Exercise 16.Using the performance monitoring tools

What this exercise is about

In this exercise, you use the performance tools available in WebSphere Application Server to monitor various application and server resources in real time, and generate tuning advice for performance.

What you should be able to do

At the end of this exercise, you should be able to:

- Enable various levels of Performance Monitoring Infrastructure (PMI) statistics for an application server
- Monitor an application server by using Tivoli Performance Viewer
- Configure user settings for Tivoli Performance Viewer
- Examine summary reports and performance modules in Tivoli Performance Viewer
- View performance messages from the Tivoli Performance Viewer Advisor
- Enable and configure the Request Metrics tool
- View Request Metrics messages in the standard logs of an application server
- Configure IBM Tivoli Composite Application Manager for WebSphere Application Server collector for an application server
- View IBM Tivoli Composite Application Manager application performance statistics by using Tivoli Performance Viewer

Introduction

WebSphere Application Server offers a collection of tools to monitor and help tune the runtime environment, including Tivoli Performance Viewer, the Tivoli Performance Viewer advisor, and Request Metrics.

The Tivoli Performance Viewer is the user interface for monitoring the performance of application servers, servlets, and other resources in the environment. It is integrated into the administrative console and can be used for a number of tasks, including viewing real-time performance data, gauging the load on servers over time, and evaluating the efficiency of resource allocations.

The Tivoli Performance Viewer advisor provides advice to help tune systems for optimal performance and recommendations to remedy inefficient server resource settings. It generates advice that is based on data that the Performance Monitoring Infrastructure (PMI) collects.

Request metrics allow you to track the response time of the individual components that a transaction traverses, providing you with an in-depth understanding of the application flow that satisfies the user request.

Requirements

This exercise requires at least one application server that runs the PlantsByWebSphere application and the Default application.

.

Exercise instructions

Preface

In the first part of this exercise, you use Tivoli Performance Viewer to monitor applications that run on WebSphere Application Server.

When the Performance Viewer is running in a Network Deployment environment, the data is collected at each of the nodes and stored in memory at the node agent. Data is then viewed from the deployment manager. With this architecture, the monitoring is distributed among the nodes.

Section 1: Resetting the WebSphere environment

Note

To reset your WebSphere environment, read **Appendix A** for instructions on how to complete this procedure.

Section 2: Verifying the environment

If you changed the maximum heap size for the deployment manager to a lower value to conserve system resources, you must restore the maximum heap size to the default setting. More memory is required because the Tivoli Performance Viewer runs inside the deployment manager.

__ 1. Verify that the deployment manager, the node agent, and WebSphere Application Server server1 in profile1 are all running.

__ 2. Start an administrative console and make sure that you are logged in as: `wasadmin`

Information

You can stop server2 and its node agent to free up physical memory if the class environment has limited memory. As a rule of thumb, if your computer has less than 2 GB of memory, it can be beneficial to run only server1 for this lab.

Section 3: Enabling performance monitoring and setting user preferences

Before the Tivoli Performance Viewer can begin monitoring data, the performance monitoring service must be started. The monitoring service is turned on by default on the application server.

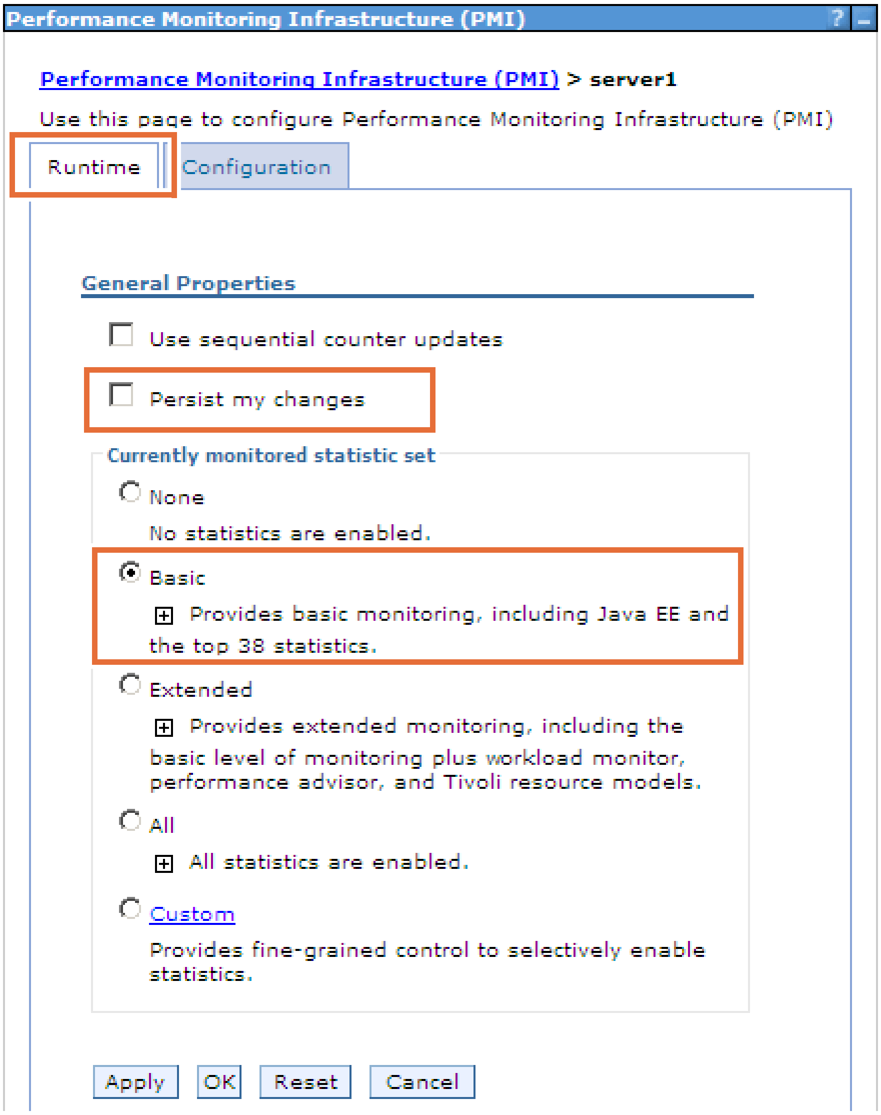
__ 1. Verify that the Performance Monitoring Infrastructure (PMI) is turned on for server1.

