TJP User Guide

Table of Contents

Welcome to Topaz for Java Performance

Topaz for Java Performance provides the information needed to tune a Java Virtual Machine (JVM) on z/OS. There are several items that affect the performance of a JVM. Topaz for Java Performance is targeted at WebSphere and Java Batch. The JVMs in these different environments will encounter completely different program load characteristics. For example, the WebSphere JEE environment will run many different relatively small transactions whereas the Java Batch typically executes a single long-running program. These different program characteristics require different resource allocation in the JVMs. These resources include memory, CPU, classes and threads in the JVMs. The right combination of these resources is vital for the efficient execution of programs and transactions.

Topaz for Java Performance focuses on the sampling memory, threads, CPU usage, and classes used. Since the JVM manages the allocation and release of memory, one important area of tuning is memory garbage collection. Garbage collection finds unused objects and compacts them into one area where the memory can be reallocated. The garbage collection event is started when the JVM determines it is running low on memory. The IBM JVM provides a few different garbage collection algorithms to try to optimize the timing and approach to reclaiming unused memory objects. The Topaz for Java Performance JVM Memory Chart helps you see the garbage collection occurring in the JVM. This chart will show whether garbage collection is occurring too frequently so increased memory allocation can be considered. Topaz for Java Performance also shows the existing memory parameters passed to the JVM.

Threads in a Java Program are a unit of execution that can run on different general CPU (GCPU) or specialty processors on z/OS, such as zAAP processors. Threads may try to access shared resources, generally memory. Sharing objects between threads can mean that a thread is waiting for resource that isn't available and therefore isn't doing any work. Topaz for Java Performance provides a table of threads that includes the percent of time a thread is waiting. These waiting values provide an indication of how much work a thread can accomplish. If a thread is constantly waiting, the Java program may have encountered a bug where the resource isn't released or the program is poorly designed. In either case, a high thread wait time means some investigation is required in a Java program or transaction.

On System z running z/OS Unix, there is always concern about the amount of CPU being used because of the charges associated with its use. IBM has provided some relief from these CPU charges by introducing specialty processors that have a different cost structure. Java code may be eligible to run on the specialty processors and at a reduced cost. This advantage is one of the attractions of running Java programs on z/OS Unix. Compuware's iStrobe product provides a view of the amount of GCPU used by a job running the JVM and it provides a view of the amount of specialty engine CPU used the job. This provides an "outside" view of the JVM CPU usage. Inside the JVM iStrobe has limited knowledge of the type of processor on which a thread is executing. Topaz for Java Performance provides information about performance inside of the JVM. Topaz for Java uses a sampling approach to approximate the amount of CPU that is by different classes. This technique provides low cost measurements that guide users to the most resource-intensive Java classes in your Java program.

The information available from Topaz for Java Performance can guide your JVM tuning approaches to improve JVM transaction throughput and reduce program execution times.

To launch Topaz for Java Performance

Do either of the following:

- In a browser, enter the following: *<machine name>*:*<CES web application listener port>*/topazjava.

- On the **Compuware Enterprise Services** landing page, click ☰ and select **Topaz for Java Performance**. If it is not installed, **Topaz for Java Performance** is disabled.

When initially launched, Topaz for Java Performance opens to a page from which users can access the product features:

- **Measurements** allows users to view, submit, and manage measurements.

- **Trend Reports** allows users to understand the java program performance over a series of measurement time frames.

From the **Topaz for Java Performance** home page, users can access the **Measurement** page, access the **Trend Reports** page, launch the online help, and view version, build, copyright, and other Topaz for Java Performance information.

To access the Measurements page

Do one of the following:

- Click .
- Click and then select **Measurements**.

The **Measurements** page appears.

To access the Trend Reports page

Do one of the following:

- Click .
- Click and then select **Trend Reports**.

The **Topaz for Java Trend Reports** page appears.

To launch the online help

Click . Online help for the selected page appears.

