Using Monitors

You can create monitors and use them to collect network information and system performance data during load testing. Monitoring metrics can be overlaid onto load test results to help you determine whether the load passed to the Web server is being handled as designed.

In this section:

Supported Monitors

- SNMP
- Windows monitors (available only on Windows installations of Load Test)
- WebSphere
- WebLogic
- JBoss
- Tomcat
- rstat
- remote
- deployed (e.g., AppDynamics, Dynatrace)
- custom

> **Performance Monitor for Oracle Service Bus (OSB/ALSB)**
>
> A performance monitor for Oracle Service Bus (OSB/ALSB) is available. Contact technical support for details.

Adding Built-In Monitors

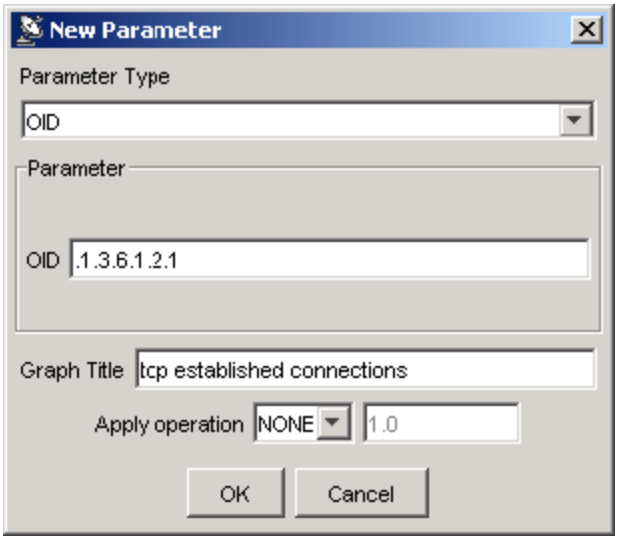
You can add the following built-in monitors.

Adding SNMP Agent Monitors

Load Test can retrieve data based on the provided OID(s) from the SNMP agents of versions 1 and 2c.

To add an SNMP agent monitor:

1. Right-click the **Monitors** node in the Load Test tab and choose **New Monitors** from the shortcut menu. A New Monitors dialog displays.
2. Select **SNMP** from the New Monitors dialog, then click **Finish**. A configuration panel for the new monitor will open in the right panel.
3. Type the SNMP agent's host name in the **Host** field.
4. Type the SNMP agent's community in the **Community** field.
5. Enter a parameter by clicking **New**. The New Parameter dialog opens.

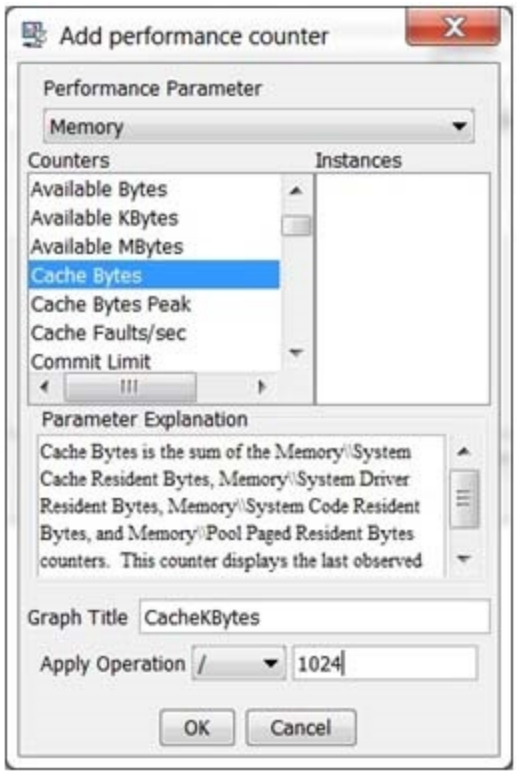


6. Complete the New Parameter dialog as follows:
    - **Parameter Type:** Select **OID** or **Composite**.
    - **Parameter:** Depending on the Parameter Type selected, the Parameter options will vary according to the following:
        - **OID:** Enter the Object ID that identifies a specific parameter within the SNMP agent community. (Only available if OID is selected in the Parameter Type drop-down menu).
        - **Composite:** Select Processor, Memory, Network, or TCP to collect corresponding data.
        - **Counters:** Depending on the selection from the Composite parameter drop-down menu, displays CPU Utilization (%), Memory Utilization (%), and Network Utilization (%); these counters are calculated by querying a set of OIDs.
          If TCP is selected from the Composite parameter drop-down menu, you must select a counter from the Counters box.
        - **Parameter Explanation:** Displays an explanation for the Composite parameter selected.
    - **Graph Title:** Enter a title for the graph that will correspond to the Monitor you are adding. This title will display in Load Test Progress graphs, and in Detailed Report graphs.
    - **Apply operation:** Select to apply simple operations to values returned by an SNMP agent. Use "*" for "multiply", "/" for "divide", and "%" for "percentage of". For example, if the agent returns the value of the allocated memory in bytes but you want to see it in KB, you would apply a "/ 1024" operation by selecting "/" from the drop-down menu and entering "1024" in the adjacent text field. You could then add "Memory in kilobytes" into the **Graph Title** field.
7. Repeat the above step until the Parameters table shows all parameters you want this SNMP agent to monitor.
8. Verify the new monitor by right-clicking the related node in the Load Tests tree's **Monitors> SNMP** branch.

Adding Windows Performance Monitors

To add a Windows performance monitor (functionality which is only available if you have a Windows installation of Load Test):

1. Right-click the Load Test tab's **Monitors** node, then choose **New Monitors** from the shortcut menu. A New Monitors dialog displays.
2. Select **Windows** from the New Monitors dialog, then click **Finish**.
3. Type the perfmon's host name in the **Host** field.
4. Enter a parameter by clicking **New**. The Add performance counter dialog opens.



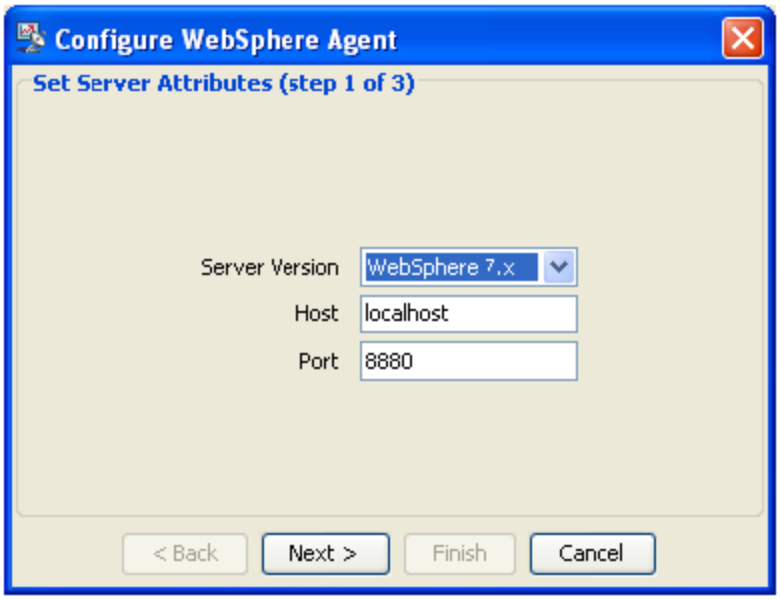
5. Complete the New Parameter dialog as follows:
    - Select the performance counter and instance of the parameter you want to monitor. An explanation of the parameter you select displays in the **Parameter Explanation** box.
    - In the **Graph Title** field, enter a title for the graph that will correspond to the Monitor you are adding. This title will display in Load Test Progress graphs, and in Detailed Report graphs.
    - In the **Apply Operation** sub-panel, select an operation and a value to be applied to the monitor data if needed. For example, by selecting the / (divide) operation and entering `1024` in the operation value field, you could convert the original monitor value from Bytes to KBytes.
6. Repeat the above step until the Parameters table shows all parameters you want this perfmon to monitor.
7. Verify the new monitor by right-clicking the related node in the Load Tests tree's **Monitors> Windows** branch.

Adding WebSphere Monitors

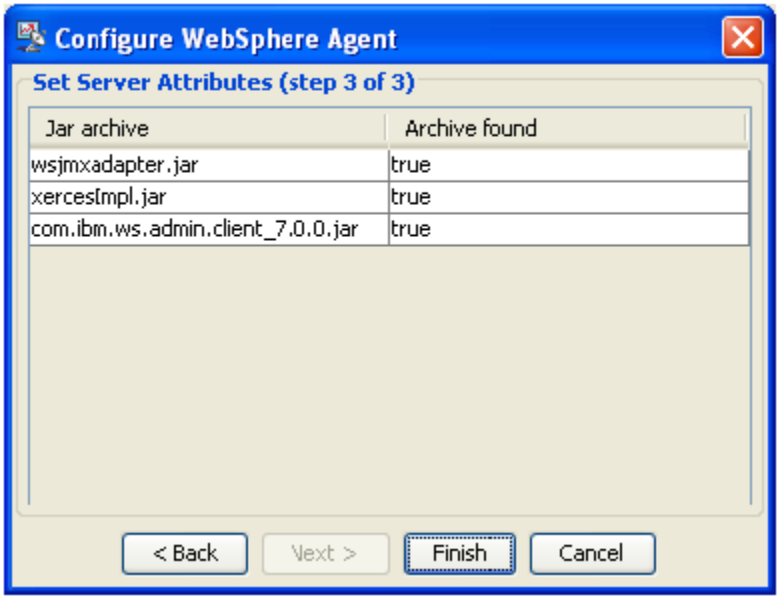
Before adding a WebSphere monitor, go to the **Security> Global security** section of the WebSphere Administrative console and ensure that the **Enable administrative security** check box is unselected.