__ a. In the administrative console navigation tree, click **Monitoring and Tuning > Performance Monitoring Infrastructure (PMI)**. In the workspace area on the right pane of the administrative console, click **server1**.

__ b. On the Configuration tab, make sure that **Enable Performance Monitoring Infrastructure (PMI)** is already selected.

__ c. In the Currently monitored statistic set area, notice that the **Basic** level is selected. Basic is the default monitoring level setting.

__ d. Click the **Runtime** tab and verify that **Basic** is selected (default statistic set).



Performance Monitoring Infrastructure (PMI)

Performance Monitoring Infrastructure (PMI) > server1

Use this page to configure Performance Monitoring Infrastructure (PMI)

General Properties

□ Use sequential counter updates

□ Persist my changes

Currently monitored statistic set

○ None

No statistics are enabled.

● Basic

⊞ Provides basic monitoring, including Java EE and the top 38 statistics.

○ Extended

⊞ Provides extended monitoring, including the basic level of monitoring plus workload monitor, performance advisor, and Tivoli resource models.

○ All

⊞ All statistics are enabled.

○ Custom

Provides fine-grained control to selectively enable statistics.

__ e. Click **Cancel**.

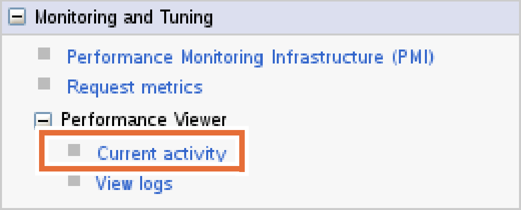
Information

The **Runtime** tab allows you to change the monitoring settings without restarting the server. The new settings are applied immediately after clicking **OK**. If the **Persist my**

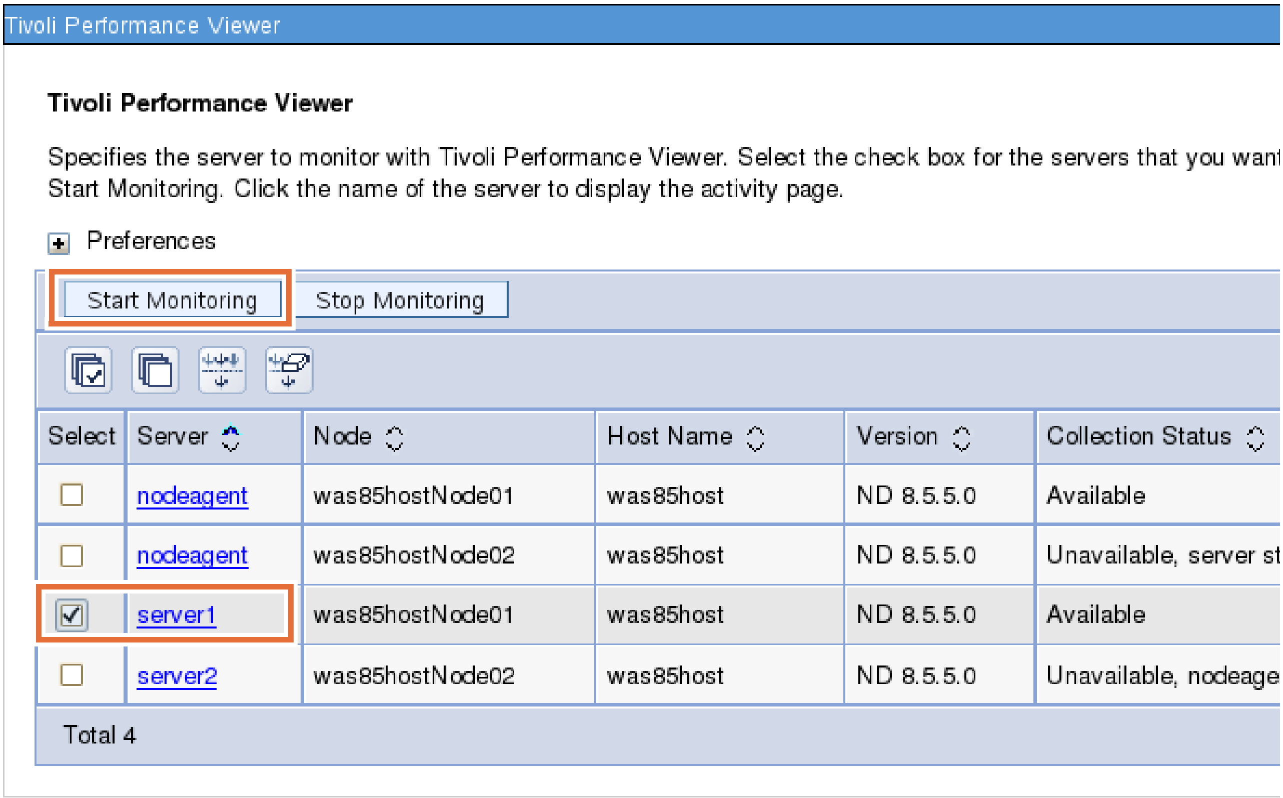
changes check box is selected, the runtime settings are saved and are shown in the **Configuration** tab. This option allows you to change the PMI settings, and persist these changes for the next time the server is started.