To view version, build, copyright, and other Topaz for Java Performance information

Click . The **About Topaz for Java Performance** dialog box appears. As well, you can collect support files to help Compuware Customer Support evaluate issues for resolution.

Using the Measurements Page

Topaz for Java Performance's **Measurements** page is used to view, submit, and manage measurements and shows the following:

- Measurements that have been performed
- Active measurements
- Startup measurement requests
- Scheduled measurement requests
- Java applications that have a Topaz for Java Performance agent configured
- Disconnected agents

The **Measurements** page is used to do the following:

- Submit any of the three types of measurement requests:
  - An immediate measurement request
  - A startup measurement request, which will run when the selected agent starts up
  - A scheduled measurement request, which will run at the specified date, time, and recurrence
- Stop running a measurement request
- View a measurement
- Delete a measurement
- Delete an agent
- Change the columns displayed

Only one measurement can run at a time on a given JVM.

To access the Measurements page

From the Topaz for Java Performance landing page, do one of the following:

- Click .
- Click and then select **Measurements**.

The **Measurements** page appears.

To use filtering on the Measurements page

To filter the measurements and agents that appear on the **Measurements** page, do any of the following in the **Filter** field:

- Enter one or more values separated by a space. For example, enter "CPU JZOS" to return only the rows that contain both "CPU" and "JZOS".
- Enter "Measurement" to return only rows with the status of **Measurement**. If desired, add the agent name to return only rows with the status of **Measurement** on a given agent.
- Enter "Disconnected" to return only rows with the status of **Disconnected**. If desired, add the agent name to return only rows with the status of **Disconnected** on a given agent.
- Enter "Waiting" to return only rows with the status of **Waiting**. If desired, add the agent name to return only rows with the status of **Waiting** on a given agent.
- Enter "Queued" to return only rows with the status of **Queued**. If desired, add the agent name to return only rows with the status of **Queued** on a given agent.

The **Measurements** page displays only the rows on which the entered characters appear in any column.

Note: Topaz for Java Performance allows agent installers to use custom naming conventions for agents.
End users need to know the convention used so they recognize the agent on the Measurements page. If you

don't recognize the agent to use, it may be that the person who performed the agent installation chose to use a custom naming convention for the agent. Ask the installer for the name that was used.

To submit an immediate measurement request

1. On the **Measurements** page, right-click an agent and select **Measurement Request>Submit Immediate Request**. The **Immediate Measurement Request** dialog box appears.

   > **Note:** If the agent is in **Active** status, **Submit Immediate Request** will be disabled and another measurement cannot be requested until the agent returns to **Waiting** status.

2. In the **Measurement report name** field, enter a name for the report.
3. In the **Estimated measurement time (minutes)** field, use the arrows to choose the desired length of time for the measurement to run.
4. In the **Number of samples** field, use the arrows to indicate the number of performance samples to be taken within the measurement time specified.

   > **Note:** The maximum number of samples per minute is 200.

5. Optionally, in the **Notification e-mail address** field, enter an email address to be sent notification when the measurement has completed.
6. Optionally, select the **Overwrite existing measurement report** check box to overwrite any existing measurement report that has the same name.
7. Optionally, click **Thresholds** to open the **Thresholds** dialog box to set minimum and maximum thresholds for the charts on the **JVM Information** page. Select the check boxes for the thresholds to be set (**CPU %**, **Heap**, **Classes**, or **Threads**), enter the desired minimum and maximum values, and click **OK**. If a threshold is set, the affected charts will display a shaded bar indicating the measurements that fell within the threshold's minimum and maximum values.
8. Click **Submit**. The **Measurement** page shows the status of the agent is **Running** and the measurement runs.
9. When the measurement time requested has elapsed, click the browser's **Refresh** button or **F5**. If the measurement has completed, the agent's status returns to **Waiting** and a new **Measurement** entry with column details appears for the measurement that ran. From here, users can view the measurement and Java performance reports.