To add a WebSphere monitor:

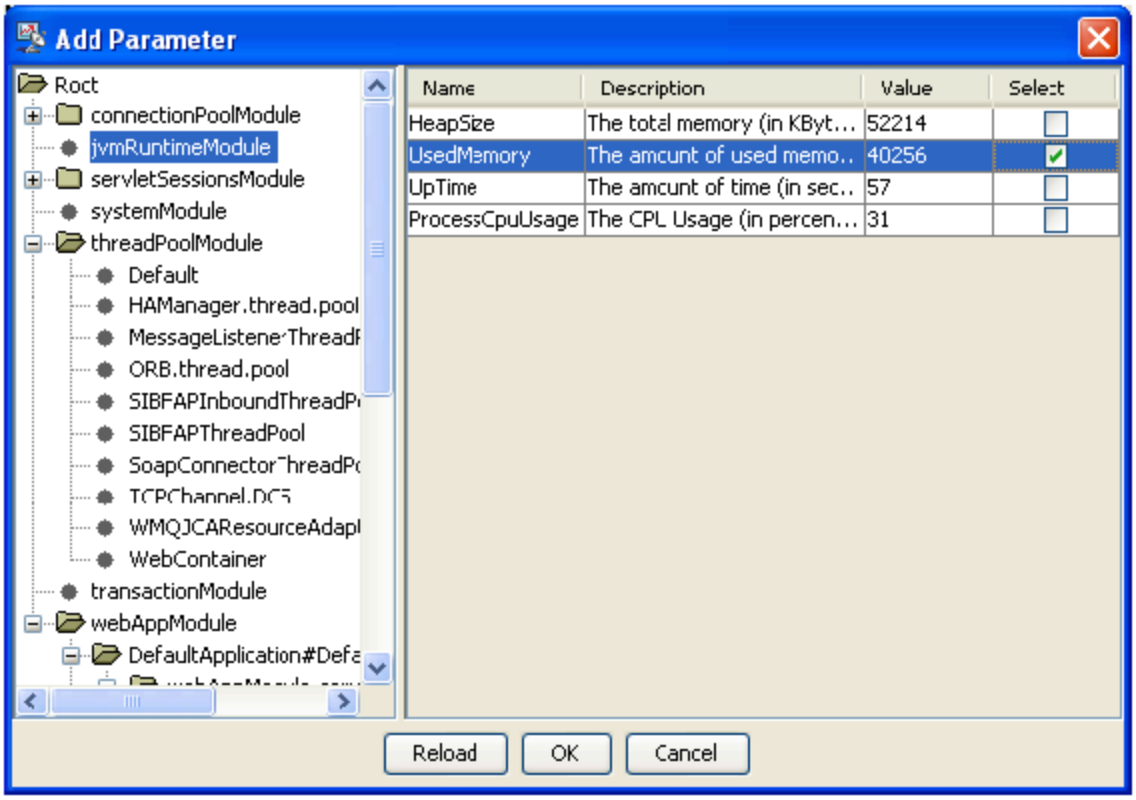
1. Right-click the Load Test tab's **Monitors** node, then choose **New Monitors** from the shortcut menu. A New Monitors dialog displays.
2. Select **WebSphere** from the New Monitors dialog, then click **Finish**.
3. In the Project Configuration panel, click **Configure**. A Configure WebSphere Agent dialog displays.



4. Enter the **Host** name, change the **Port** number if necessary, then click **Next**.
5. In the Configure WebSphere Agent panel, enter or browse to the appropriate Path of the WebSphere installation. For example: C:\IBM\WebSphere.
6. Click **Next**. The Configure WebSphere Agent dialog displays a table of the required Jar files for the server and indicates whether or not the appropriate Jar files were found.



7. In the Configure WebSphere Agent dialog, click **Finish**. The Add Parameter dialog opens.

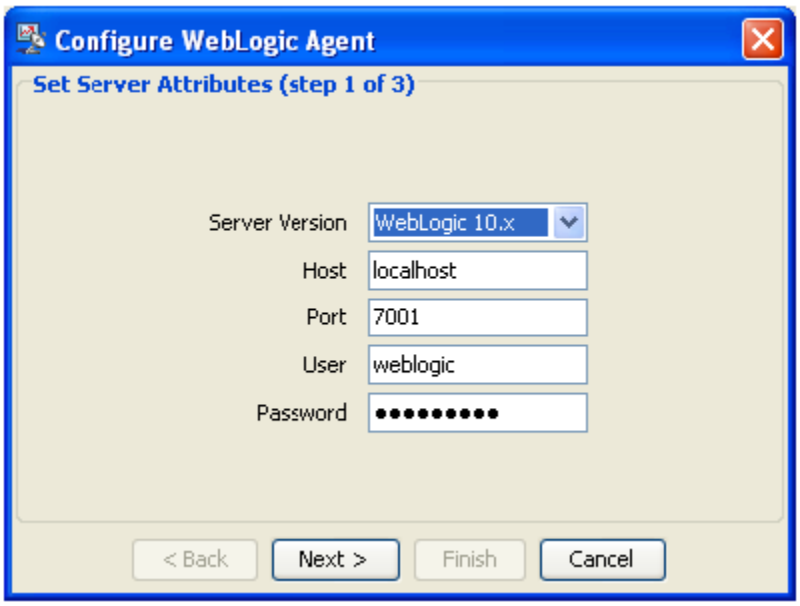


8. Select a Parameter from the left GUI panel of the Add Parameter dialog. The Name, Description, and Value for each instance of the parameter you want to monitor displays in the right GUI panel of the Add Parameter dialog.
9. In **Select** column from the right GUI panel, check the parameter instances that you want to monitor, then click **OK**. The parameters you choose will display in Load Test Progress graphs, and in Detailed Report graphs.

Adding WebLogic Monitors

To add a WebLogic Monitor:

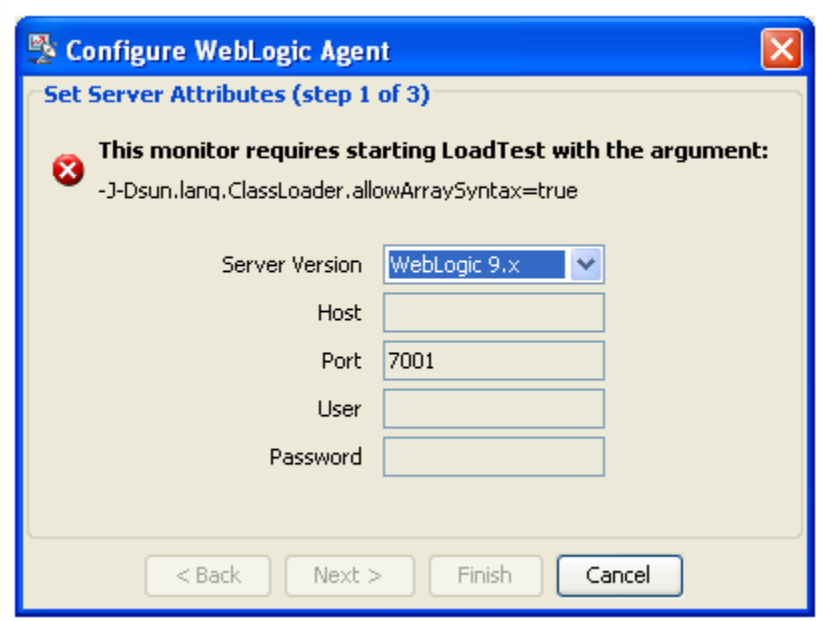
1. Right-click the Load Test tab's **Monitors** node, then choose **New Monitors** from the shortcut menu. A New Monitors dialog displays.
2. Select WebLogic from the New Monitors dialog, then click **Finish**.
3. In the Project Configuration panel, click **Configure**. A Configure WebLogic Agent dialog displays.



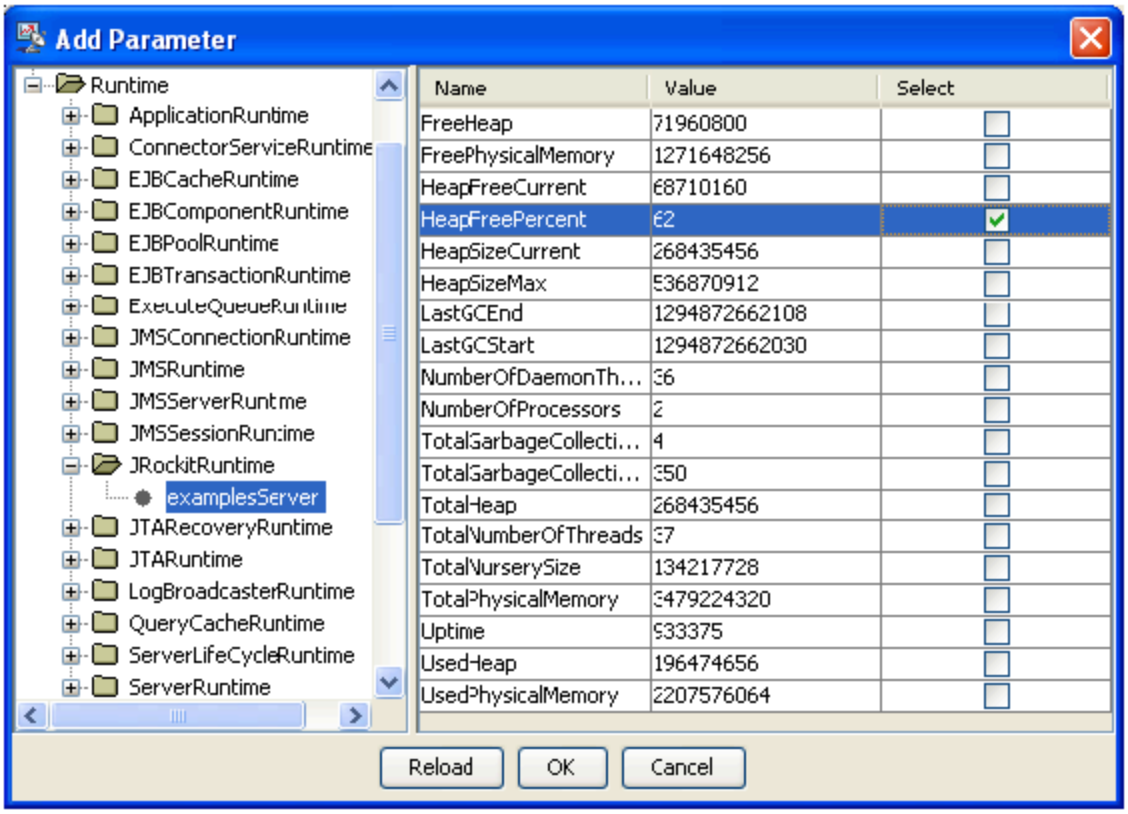
Note

For WebLogic 8.1 SP6 and WebLogic 9.x monitors, Load Test needs to be started with the `-Dsun.lang.ClassLoader.allowArraySyntax=true` option. To do that in the command prompt, navigate to the Load Test installation directory and run the following command:

```
lt -Dsun.lang.ClassLoader.allowArraySyntax=true
```