__ 2. To get more frequent data collections in the Tivoli Performance Viewer, change the **Monitoring Refresh Rate** to 20 seconds.

__ a. In the administrative console, under Monitoring and Tuning, expand **Performance Viewer** and click **Current activity**.



__ b. On the right pane, select the check box for **server1** and start the monitoring process for this server by clicking **Start Monitoring**.



Tivoli Performance Viewer

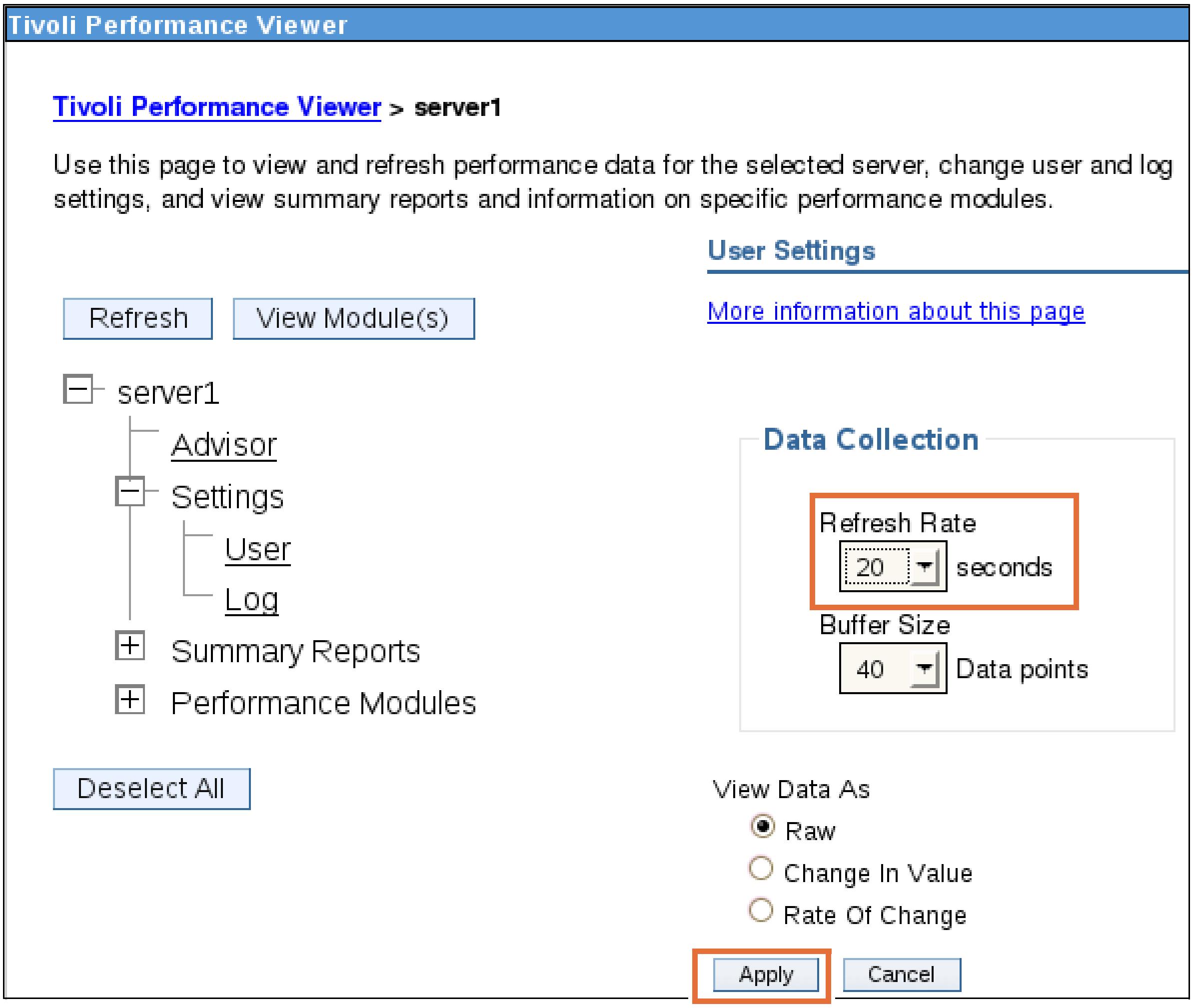
Specifies the server to monitor with Tivoli Performance Viewer. Select the check box for the servers that you want
Start Monitoring. Click the name of the server to display the activity page.