Notes:

- Topaz for Java Performance allows agent installers to use custom naming conventions for agents. End users need to know the convention used so they recognize the agent on the Measurements page. If you don't recognize the agent to use, it may be that the person who performed the agent installation chose to use a custom naming convention for the agent. Ask the installer for the name that was used.
- At any time, click **Reset** to reset the fields to their default values.
- It is not necessary to wait until the agent finishes running before submitting a measurement request for another agent. Further, startup measurement requests and scheduled measurement requests can be submitted for the original agent while it is running a request.

To submit a startup measurement request

1. On the **Measurements** page, right-click an agent and select **Measurement Request>Submit Startup Request**. The **Startup Measurement Request** dialog box appears.
2. In the **Measurement report name** field, enter a name for the report.
3. In the **Estimated measurement time (minutes)** field, use the arrows to choose the desired length of time for the measurement to run.
4. In the **Number of samples** field, use the arrows to indicate the number of performance samples to be taken within the measurement time specified.

   > **Note:** The maximum number of samples per minute is 200.

5. Optionally, in the **Notification e-mail address** field, enter an email address to be sent notification when the measurement has completed.
6. Optionally, select the **Overwrite existing measurement report** check box to overwrite any existing measurement report that has the same name.

7. Optionally, click **Thresholds** to open the **Thresholds** dialog box to set minimum and maximum thresholds for the charts on the **JVM Information** page. Select the check boxes for the thresholds to be set (**CPU %**, **Heap**, **Classes**, or **Threads**), enter the desired minimum and maximum values, and click **OK**. If a threshold is set, the affected charts will display a shaded bar indicating the measurements that fell within the threshold's minimum and maximum values.
8. Click **Submit**. The **Measurement** page shows the status of the agent is **Queued**.
9. The next time the selected agent starts up, the status of the agent changes to **Running** and the measurement runs.
10. When the measurement time requested has elapsed, click the browser's **Refresh** button or **F5**. If the measurement has completed, the agent's status changes to **Waiting** and a new **Measurement** entry with column details appears for the measurement that ran. From here, users can view the measurement and Java performance reports.

Notes:

- Topaz for Java Performance allows agent installers to use custom naming conventions for agents. End users need to know the convention used so they recognize the agent on the Measurements page. If you don't recognize the agent to use, it may be that the person who performed the agent installation chose to use a custom naming convention for the agent. Ask the installer for the name that was used.
- At any time, click **Reset** to reset the fields to their default values.
- It is not necessary to wait until the agent finishes running before submitting a measurement request for another agent. Further, startup measurement requests and scheduled measurement requests can be submitted for the original agent while it is running a request.

To submit a scheduled measurement request

1. On the **Measurements** page, right-click an agent and select **Measurement Request>Submit Scheduled Request**. The **Scheduled Measurement Request** dialog box appears.
2. In the **Measurement report name** field, enter a name for the report.
3. In the **Estimated measurement time (minutes)** field, use the arrows to choose the desired length of time for the measurement to run.
4. In the **Number of samples** field, use the arrows to indicate the number of performance samples to be taken within the measurement time specified.

> **Note:** The maximum number of samples per minute is 200.