4. Enter the **Host** name, change the **Port** number if necessary, then click the **Next**.
5. In the Configure WebLogic Agent panel, enter or browse to the appropriate Path of the desired WebLogic implementation on your machine. For example: C:\Oracle\Middleware\wlserver_10.3.
6. Click **Next**. The Configure WebLogic Agent dialog displays a table of the required Jar files for the server and indicates whether or not the appropriate Jar files were found.
7. In the Configure WebLogic Agent dialog, click **Finish**. The Add Parameter dialog opens.

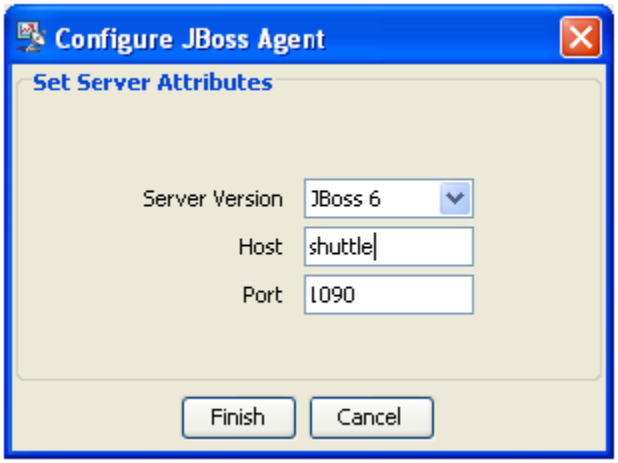


8. Select a Parameter from the left GUI panel of the Add Parameter dialog. The Name, Description, and Value for each instance of the parameter you want to monitor displays in the right GUI panel of the Add Parameter dialog.
9. In **Select** column from the right GUI panel, check the parameter instances that you want to monitor, then click **OK**. The parameters you choose will display in Load Test Progress graphs, and in Detailed Report graphs.

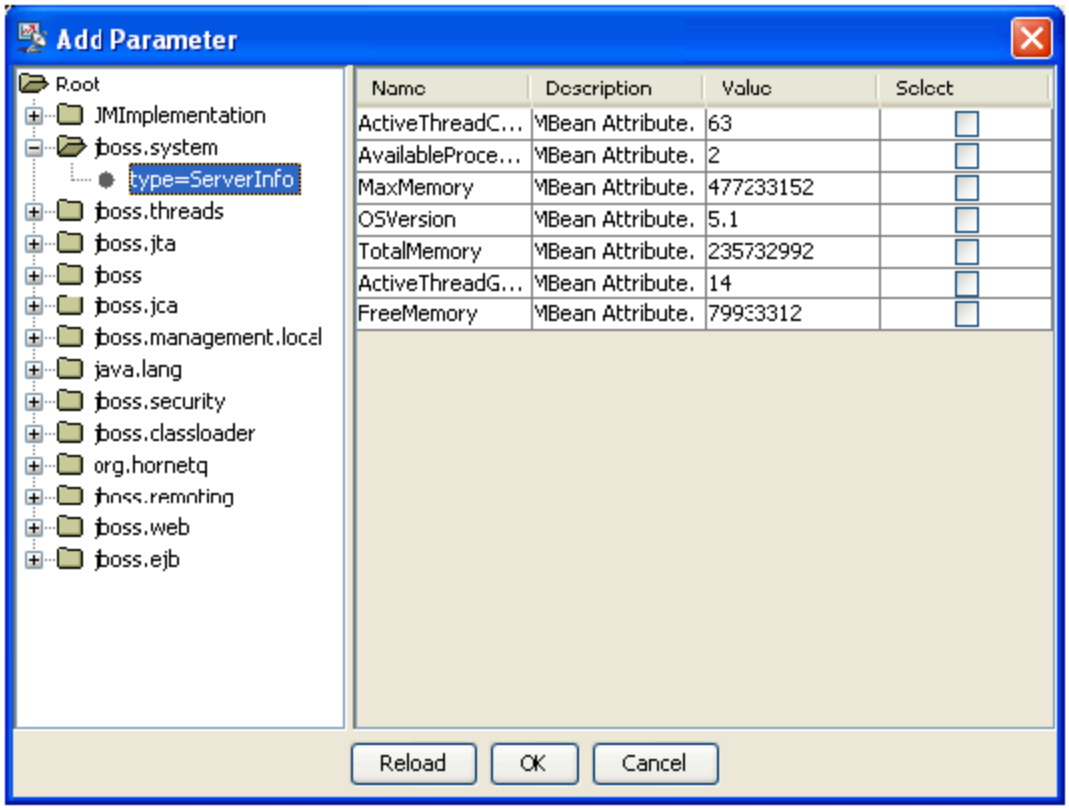
Adding JBoss Monitors

To add a JBoss monitor:

1. Right-click the Load Test tab's **Monitors** node, then choose **New Monitors** from the shortcut menu. A New Monitors dialog displays.
2. Select **JBoss** from the New Monitors dialog, then click **Finish**.
3. In the Project Configuration panel, click **Configure**. A Configure JBoss Agent dialog displays.



4. In the Configure JBoss Agent dialog, enter host name, change port number if necessary, then click **Finish**. The Add Parameter dialog opens.



5. Select a Parameter from the left GUI panel of the Add Parameter dialog. The Name, Description, and Value for each instance of the parameter you want to monitor displays in the right GUI panel of the Add Parameter dialog.
6. In **Select** column from the right GUI panel, check the parameter instances that you want to monitor, then click **OK**. The parameters you choose will display in Load Test Progress graphs, and in Detailed Report graphs.

Adding Tomcat Monitors

Prerequisites

In order to use this monitor, Tomcat should be started with the following Java system properties:

- -Dcom.sun.management.jmxremote
- -Dcom.sun.management.jmxremote.port=7744
- -Dcom.sun.management.jmxremote.ssl=false
- -Dcom.sun.management.jmxremote.authenticate=false

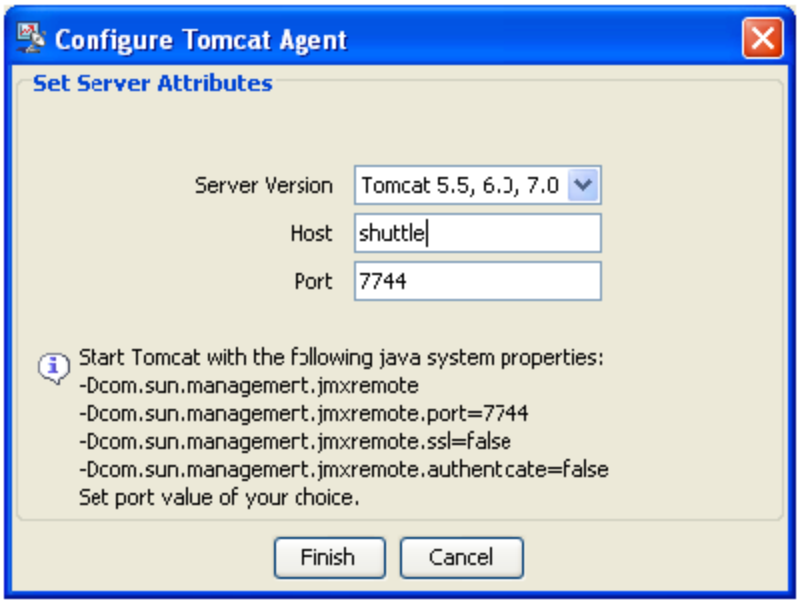
Use the port of your choice with the com.sun.management.jmxremote.port system property.