Select	Server	Node	Host Name	Version	Collection Status
□	nodeagent	was85hostNode01	was85host	ND 8.5.5.0	Available
□	nodeagent	was85hostNode02	was85host	ND 8.5.5.0	Unavailable, server st
☑	server1	was85hostNode01	was85host	ND 8.5.5.0	Available
□	server2	was85hostNode02	was85host	ND 8.5.5.0	Unavailable, nodeage

Total 4

__ c. Click the **server1** link to view its current activity. If the tree for server1 is collapsed, click **[+]** next to server1 to expand it. Expand **Settings** and click **User**.

__ d. In the User Settings panel, change the Data Collection **Refresh Rate** to **20** seconds. Click **Apply**.



Tivoli Performance Viewer

Tivoli Performance Viewer > server1

Use this page to view and refresh performance data for the selected server, change user and log settings, and view summary reports and information on specific performance modules.

User Settings

More information about this page

Data Collection

Refresh Rate 20 seconds

Buffer Size 40 Data points

View Data As

● Raw

○ Change In Value

○ Rate Of Change

Section 4: Viewing servlet and web applications module data

In this section, you use the Tivoli Performance Viewer to generate and view performance metrics.

__ 1. Open a new browser and start the **Snoop** servlet by entering the web address:

```
http://was85host:9080/snoop
```

If application security is on, you are prompted for a user ID and password. Log in as `wasadmin` if necessary.
This URL runs the Snoop servlet, which is part of the Default application, and shows a page with various information about the servlet. A servlet must be loaded in order for data collection to take place. Leave the browser window open, as you are going to return here soon.

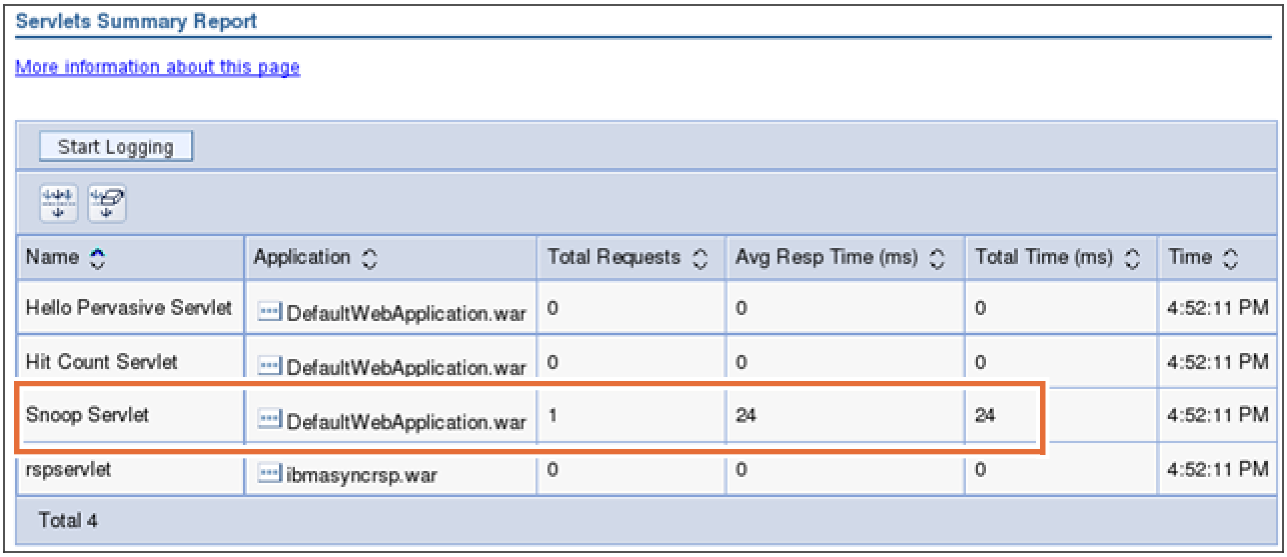
__ 2. Using the administrative console, go to the Tivoli Performance Viewer and monitor server1.