5. Optionally, in the **Notification e-mail address** field, enter an email address to be sent notification when the measurement has completed.
6. Optionally, select the **Overwrite existing measurement report** check box to overwrite any existing measurement report that has the same name.
7. Optionally, click **Thresholds** to open the **Thresholds** dialog box to set minimum and maximum thresholds for the charts on the **JVM Information** page. Select the check boxes for the thresholds to be set (**CPU %**, **Heap**, **Classes**, or **Threads**), enter the desired minimum and maximum values, and click **OK**. If a threshold is set, the affected charts will display a shaded bar indicating the measurements that fell within the threshold's minimum and maximum values.
8. Use the **Starting times** calendar to select the day(s) and time(s) for the measurement to begin. If the Agent, Manager, and web browser interface are in different time zones, the measurement needs to be scheduled based on the time zone of the Manager.
9. Set the **Starting date** and **Ending date** fields to reflect the desired start and end dates for the measurement. For a one-time measurement, set the start and end dates to the same date. For a recurring measurement, set the start and end dates accordingly.
10. Click **Submit**. The **Measurement** page shows the status of the agent is **Scheduled**.
11. At the scheduled date and time, the status of the agent changes to **Running** and the measurement runs. If, at that time, the agent is running another measurement, the scheduled measurement will not run.
12. When the measurement time requested has elapsed, click the browser's **Refresh** button or **F5**. If the measurement has completed, the agent's status changes to **Waiting** and a new **Measurement** entry with column details appears for the measurement that ran. From here, users can view the measurement and Java performance reports.

Notes:

- Topaz for Java Performance allows agent installers to use custom naming conventions for agents. End users need to know the convention used so they recognize the agent on the Measurements page. If you don't recognize the agent to use, it may be that the person who performed the agent installation chose to use a custom naming convention for the agent. Ask the installer for the name that was used.
- At any time, click **Reset** to reset the fields to their default values.
- It is not necessary to wait until the agent finishes running before submitting a measurement request for another agent. Further, startup measurement requests and scheduled measurement requests can be submitted for the original agent while it is running a request.

To edit a scheduled measurement request

1. On the **Measurements** page, right-click the agent with the scheduled measurement request and select **Measurement Request>Edit Scheduled Request**. The **Scheduled Measurement Request** dialog box appears.
2. Make any desired changes to the measurement request.
3. Click **Submit**. The **Measurement** page updates to reflect the changes made to the request.

Note: Each agent can have only one scheduled measurement request.

To stop running a measurement request

On the **Measurements** page, right-click the agent and select **Measurement Request> Stop Request**. The measurement request is cancelled.

Notes:

- Topaz for Java Performance allows agent installers to use custom naming conventions for agents. End users need to know the convention used so they recognize the agent on the Measurements page. If you don't recognize the agent to use, it may be that the person who performed the agent installation chose to use a custom naming convention for the agent. Ask the installer for the name that was used.
- The measurement can only be stopped when its status is Running.

To view a measurement

On the **Measurements** page, right-click the measurement and select one of the following:

- To view the measurement in the same window, select **View Measurement**. The measurement appears in the same window.
- To view the measurement in a separate window, select **View Measurement in New Tab**. The measurement appears in a separate tab.

Note: Topaz for Java Performance allows agent installers to use custom naming conventions for agents. End users need to know the convention used so they recognize the agent on the Measurements page. If you don't recognize the agent to use, it may be that the person who performed the agent installation chose to use a custom naming convention for the agent. Ask the installer for the name that was used.

To import a measurement into another instance of Topaz for Java Performance

1. Copy the .bin file for the measurement to be imported (named <*measurementname.bin*> by default and located in C:\ProgramData\Compuware\CES\data\cjp\reports by default). The .bin file is the measurement file collected from the mainframe agent.
2. Paste the file into the corresponding location for the second instance of Topaz for Java Performance.
3. Restart Compuware Enterprise Services.
4. Restart the Manager. The .bin file is processed and the measurement appears on the **Measurements** page.

Notes:

- Topaz for Java Performance allows agent installers to use custom naming conventions for agents. End users need to know the convention used so they recognize the agent on the Measurements page. If you don't recognize the agent to use, it may be that the person who performed the agent installation chose to use a custom naming convention for the agent. Ask the installer for the name that was used.

- A folder name cannot exist in the target location with the same name as the .bin file. (For example, XFSRO11S.bin to cannot be imported in the target location if a folder XFSRO11S already exists in the target location.)