For example, on Windows you can create a tomcat.bat file in the Tomcat installation "bin" directory and run this batch file to start Tomcat, or you can run this command from the command prompt:

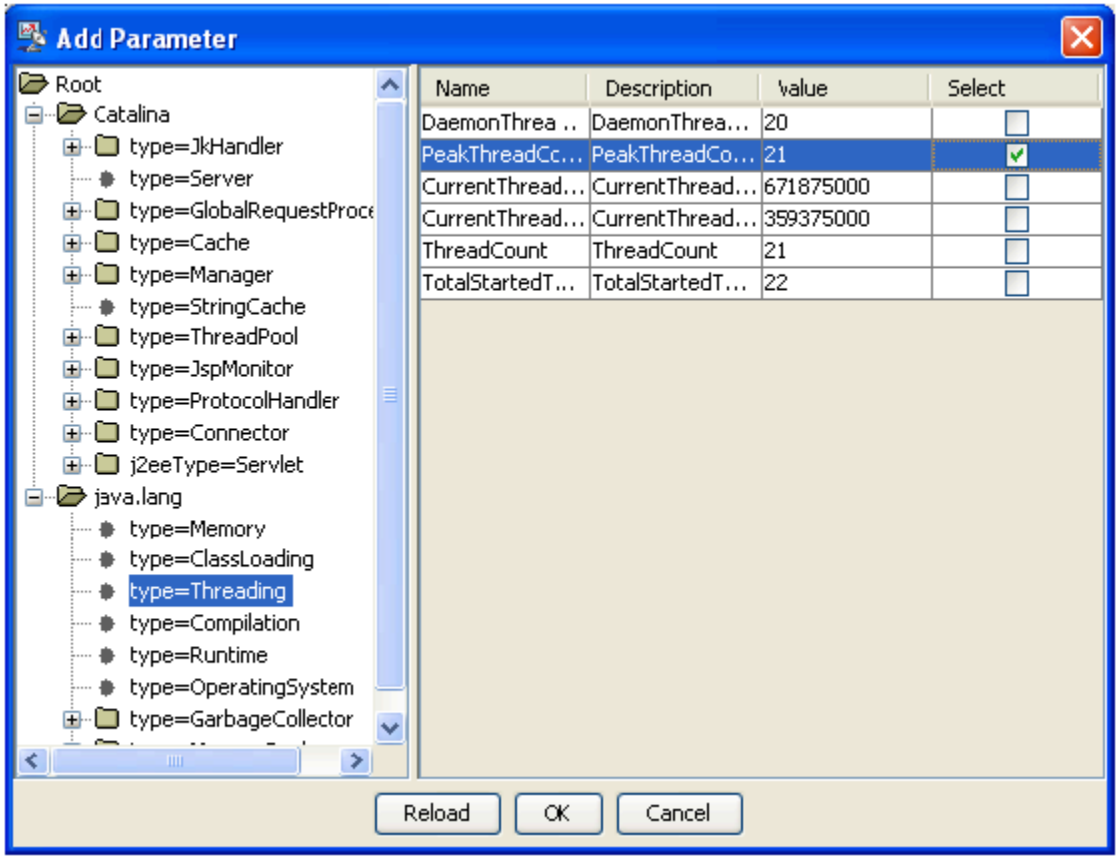
```
java -Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=7744
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=false -jar bootstrap.jar
```

To add a Tomcat monitor:

1. Right-click the Load Test tab's **Monitors** node, then choose **New Monitors** from the shortcut menu. A New Monitors dialog displays.
2. Select **Tomcat** from the New Monitors dialog, then click **Finish**.
3. In the Project Configuration panel, click **Configure**. A Configure Tomcat Agent dialog displays.



4. In the Configure Tomcat Agent dialog, enter the host name, change the port number if necessary, then click **Finish** The Add Parameter dialog opens.



5. Select a Parameter from the left GUI panel of the Add Parameter dialog. The Name, Description, and Value for each instance of the parameter you want to monitor displays in the right GUI panel of the Add Parameter dialog.
6. In **Select** column from the right GUI panel, check the parameter instances that you want to monitor, then click **OK**. The parameters you choose will display in Load Test Progress graphs, and in Detailed Report graphs.

Adding rstat Monitors

To add an rstat monitor:

1. Right-click the Load Test tab's **Monitors** node, then choose **New Monitors** from the shortcut menu. A New Monitors dialog displays.
2. Select **rstat** from the New Monitors dialog, then click **Finish**.

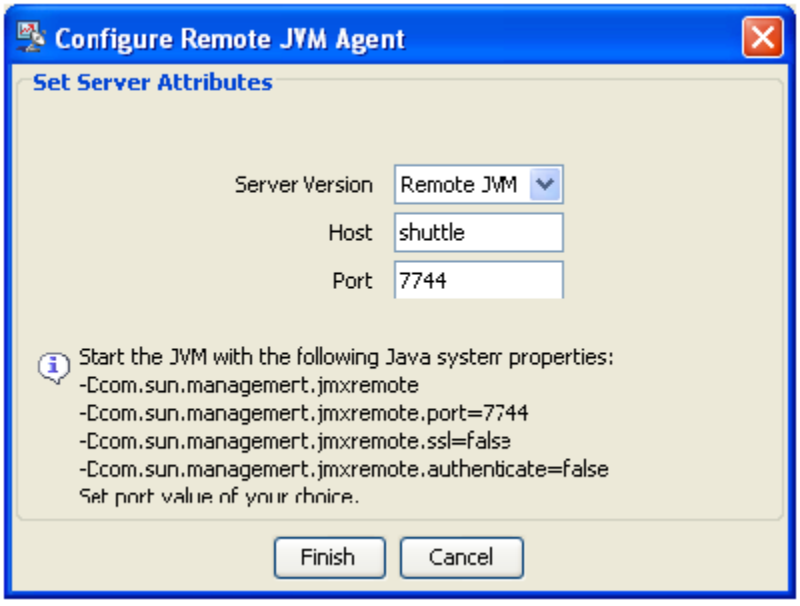
> **Note**
>
> For rstat monitors to be enabled, `djrpc.jar` and `rstatcl.jar` need to be in <SOAtest installation>\eclipse\plugins\com.parasoft.xtest.libs.web_<version>/root and added to the classpath.
>
> The rstatcl.jar is added to <SOAtest_Installation>\<SOAtest_Version>\eclipse\plugins\com.parasoft.xtest.libs.web_<version>\root upon installation.
>
> The djrpc.jar must be purchased and copied to <SOAtest installation>\eclipse\plugins\com.parasoft.xtest.libs.web_<version>/root.

Adding Remote JVM Monitors

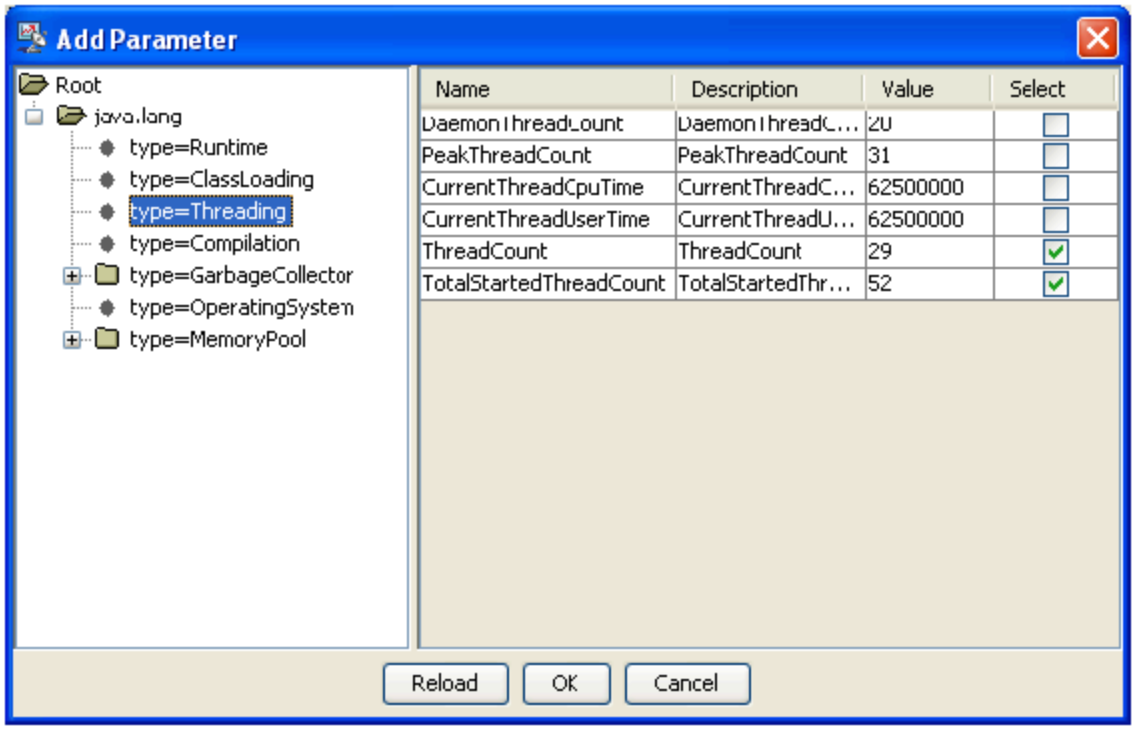
> **Note**
>
> To use this monitor, the JVM should be started with the following Java system properties:
>
> ```
> -Dcom.sun.management.jmxremote
> -Dcom.sun.management.jmxremote.port=7744
> -Dcom.sun.management.jmxremote.ssl=false
> -Dcom.sun.management.jmxremote.authenticate=false
> ```
>
> Use the port of your choice with the `com.sun.management.jmxremote.port system` property.

To add a remote JVM Monitor:

1. Right-click the **Load Tests** tab and choose **Monitors** node, then choose **New Monitors** from the shortcut menu. A New Monitors dialog displays.
2. Select **Remote JVM** from the New Monitors dialog, then click **Finish**.
3. In the Project Configuration panel, click **Configure**. A Configure Remote JVM Agent dialog displays.



4. In the Configure Remote JVM Agent dialog, enter host name and change port number if necessary and press the Finish button. The Add Parameter dialog opens.



5. Select a Parameter from the left GUI panel of the Add Parameter dialog. The Name, Description, and Value for each instance of the parameter you want to monitor displays in the right GUI panel of the Add Parameter dialog.
6. In **Select** column from the right GUI panel, check the parameter instances that you want to monitor, then click **OK**. The parameters you choose will display in Load Test Progress graphs, and in Detailed Report graphs.