__ a. Click **Monitoring and Tuning > Performance Viewer > Current activity > server1**.

__ b. Expand **Summary Reports** and select **Servlets**. In the Servlets Summary Report pane, locate the entry for the **Snoop servlet**, which is part of the `DefaultWebApplication.war` file.

Information

You might need to go to the next page of the report to find the listing for the Snoop servlet. Use the arrows at the bottom of the page to go to the different pages of the report. As an alternative, you can use filters to reduce the amount of information that is shown, or sort by application name or another attribute.



Servlets Summary Report

More information about this page

Start Logging

Name	Application	Total Requests	Avg Resp Time (ms)	Total Time (ms)	Time
Hello Pervasive Servlet	DefaultWebApplication.war	0	0	0	4:52:11 PM
Hit Count Servlet	DefaultWebApplication.war	0	0	0	4:52:11 PM
Snoop Servlet	DefaultWebApplication.war	1	24	24	4:52:11 PM
rspservlet	ibmasyncrsp.war	0	0	0	4:52:11 PM
Total 4					

__ c. Notice that there is one request for the Snoop servlet and an average response time in milliseconds. Record the average response time here: _____________

__ d. Go back to the browser and reload the page several times.

__ e. Under the **Summary Reports**, click **Servlets** again to refresh the view. Look at the Servlets report again. What is the average response time now?____________
Is the response time longer or shorter after several requests are processed? (The response time is now shorter because of caching.)

__ 3. Open a new browser and enter the web address for PlantsByWebSphere:

```
http://was85host:9080/PlantsByWebSphere
```

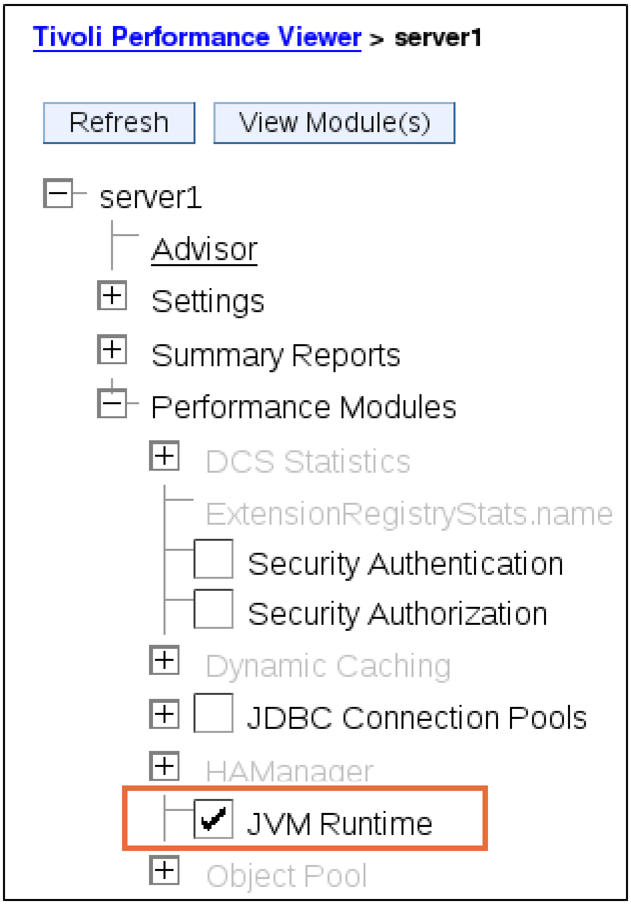
Click the **Flowers** tab and the link for **Lily**.

__ a. Explore more summary reports:

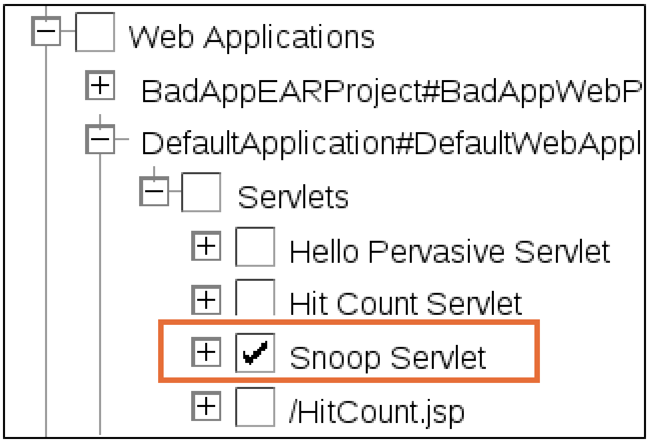
- Are there any Enterprise JavaBeans being monitored? Click **EJBs** under **Summary Reports**. The EJBs Summary Report lists all enterprise beans that are currently running on this server. It also shows the amount of time that is spent in their methods, the number of EJB invocations, and the total time that is spent in each enterprise bean.
- Click **Connection Pool**. The Connection Pool Summary Report lists all data source connections that are defined in the application server and shows their usage over time. The performance data is shown in graph form.
- Click **Thread Pool**. The Thread Pool Summary Report shows the usage of all thread pools in the application server over time.