To delete a measurement

On the **Measurements** page, right-click the measurement and select one of the following:

- To delete the files stored on disk both locally and from the database, select **Delete>Measurement>With Database**. The files are deleted both locally and from the database and are not used in trend reporting.
- To delete the files stored on disk locally but not from the database, select **Delete>Measurement>Without Database**. The files are deleted locally but remain in the database for use in trend reporting.

Note: Topaz for Java Performance allows agent installers to use custom naming conventions for agents. End users need to know the convention used so they recognize the agent on the Measurements page. If you don't recognize the agent to use, it may be that the person who performed the agent installation chose to use a custom naming convention for the agent. Ask the installer for the name that was used.

To delete an agent

On the **Measurements** page, right-click the agent and select **Delete>Agent**. The agent is deleted both locally and from the database.

Notes:

- Topaz for Java Performance allows agent installers to use custom naming conventions for agents. End users need to know the convention used so they recognize the agent on the Measurements page. If you don't recognize the agent to use, it may be that the person who performed the agent installation chose to use a custom naming convention for the agent. Ask the installer for the name that was used.
- A disconnected agent is the only agent that can be deleted.

To change the columns displayed

1. On the **Measurements** page, right-click the agent and select **View Columns**. The **Select Columns** dialog box appears.
2. Use the check boxes and the **Select All**, **Deselect All**, and **Reset** buttons to select the columns to be displayed in the **Measurements** page.
3. Click **OK**. The **Measurements** page refreshes to show only the selected columns.

By default, all available columns display, however only **Measurement Name** and **Agent Name** are required columns. The following columns are available:

- Status: The status of the agent or measurement. Possible values include the following:
    - ⊘Disconnected - The agent is not running.
    - 🔴Waiting - The agent is running but isn't taking a measurement.
    - ▶Active - The agent is running and is actively taking a measurement.
    - ✅Measurement - A successful measurement report.
    - ❌Invalid - A measurement report that contains errors.
    - ⌛Queued - A measurement that is set to start when the selected agent starts up.
    - ⚠[Registration Error](#) - The agent couldn't successfully register due to an error.
    - 🕐Scheduled - A measurement that is schedule to start at a specific date and time.
- Measurement Name: The name of the measurement report.
- Initiated By: The product that initiated the measurement. Possible values include the following:
    - Strobe - The measurement was initiated from a Strobe measurement request.
    - Topaz for Java Performance - The measurement was initiated from a Topaz for Java Performance measurement request.
- Agent Name: The name of the agent that was used to create a measurement report or, for non-measurement rows, the name of the agent.

- Job Name: The name of the job being measured.
- System Name: The name of the system on which the measurement is run.
- LPAR: The LPAR on which the measurement is run.
- Step Name: The name of the step that is executing when the measurement is run.
- Program Name: The name of the program that is running when the measurement is run.
- Average CPU Time (ms): The average time the CPU used, in milliseconds, while the measurement was run.
- Maximum CPU Time (ms): The maximum time the CPU used, in milliseconds, while the measurement was run.
- Average Heap (Mb): The average heap, in megabytes, that was used while the measurement was run.
- Maximum Heap (Mb): The maximum heap, in megabytes, that was used while the measurement was run.
- Average Thread Count: The average number of threads that were live while the measurement was run.
- Maximum Thread Count: The maximum number of threads that were live while the measurement was run.
- Start Time: The date and time for when the measurement was started.
- Total Samples: The total number of samples that were taken for a measurement.
- Duration (minutes): The length of time, in minutes, that the measurement ran.

Viewing the Measurement and Java Reports

The **JVM Information** page contains charts about CPU usage, heap, classes, and threads, as well as detailed tables about JVM arguments and system properties.

To view the measurement

On the **Measurements** page, click the measurement name. The **JVM Information** page appears.

To view threshold information

If thresholds were set on the measurement, shaded areas on the applicable charts show how the samples performed against the minimum and maximum threshold limits.

To zoom in on chart information

1. Select the chart then click and drag on the portion of the chart to be zoomed in on. Each of the charts on the **JVM Information** page synchronously zooms to the same level, making it easier to correlate data across the charts.