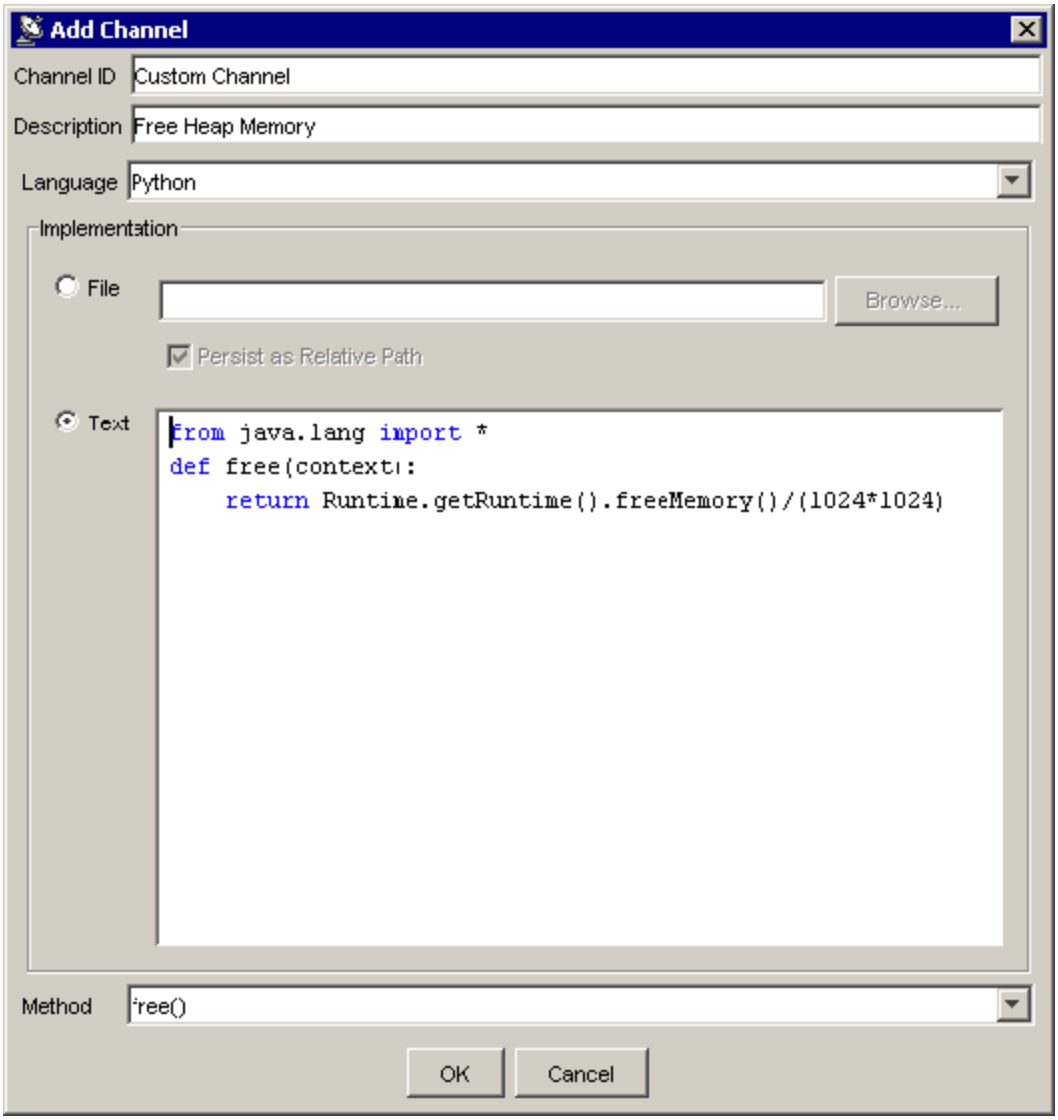
Adding Custom Monitors

Custom monitors can be implemented by JavaScript or Jython scripts, or by Java class methods. This is especially useful if you would like to collect data from sources that are not listed above.

> **Note**
>
> A channel method must return a java.lang.Number type in order to be plotted in Load Test Progress and Detailed Report graphs.

To add a custom monitor:

1. Right-click the Load Test tab's **Monitors** node, then choose **New Monitors** from the shortcut menu. A New Monitors dialog displays.
2. Select **Custom** from the New Monitors dialog, then click **Finish**.
3. Enter a name for the Custom monitor in the **Name** field.
4. Click the **New** button. An **Add Channel** dialog opens.



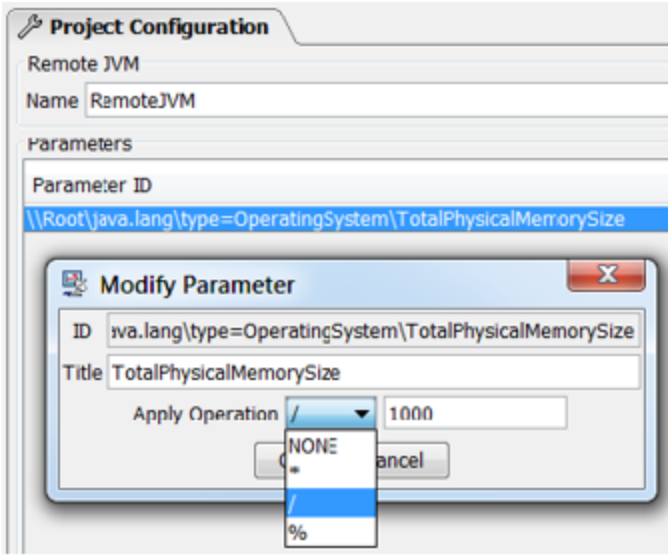
5. Enter a name for the channel in the **Channel ID** field. The name you entered for the Custom Monitor, together with the Channel ID, will comprise the title of the monitor. This title will display in Load Test Progress graphs, and in Detailed Report graphs.
6. (Optional) Enter a short description of the channel in the **Description** field.
7. Select a language from the **Language** menu.
8. Specify the script by doing one of the following:
    - Select the **File** radio button and click the **Browse** button to select the appropriate file.
    - Select the **Text** radio button and enter the script in the text field.
9. Specify the method you want invoked from the **Method** menu.
10. Click **OK**. The new custom channel is added to the Channels table in the right GUI panel.

Modifying Built-In Monitors

After a monitor has been created, you can set or modify the Apply Operation parameter for the following built-in monitor types:

- Windows
- JBoss
- Tomcat
- Remote JVM
- SNMP

To modify the operation, double-click the monitor channel in the Parameters panel of the monitor configuration view and set the appropriate operation type and value.



Adding Deployed Monitors

Load Test allows dynamically adding monitor types to the Load Test installation. Deployed monitors are displayed in the Deployed monitor types table of the Performance Monitors wizard.

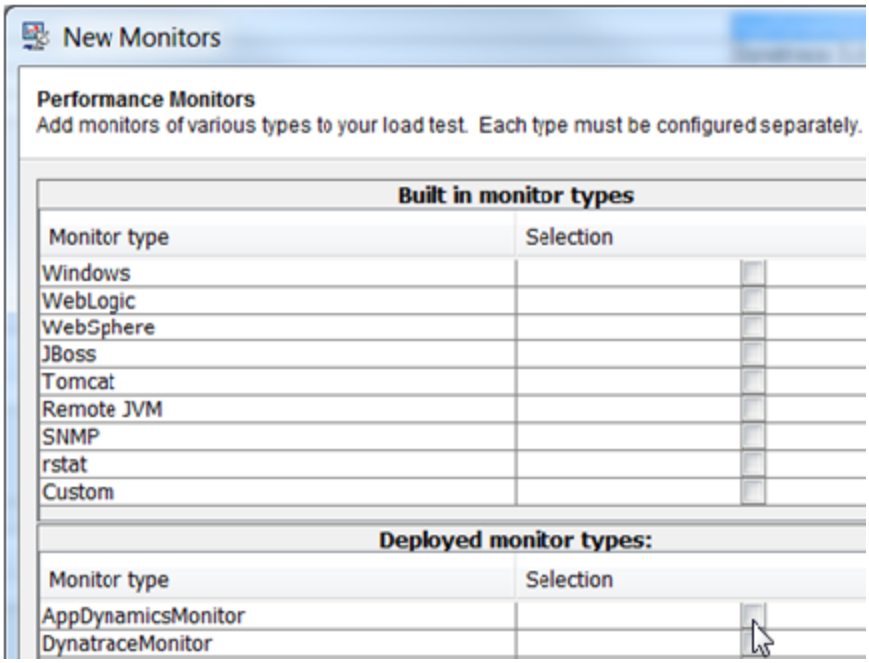
- AppDynamics
- Dynatrace
- JVM Threads

AppDynamics

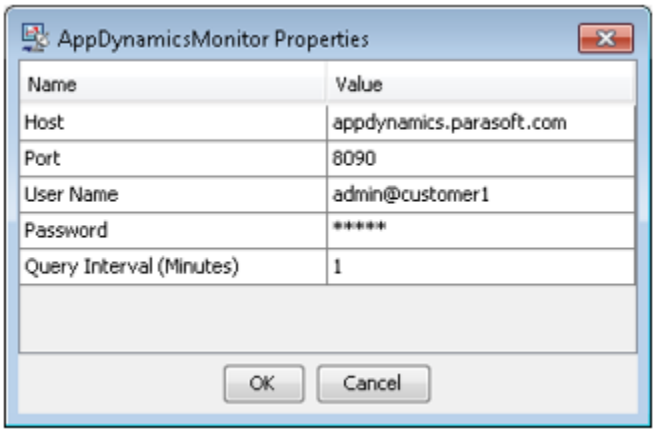
Adding AppDynamics Monitors

To add an AppDynamics monitor:

1. Right-click the Load Test tab's **Monitors** node, then choose **New Monitors** from the shortcut menu. A New Monitors dialog displays.
2. Select **AppDynamicsMonitor** from the New Monitors dialog's Deployed monitor types table, then click **Finish**.