__ 4. Inside Tivoli Performance Viewer, view the available performance counters for the Snoop servlet and the JVM Runtime module.

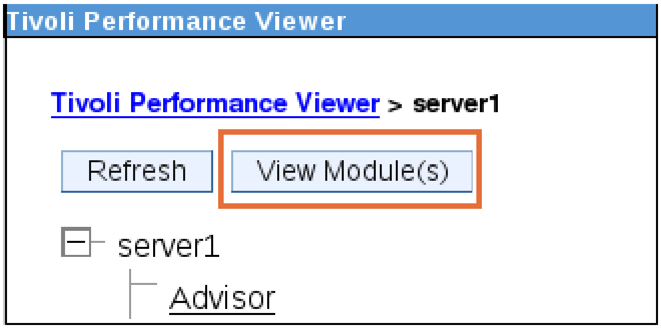
__ a. In Tivoli Performance Viewer, expand **Performance Modules** and select the **JVM Runtime** check box.



__ b. Expand **Web Applications > DefaultApplication#DefaultWebApplication.war > Servlets** and select **Snoop Servlet**.

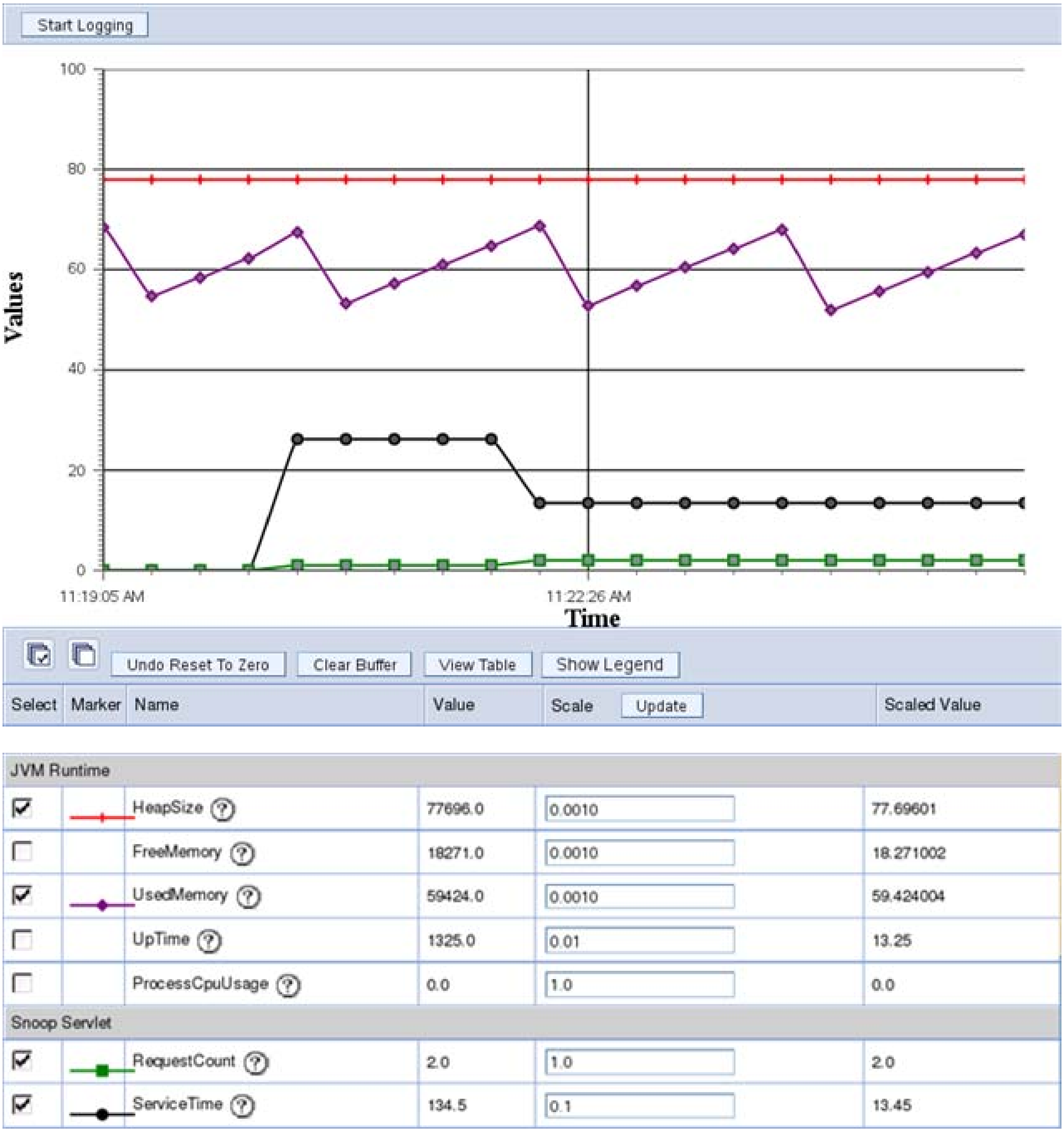


__ c. Click **View Module** at the top of the viewer (if necessary, scroll in your browser to see this screen). You see a table or a graph that shows the monitored data.