2. Double-click the chart to return it to normal.

To view detailed chart information

The **Classes** and **Threads** charts are linked to reports that give more detailed information.

Click a node in the chart to open the corresponding report, as follows:

- The **Classes** chart opens the Java Call Stacks report.
- The **Threads** chart opens the Threads report.

To use the JVM Arguments table

The **JVM Arguments** table contains information about options passed to the JVM when starting the Java application. Hover over any argument to see a tool tip description of the argument.

To use the System Properties table

The **System Properties** table contains information about the Java application that is running. Hover over any property to get a tool tip description of the property.

Viewing Trend Reports

The **Topaz for Java Trend Reports** page is used to view and manage trend reports to understand the Java program performance over a series of measurement time frames. The reports summarize Topaz for Java Performance measurements to help identify trends based on data gathered from measurements. The reports are rendered using the Eclipse-based Business Intelligence and Reporting Tool (BIRT). Information about BIRT can be found at the Eclipse website (http://www.eclipse.org/birt).

The **Topaz for Java Trend Reports** page shows the following trend reports:

- **Java CPU and Heap Performance**: Provides an overview of all samples taken for a given date range.

- **Java CPU and Heap Performance for Time Interval**: Provides an overview of all samples taken for a given date range but further specifies a particular clock time period for those days within the date range.

To access the Trend Reports page

From the Topaz for Java Performance landing page, do one of the following:

- Click .
- Click and then select **Trend Reports**.

The **Topaz for Java Trend Reports** page appears, showing the following reports for the measurements that have been run:

- Java CPU and Heap Performance
- Java CPU and Heap Performance for Time Interval

To specify a data range

From the **Date range** list, select a date range or click in the **From** or **To** boxes and modify the dates to create a custom range.

To specify a time range

Note: This functionality applies to the **Java CPU and Heap Performance for Time Interval** report.

From the **Time range** boxes, select a time by clicking the time and specifying the hour or minute. Use the directional arrows in the box to move between hours and minutes. Click the button in the center of the directional arrows to automatically select the current time.

To use filtering on the Topaz for Java Trend Reports page

To filter the reports that appear on the **Topaz for Java Trend Reports** page, in the **Filter** field, enter one or more values separated by a space, such as a partial report name, group name, or description. The **Topaz for Java Trend Reports** page displays only the rows on which the entered characters appear in any column.

Java Call Stacks Report

The Java Call Stacks report shows the call stacks for the methods that were executing at the time the samples were taken for the Java application being measured. The methods are determined by examining the threads that are inside the JVM when the sample is taken. This can help identify possible sources of performance issues in the Java application by determining the methods where the application is spending most of its time.

Information includes the following:

- Method name: The name of the method that was executing as part of a call stack during the measurement.
- Executing %: The percentage of the occurrences that the class was observed on the top of the call stack and in a 'Runnable' status. These methods are actively processing on the systems processors at the time of the measurement.
- Non-executing %: The occurrences that the class was observed at the top of the stack and was in one of the other thread statuses (waiting, timed waiting, suspended, blocked, zombie). These methods are unable to perform processing at the time of the measurement, usually due to resources required to process being unavailable at the moment of the measurement.
- Observed count: The number of times the method was observed at the top of a call stack during the measurement.

For example, assume there were two methods that were observed (method A and method B) for a total of 10 observations. Method A was observed executing 4 times and non-executing 1 time, while method B was observed executing 1 time and non-executing 4 times. The calculated percentages (<*the observed count of method executions for this method*> ÷ <*the observed count of all method executions*>) for those two methods would be as follows:

- Method A: 40% executing and 10% non-executing = 5/10
- Method B: 10% executing and 40% non-executing = 5/10

In this scenario, Method A could be investigated to determine whether it can be optimized to process more efficiently, and Method B could be investigated to determine what resources are preventing it from being able to complete its processing.