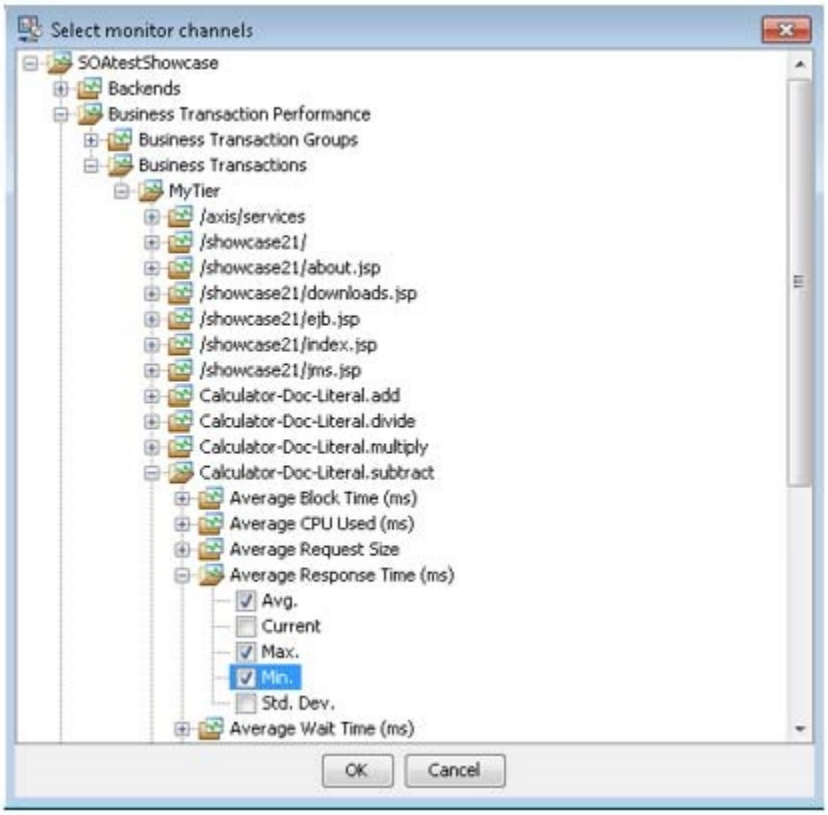
3. In the upper right corner of the Project Configuration panel, click the **Properties** button and configure the AppDynamics properties.



- **Host** – host where the AppDynamics server is installed.
- **Port** – port where AppDynamics is listening for REST calls.
- **User Name** – user name for REST calls authentication.
- **Password** – password for REST calls authentication.
- **Query Interval (Minutes)** – time in minutes over which the monitor values are averaged. The minimum value is 1 minute.

4. Click **OK** to save the property values in the project.
5. To add channels to the AppDynamics monitor:
    1. Click the **New** button in the **Deployed monitor** panel of the Project Configuration tab. The monitor channel selection dialog displays. The top level nodes in monitor channel tree are applications monitored by AppDynamics.
    2. Expand the top level application nodes until you reach the channel leaf nodes with check boxes.
    3. Ensure that the items you want to monitor are checked.

4. Click **OK** to have selected nodes added to the Parameters table of the Deployed monitor configuration panel. The parameters you choose will display in Load Test Progress graphs as well as in Detailed Report graphs.



Tip

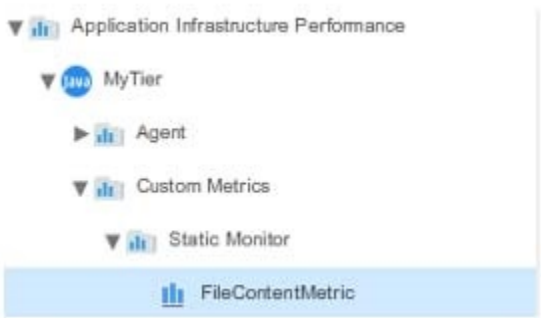
Open the Properties dialog and click the **OK** button to reset the monitor channel selection tree. The monitor channels tree will display the top level application nodes based on the data provided by AppDynamics. The child nodes will be dynamically constructed based on queries sent to AppDynamics as you click the tree's expand icons "[+]".

Monitoring Data Available from AppDynamics Monitoring Extensions

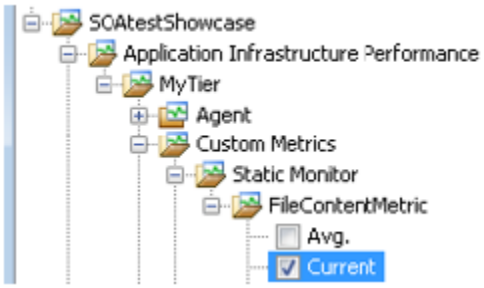
The AppDynamics Exchange site offers a number of Monitoring Extensions. Monitoring Extension data available in AppDynamics Metric Browser will be available in the Load Test AppDynamics monitor.

The screenshots below show the data exposed by the 'Static' Monitoring Extension available from the Exchange in the AppDynamics Metric Browser View and the Load Test AppDynamics monitor channel selection view.

Custom FileContentMetric in AppDynamics:



Custom FileContentMetric in Load Test AppDynamics Monitor:

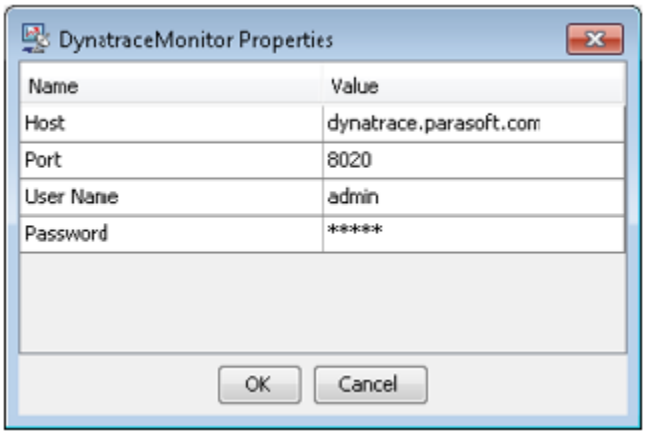


Dynatrace

Adding Dynatrace Monitors

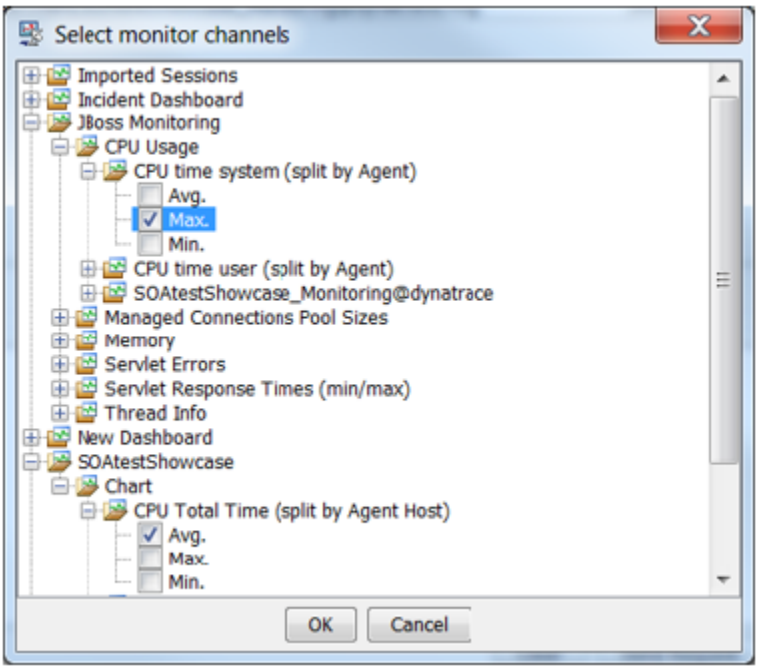
To add a Dynatrace monitor:

1. Right-click the Load Test tab's **Monitors** node, then choose **New Monitors** from the shortcut menu. A New Monitors dialog displays.
2. Select **DynatraceMonitor** from the New Monitors dialog's Deployed monitor types table, then click **Finish**.
3. In the upper right corner of the Project Configuration panel, click the **Properties** button and configure the Dynatrace properties.



   - **Host** – host where the Dynatrace server is installed.
   - **Port** - port where Dynatrace is listening for REST calls. Default port is 8020. Refer to the Dynatrace documentation for more details.
   - **User Name** - user name for REST calls authentication.
   - **Password** - password for REST calls authentication.
4. Click **OK** to save the property values in the project.
5. To add channels to the Dynatrace monitor:
   1. Click the **New** button in the **Deployed monitor** panel of the Project Configuration tab. The monitor channel selection dialog displays.
   2. Expand the top level application nodes until you reach the channel leaf nodes with check boxes.
   3. Ensure that the items you want to monitor are checked.

4. Click **OK** to have selected nodes added to the Parameters table of the Deployed monitor configuration panel. The parameters you choose will display in Load Test Progress graphs as well as in Detailed Report graphs.



The monitor channel selection view tree is structured in the following way:

- Level 1: Dynatrace dashboards
  - Level 2: Chart Dashlets in a dashboard
    - Level 3: Measures in a Chart Dashlet
      - Level 4: Min/Max/Avg vales of the Measure

You can monitor either existing Chart Dashlets of Line Chart type or create new ones. You can add multiple Measures (data series) to a Chart Dashlet. See the Dynatrace vendor documentation for more details.

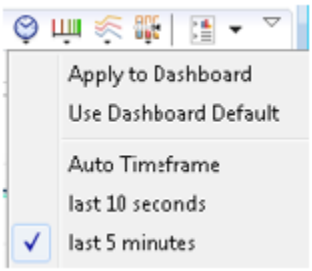
> **Tip**
>
> Open the Properties dialog and click the **OK** button to reset the monitor channel selection tree. The monitor channels tree will display the top level application nodes based on the data provided by Dynatrace. The child nodes will be dynamically constructed based on queries sent to Dynatrace as you click the tree's expand icons "[+]".

Dynatrace REST Query Response Size Considerations

The sizes of the Dynatrace REST responses can be quite large (100KB and more), depending on the Chart and Dashboard configurations.