Tivoli Performance Viewer
Tivoli Performance Viewer > server1
Refresh
View Module(s)
server1
Advisor

__ d. Open a new browser and start the **Snoop** servlet by entering the web address: `http://was85host:9080/snoop`

__ e. Use the browser to **reload the Snoop servlet** several times by clicking the browser refresh button. Review the changes in the console.



Select	Marker	Name	Value	Scale	Scaled Value
JVM Runtime					
☑	+	HeapSize ?	77696.0	0.0010	77.69601
☐		FreeMemory ?	18271.0	0.0010	18.271002
☑	◆	UsedMemory ?	59424.0	0.0010	59.424004
☐		UpTime ?	1325.0	0.01	13.25
☐		ProcessCpuUsage ?	0.0	1.0	0.0
Snoop Servlet					
☑	■	RequestCount ?	2.0	1.0	2.0
☑	●	ServiceTime ?	134.5	0.1	13.45

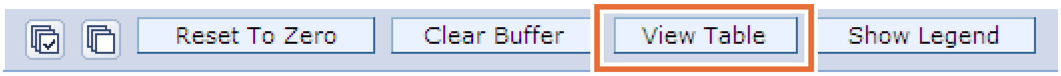
Information

You see changes in the metrics for the Snoop servlet. The request count increases, and the service time changes. Keep in mind that the JVM runtime counters change too. Notice in the screen capture, the FreeMemory metric was cleared and it therefore is not plotted on the graph.

Warning

When viewing graphs and comparing lines, take note of the Scale value. The Performance viewer scales values such that all data points can fit on the graph.

__ f. In the Performance viewer, click **View Table** to switch to a tabular view of the performance data. You can toggle back and forth between the table and graph views by selecting **View Table** or **View Graph**.

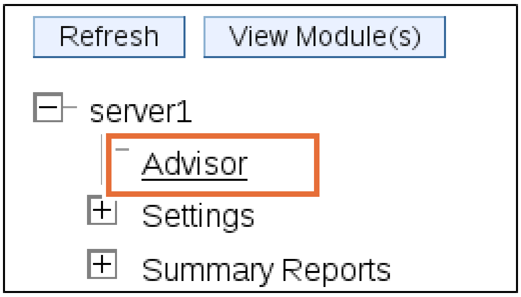


__ g. Reload the Snoop servlet several times and observe the servlet metrics.

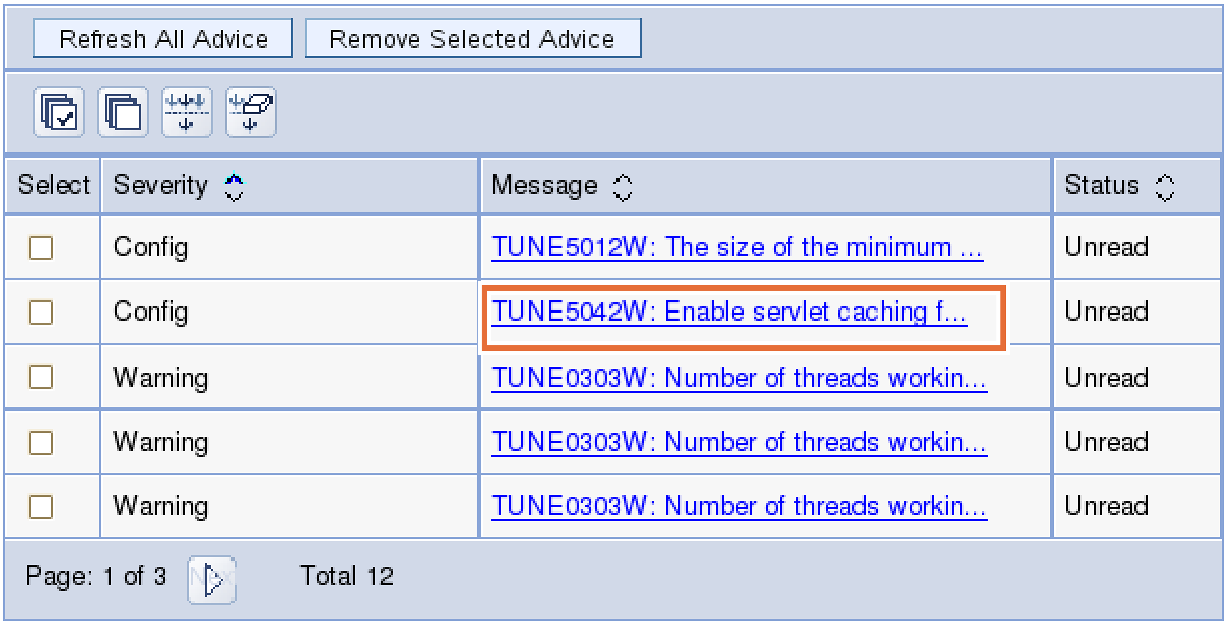
Section 5: Using the Tivoli Performance Viewer performance advisor