To use the Java Call Stacks report

1. Do one of the following:

    - On the **JVM Information** page, click a node in the **Classes** chart.
    - Click , then select **Reports>Java Call Stacks**.

    The **Java Call Stacks** report appears.

2. Select or clear the **Include JVM internal methods** check box to include or exclude methods run by the JVM and not as part of a specific application. Note that selecting this check box will also include Topaz for Java Performance collection data in the Java Call Stacks report. By default, this information is excluded. If changes are made to this setting, the report is automatically rerun and the report is refreshed.

3. To get detailed information, do any of the following:

    - To expand a node, click or right-click the node and select **Expand**.
    - To collapse a node, click or right-click and select **Collapse**.
    - To fully expand the first branch under a selected node, right-click the node and select **Expand Branch**.
    - To collapse all nodes below the selected node, right-click and select **Collapse All**.

    The subentries indicate the methods that called the top-level method.

Threads Report

The Threads report shows the threads seen by the agent and the time and percentage each thread was in each state. It shows the following information for the chosen measurement:

- Thread Name: The name of the thread.
- Thread Group Name: The name of the group to which the thread belongs if it belongs to a group.
- Thread Type: Specifies whether the thread is a Daemon thread or an active thread. Valid values are Daemon and Live.
- State of the thread: The state of the thread during the measurement. Since the state can change during the measurement, the thread state can be one or more of the following:
  - Runnable: The thread is able to run when given the chance.
  - Waiting: A thread that is waiting indefinitely for another thread to perform a particular action is in this state.
  - Timed Waiting: A thread that is waiting for another thread to perform an action for up to a specified waiting time is in this state.
  - Suspended: The thread has been suspended by another thread.
  - Blocked: The thread is waiting to obtain a lock that something else currently owns.
  - Zombie: The thread has been killed.
- Total Thread Time: The total amount of time this thread was observed in any of the states.

To use the Threads report

1. Do one of the following:
   - On the **JVM Information** page, click a node in the **Threads** chart.
   - Click , then select **Reports>Threads**.

   The **Threads** report appears. By default, the table shows the time and percent for each thread state.

2. Select or clear the **Include Topaz for Java Performance data** check box to include or exclude Topaz for Java Performance collection data from the Threads report. By default, this information is excluded. If changes are made to this setting, the report is automatically rerun and the report is refreshed.
3. Optionally, to change the thread states to display differently, select any of the following from the **Show in a Thread State** dropdown list:

   1.
   - Time and Percent (default)
   - Time
   - Percent

   The table refreshes to reflect the selection.

Configuring Email Notification

The email notification option allows automated email messages to be sent to users when measurements have been downloaded. The email contains a link to the measurement.

To configure email notification

1. From the **Topaz for Java Performance** home page, click and then select **Email Settings**. The **Email Settings** page in Compuware Enterprise Services appears.

2. Refer to the *Compuware Enterprise Services User Guide* for configuration details.

Error Messages

Agent Error Messages

Agent processing is traced and the log files are created within a subdirectory called tjp_log in the root directory (user.home by default). The following messages may appear:

Message Sequence	Meaning
Unable to create connection to Manager at localhost: *<host>*	The Topaz for Java Performance Agent cann the manager.
• Processing Agent shutdown hook. • Measurement runtime stopped. • Profiler runtime stopped. • Un-registered from Topaz for Java Performance Manager. • Agent TCP/IP listener command processing ended. • AgentListener stopped. • Agent TCP/IP listener thread ended. • Agent shutdown hook completed.	The Topaz for Java Performance Agent shut because the JVM ended.
• A SocketException. Exceptionjava.net.SocketException: Connection reset • Processing request response from Manager. • Agent TCP/IP listener command processing ended. • Resetting Agent TCP/IP listener thread to reconnect to manager. • Unable to create connection to Manager at localhost:*<host>*. Exception: java,net.ConnectException: Connection refused: connect. • Waiting '1' minute(s) to retry connect to Manager at localhost:*<host>*.	The Topaz for Java Performance Manager h
Unable to register with Topaz for Java Performance Manager ReturnCode: 8 ReasonCode: 128 Text:No License.	The license check failed.
• Unable to write data '*<directory/file where agent captures measurement data>*' Exception: java.io.IOException: There is not enough space on the disk • Measurement ended, because an exception occurred capturing data. Exception: There is not enough space on the disk **Note:** *<directory/file where agent captures measurement data>* is formatted as in the following example: "/SYSTEM/tmp/tjp_data/LPAR1_USER001A_10_20_0_125/tjp_measurement	There is not enough disk space to write the r file.