To minimize the size of the Dynatrace REST query responses:

1. In the charts of the dashboards that are being monitored by Load Test, choose the following settings in the Chart control panel in the top right corner of the Chart view.
   - **Select Timeframe**: set to 'last 5 minutes'.



- **Select Chart Resolution:** set to '10s'

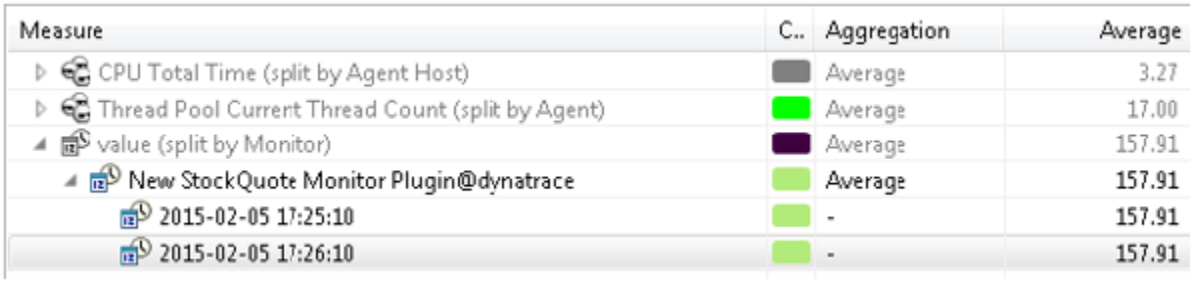
2. To apply these settings to all charts in a dashboard, choose the **Apply to Dashboard** command in the right-click menu shown in screenshots above.
3. To estimate the sizes of responses of REST queries of dashboards that are used by the Load Test Dynatrace monitor, open the query results in a browser using the following URL template (after substituting the host, port and the dashboard names to the ones relevant to your environment): http://dynatrace.parasoft.com:8020/rest/management/dashboard/your_dashboard_name

Monitoring Data Available from Dynatrace Plugins

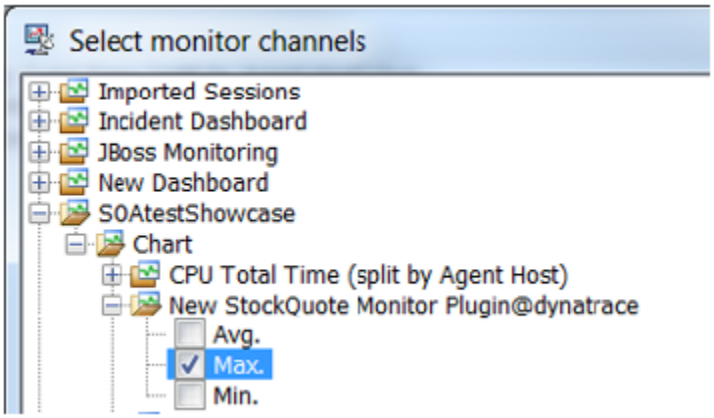
The Dynatrace 'PLUGIN CENTRAL' site and 'Community Plugins and Extensions' site contain a number of plugins, many of which can be used to extend the out-of-the box Dynatrace monitoring capabilities. The Dynatrace plugin data added as a Measure to a Chart Dashlet will be available in the Load Test Dynatrace monitor.

The screenshots below show a StockQuote sample monitor plugin available from the 'Community Plugins and Extensions' site added as Measure in a Dynatrace Chart Dashlet and in Load Test Dynatrace monitor channel selection view.

Custom StockQuote Measure in Dynatrace:



Custom StockQuote Channel in Load Test Dynatrace monitor:



JVM Threads

Adding Remote JVM Threads Monitors

The JVM Threads Monitor allows you to observe threads statistics, as well as record individual thread details of a remote Java application using Java JMX technology.

Enabling JMX Monitoring

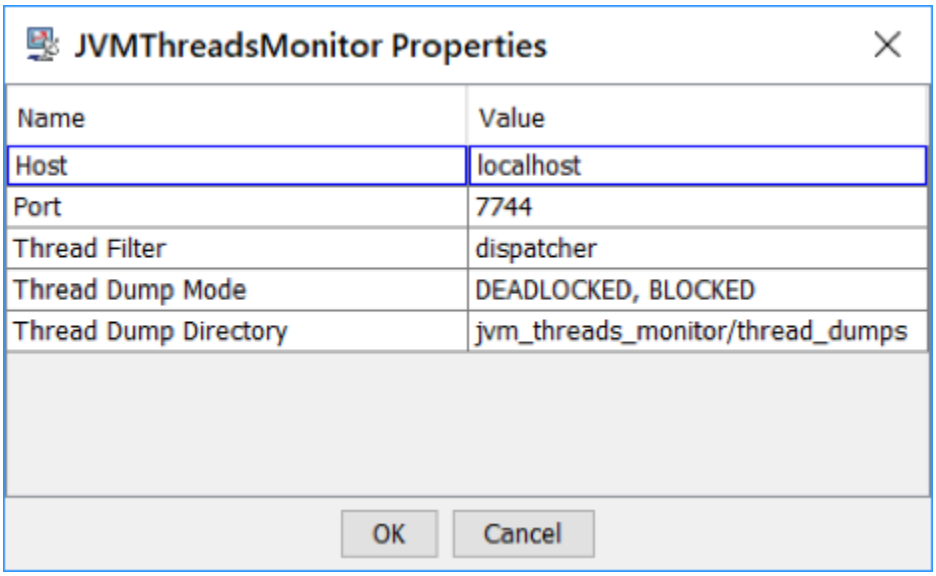
To use this monitor, the JVM should be started with the following Java system properties to enable JMX monitoring:

```
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=7744
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=false
```

Use the port of your choice with the `com.sun.management.jmxremote.port` system property.

To add a Remote JVM Threads Monitor

1. Right-click the **Monitors** node in the Load Test tab and choose **New Monitors** from the shortcut menu.
2. Choose **JVMThreadsMonitor** from the New Monitors dialog's Deployed monitor types table and click **Finish**.
3. In the upper right corner of the Project Configuration panel, click the **Properties** button and configure the JVMThreadsMonitor properties.

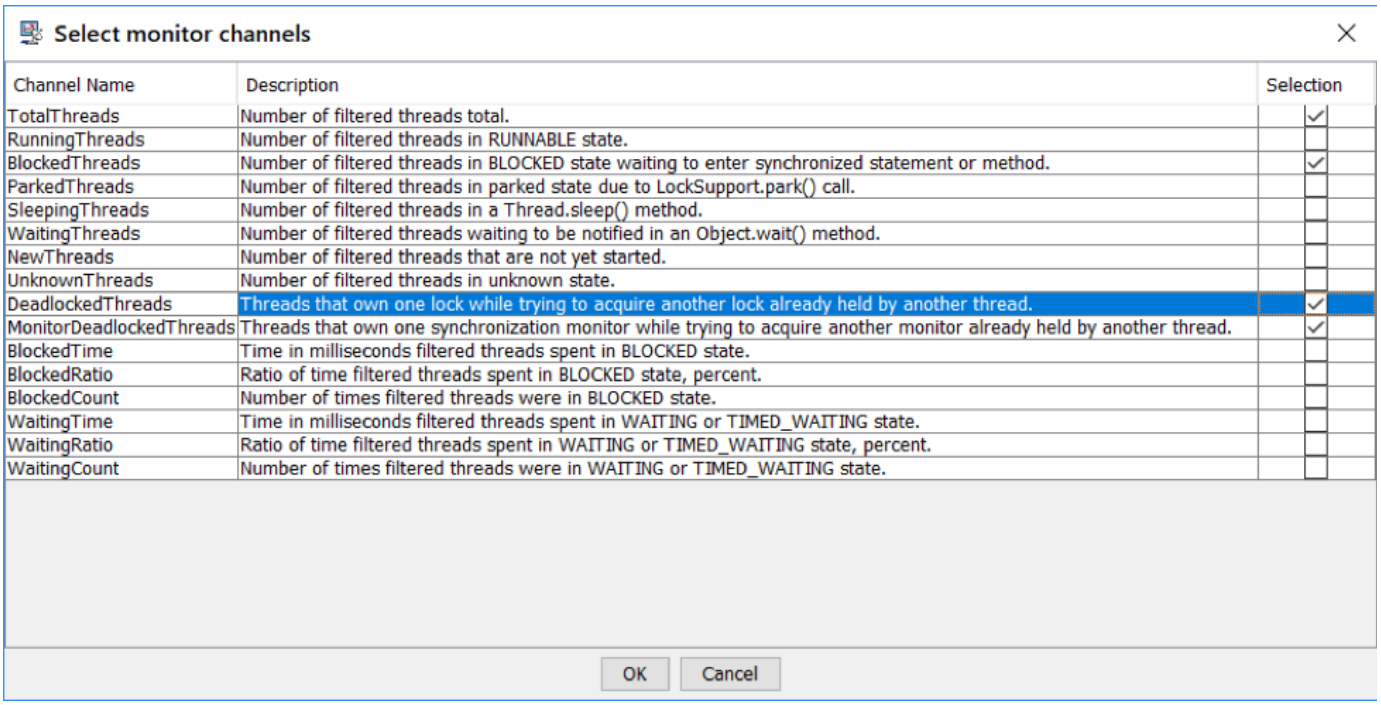


- **Host:** host name where the JVM is running
- **Port:** the JMX port of the JVM (See Enable JVX Monitoring).
- **Thread Filter:** Comma separated list of strings. If a thread name contains any of the strings in the list, the thread will be processed by the monitor. Thread name matching is case insensitive. (The filter is applied after the monitor receives all available thread data from a remote JVM process).
- **Thread Dump Mode:** Comma separated list of thread states to be dumped for further analysis. See Taking Selective Thread Dumps for a detailed description.
- **Thread Dump Directory:** Directory path where threads should be dumped. See Taking Selective Thread Dumps for a detailed description.

