggregation** details can be found by selecting a dump file from list and clicking on the thread analysis.
Comparing threads

Select multiple thread dumps from the list and click Analysis → compare threads.
O!, that’s nice but when to use this tool?

- To Analyze Java Core files
- To diagnose monitor locks
- To find out java thread activities
- To find out and diagnose hangs in threads
- To diagnose deadlocks
- To diagnose resource bottlenecks/resource starvation
Alright, this tool is all to do with Java thread dumps. How do I get those thread dumps?

If you get unexplained server hangs under WebSphere, you can obtain, from the WebSphere server, a thread dump to help diagnose the problem. Thread dumps can either be generated automatically or manually.

In the case of a server hang, you can force an application to create a thread dump.

On unix/Linux machines find the process id (PID) of the hung JVM and issue kill -3 PID. Look for an output file in the installation root directory with a name like javacore.date.time.id.txt.

Or you can use wsadmin prompt,

- get the handle of the server
  
  wsadmin> set jvm [$AdminControl completeObjectName type=JVM,process=server1,*]

- execute
  
  wsadmin>$AdminControl invoke $jvm dumpThreads

If an application server spontaneously dies, look for a file. The JVM creates the file in the product directory structure, with a name like javacore[number].txt.

From WebSphere Application Server v8 onwards.. You can generate these dumps using administration console as well.
References

1. IBM Thread and Monitor Dump Analyzer for Java (Tech Preview)
2. How to Diagnose Java Resource Starvation
About Author

Joseph Amrith Raj is a technical consultant for multiple WebSphere products. He worked on various product consulting and support teams including WebSphere Application Server, WebSphere MQ, WebSphere Message Broker, Enterprise Service Bus and WebSphere Process Server. He has 7 years of experience in administration, troubleshooting, consulting and he has significant experience in architecture, strategy and leadership positions. He is IBM certified for WAS, WMQ, WPS, SOA and Cloud Computing.
Connect With US

- **Get notified on upcoming tutorials**
  - Subscribe, to get this WebSphere blog posts directly to your inbox: [here](#)

- **Tell us your feedback and what you want**
  - Please leave your comments in the comments section below the post
  - Send us suggestions for future topics and improvements about the content at [webspherelibrary@live.com](mailto:webspherelibrary@live.com)

- **Be Connected!**
  - Connect with us on [Facebook](#)
  - Connect with us on [Twitter](#)
  - More about us: [profile](#)