_2015-05-26-13-17-09-716.dat"
 	
where:
 	
   ▪ LPAR is the user's LPAR.
   ▪ USER001A is the id jobName.
   ▪ 10_20_0_125 is the IP address of the user's machine.
   ▪ 2015-05-26-13-17-09-716 is the current date.	

• 2015-06-01 12:30:28.549 INFO 1 com.compuware.cjp.agent.timer.Sampler.<init> - JNI functions will not be used capture process CPU time, because tjp.measurement.jni.processcputime=false • 2015-06-01 12:30:28.549 INFO 1 com.compuware.cjp.agent.timer.Sampler.<init> -Minimum Java version to use OperatingSystemMXBean: 1.7.0-20150313. Current Java version is: 1.7.0-20141219	Process CPU time is captured using the OperatingSystemMXBean, which requires a version. If Java is not at the required version for Java Performance Agent's JNI functions by setting the tjp.measurement.jni.processcp property to true. In addition, the tjp.measurement.jni.processcputime.every.n property, which sets the frequency of sampli every fifth sample, needs to be set. 0 indicat
• 2015-05-29 22:12:03.098 INFO 1 com.compuware.cjp.agent.timer.Sampler.<init> - Minimum Java version to use OperatingSystemMXBean: 1.7.0- 20150313. Current Java version is: 1.7.0-20141219 • 2015-05-29 22:12:03.113 INFO 1 com.compuware.cjp.agent.timer.Sampler.init - JNI functions will be used capture process CPU time every 10 samples	Process CPU time is currently set to be capt the Topaz for Java Performance Agent's JN but be aware that it is set to capture every te Alternatively, to capture CPU time with Operatin Java needs to be updated to the required version.
• 2015-05-29 22:21:05.750 INFO 1 com.compuware.cjp.agent.timer.Sampler.<init> -JNI functions will not be used capture process CPU time, because tjp.measurement.jni.processcputime.every.nth.sample=0 • 2015-05-29 22:21:05.750 INFO 1 com.compuware.cjp.agent.timer.Sampler.<init> - Minimum Java version to use OperatingSystemMXBean: 1.7.0-20150313. Current Java version is: 1.7.0-20141219	The Topaz for Java Performance Agent's JN needs to be set to a non-zero value so it can capture process CPU time. Alternatively, Ja updated to the required version.

Registration Error Messages

Registration errors are written to the cjp.log, which is located at data/cjp/logs. The following messages may appear:

Message	Meaning	User Action
An agent with that name has already been registered.	An agent is trying to register using an agent name that is already registered.	Rename the agent or stop the agent that is running with the duplicate name.
There are no seat licenses available for this agent.	All of the Topaz for Java Performance licenses have been checked out.	Wait until a license is checked back in.

This agent is not licensed for Topaz for Java Performance.	The LPAR on which the agent is running isn't licensed for Topaz for Java Performance.	Obtain a license for that LPAR.
The version of this agent is not supported.	The agent version isn't supported.	Update the agent to a supported version.
The version of this agent is no longer supported.	The agent version is no longer supported.	Update the agent to a supported version.
A license couldn't be retrieved. Please create a Host Connection and try again.	A license couldn't be retrieved for the agent because there are no HCI connections defined.	Define an HCI connection.