The JVM Threads Monitor may affect the performance of your application under test (AUT)

The performance effects associated with adding the JVM Threads Monitor are comparable to using the `jvisualvm` Java performance diagnostic and monitoring tool. The effect should be negligible in most cases, however we recommend evaluating it by comparing key performance indicators of the AUT, such as CPU utilization, Average and Maximum response times, etc., before and after applying the JVM Threads Monitor to your Load Test project.

4. Click **OK** to save the property values in the project.
5. To add channels to the JVM Threads monitor:
    1. Click **New** in the Deployed monitor panel of the Project Configuration tab. The monitor channel selection dialog displays.
    2. Enable the channels you want to monitor. Channel details are available in the Description column of the channel selection dialog.



About Channel Descriptors

TotalThreads	Number of filtered threads.
RunningThreads	Number of filtered threads in RUNNABLE state
BlockedThreads	Number of filtered threads in BLOCKED state waiting to enter synchronized statement or method.
ParkedThreads	Number of filtered threads in parked state due to `LockSupport.park()` call.
SleepingThreads	Number of filtered threads in a `Thread.sleep()` method.
WaitingThreads	Number of filtered threads waiting to be notified in an `Object.wait()` method.
NewThreads	Number of threads that are not yet started.
UnknownThreads	Number of threads in unknown state.
DeadlockedThreads	Threads that own one lock while trying to acquire another lock already held by another thread. This channel finds locks involving both object monitors and ownable synchronizers, such as `java.util.concurrent.locks.ReentrantLock` and `java.util.concurrent.locks.ReentrantReadWriteLock`
MonitorDeadlockedThreads	Threads that own one synchronization monitor while trying to acquire another monitor already held by another thread. This method finds deadlocks involving only object monitors.
BlockedTime	Approximate time in milliseconds filtered threads spent in BLOCKED state during the Load Test data collection interval.
BlockedRatio	Approximate ratio of time filtered threads spent in BLOCKED state measured in percent.
BlockedCount	Number of times filtered threads were in BLOCKED state.
WaitedTime	Approximate time in milliseconds filtered threads spent in WAITING or TIMED_WAITING state.
WaitedRatio	Approximate ratio of time filtered threads spent in WAITING or TIMED_WAITING state measured in percent.

WaitedCount	Number of times filtered threads were in WAITING or TIMED_WAITING state.

All data returned by monitor channels is related to the last Load Test data collection interval.

The NEW, RUNNABLE, BLOCKED, WAITING, TIMED_WAITING, TERMINATED are the states from the `State` enumeration of the `java.lang.Thread` class.

For details about JVM thread states, see the Java documentation for relevant Java classes:

- java.lang.management.ThreadInfo:
  https://docs.oracle.com/javase/8/docs/api/java/lang/management/ThreadInfo.html
- java.lang.managementThreadJMXBean:
  https://docs.oracle.com/javase/8/docs/api/java/lang/management/ThreadMXBean.html

Upon initialization, the JMX Threads Monitor checks if thread contention monitoring is enabled on a remote JVM. If contention monitoring is disabled, the JMX Threads Monitor attempts to enable it. If this attempt fails, the BlockedTime, BlockedRatio, WaitedTime and WaitedRatio channels will return zero values.

Taking Selective Thread Dumps

The JMX Threads Monitor allows you to take selective dumps of a remote application's threads and save them to disk for further analysis. This feature helps you catch undesired thread states that appear sporadically or record thread states during the automated load test runs.

The JMX Threads Monitor will only dump threads when a load test is running. Thread dumps will not be taken when clicking **Send Request** in the monitor configuration view.

Set the following monitor properties to configure selective thread dumps.

Thread Filter	Only threads matching the Thread Filter will be dumped.
Thread Dump Mode	The following values are available: • ALL: Dump all threads • NONE or empty: Do not create thread dumps • DEADLOCKED: Dumps deadlocked threads • BLOCKED: Dump blocked threads • PARKED: Dump parked threads No thread dumps will be taken if this property is not configured (empty). Enter a comma separated list of values to configure dumps of multiple thread states: DEADLOCKED, BLOCKED, PARKED
Thread Dump Directory	You can enter either an absolute path or a path relative to the following directory in SOAtest/LoadTest installation: ${SOATEST_INSTALL_DIR}\eclipse\plugins\com.parasoft.xtest.libs.web_version\root No thread dumps will be taken if this property is not configured (empty). The path provided in this setting will be used as a thread dump root (see Examining Thread Dumps).

Examining Thread Dumps

The path provided in the Thread Dump Directory setting will be used as a thread dump root. The JVM Threads Monitor will create a separate project thread dump directory under the thread dump root for each load test run. These project thread dump directories will have the following name pattern: PROJECTNAME_YYYY-MM-DD_hh-mm-ss.

- **PROJECTNAME**: the name of the LoadTest project.
- **YYYY**: current year
- **MM**: current month
- **DD**: current date
- **hh**: hour of the day in 24 hour format
- **mm**: minute

- **ss**: second

This format allows you to clearly see which Load Test project a thread dump is related to and when it was created. The PROJECTNAME_YYYY-MM-DD_hh-mm-ss directory will not be created if the monitor does not find threads to dump that match the threads dump configuration during a load test run.

The monitor will create thread dump files with the following name pattern inside the project thread dump directory: S..S_hh-mm-ss.txt

- S..S: Elapsed Time in seconds since the beginning of the load test run when the threads dump was taken.
- hh-mm-ss: System Time when the thread dump was taken.
  - hh: hour of the day in a 24 hour format
  - mm: minute
  - ss: second

Each thread dump file will contain one or more thread stack traces taken by the monitor. A thread stack trace will have the following format:

```
THREAD_STATE thread: THREAD_NAME
      thread_stack_entries
```

Each BLOCKED thread will be followed by a stack trace of the thread that is blocking it. A BLOCKED thread output will have the following format:

```
THREAD_STATE thread: THREAD_NAME
      thread_stack_entries
      ------------------
      Blocked by:
      BLOCKING_THREAD_STATE thread: THREAD_NAME
            blocking_thread_stack_entries
```

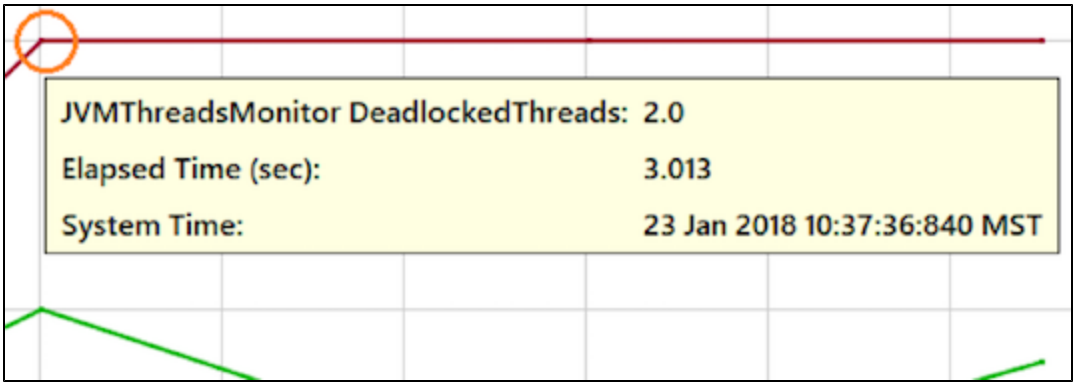
Example:

```
BLOCKED thread: Clinet Rec: LTPub_0.4104694862292172
    com.acme.connections.ClientComms.isDisconnecting:561
    ...
    java.util.concurrent.ThreadPoolExecutor$Worker.run:617
    java.lang.Thread.run:745
    ------------------
    Blocked by:
    PARKED thread: acme_pool-2-UserCodeRunnable-135
        sun.misc.Unsafe.park:-2
        java.util.concurrent.locks.LockSupport.parkNanos:215
        java.util.concurrent.locks.
AbstractQueuedSynchronizer$ConditionObject.awaitNanos:2078
        java.util.concurrent.ThreadPoolExecutor.awaitTermination:1465
        ...
```

Matching Thread Dumps to Graph Points in the Load Test Report

To find a thread dump for a particular point in the Load Test report graph follow there steps:

1. Hover the mouse cursor over a point in the Load Test report graph
2. Use either the Elapsed Time or the System Time in the tooltip to find a dump file with the closest Elapsed or System time.



For example, the 3_10-37-37.txt file will be the thread dump file for the graph point highlighted in the screenshot.

Querying Monitors

Once some parameters have been added to a monitor, they can be queried.

To send a query:

1. Select the Load Tests tree node that represents the monitor you want to query.
2. Click the **Send Request** button in the Response View area of the monitor's configuration panel.

Query results will be displayed in the Response View area of the monitor's configuration panel.

Verifying Monitor Status

The Load Tests tree uses color bubbles to indicate the status of each monitor. The following colors are used:

- White: Unchecked.
- Green: All parameters are available.
- Yellow: Some parameters are either unavailable, or their values are not numeric.
- Red: Monitor is unavailable or unreachable.

To verify the current state of one or more monitors:

1. Right-click the Load Tests tree node that represents the monitor(s) you want to verify (either the **Monitors** node, the **Windows** node, the **SNMP** node, the **Custom** node, or the node that represents an individual monitor).
2. Choose **Verify** from the shortcut menu.

Monitoring Behavior During Test Suite Load Testing

The monitors that you have added to the Load Tests tree will be used during test suite load testing.

During a load test that uses monitors, Load Test adds a bar for each monitor within the Graph tab of the Load Test Progress panel. The Graph tab displays data collected from the monitors. The names of these bars are based on the monitors' Graph Title values, not by OID numbers or parameter IDs.

Accessing Monitor Results

Monitor data collected during load testing is saved in the Detailed Report, where it can be viewed together with and correlated to the default data (such as "Number of Virtual Users", etc.)