WebSphere Application Server includes a performance advisor, the Tivoli Performance Viewer advisor, which is accessed from inside Tivoli Performance Viewer. The Tivoli Performance Viewer advisor provides helpful tuning advice for various resources, cache size, JVM heap size, and more. The Tivoli Performance Viewer advisor also provides recommendations to address inefficient settings.

__ 1. To access Tivoli Performance Viewer advisor messages in the administrative console, select **Monitoring and Tuning > Performance Viewer > Current Activity > server1**. In the Tivoli Performance Viewer navigation pane, click **Advisor** under **server1**.



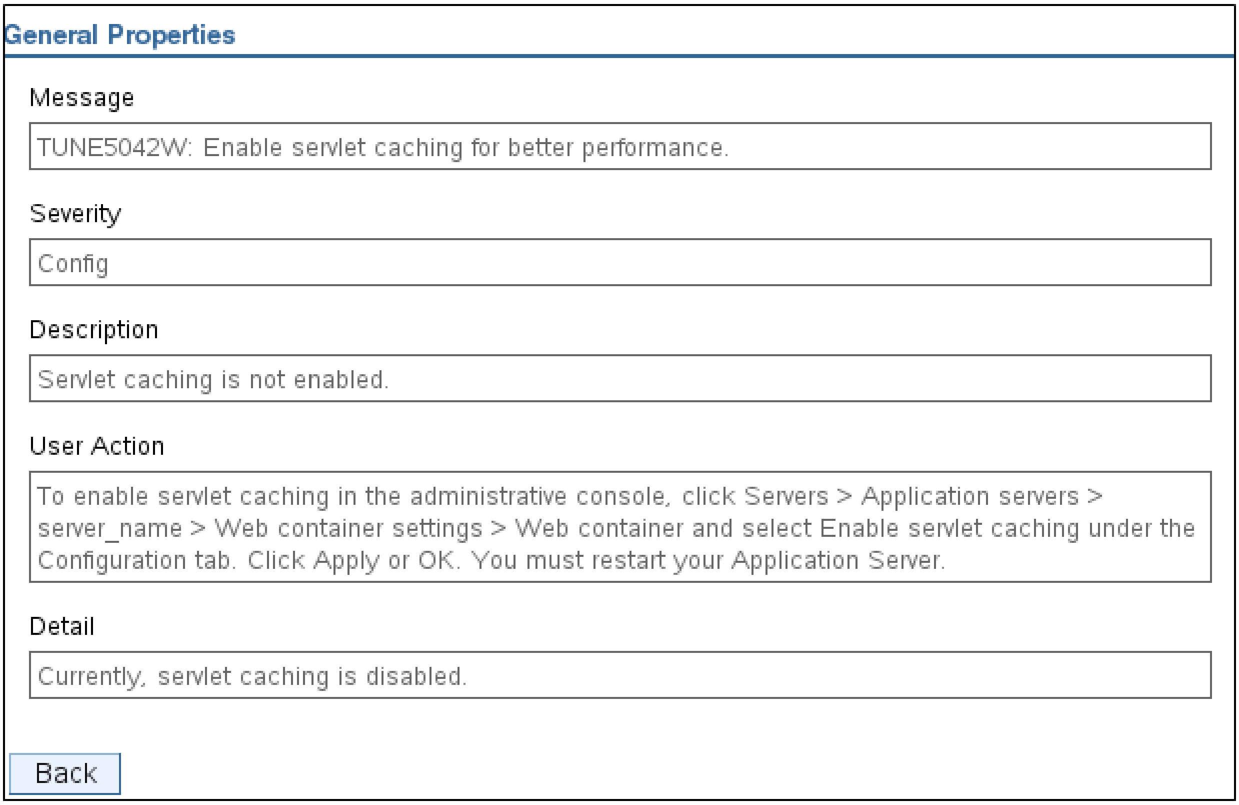
__ a. In the pane on the right, read the provided advice messages.
Are there some Alert messages?________
Is there a configuration advice message?________
If more than one page is available, view the messages on subsequent pages.



Select	Severity	Message	Status
□	Config	TUNE5012W: The size of the minimum ...	Unread
□	Config	TUNE5042W: Enable servlet caching f...	Unread
□	Warning	TUNE0303W: Number of threads workin...	Unread
□	Warning	TUNE0303W: Number of threads workin...	Unread
□	Warning	TUNE0303W: Number of threads workin...	Unread

Page: 1 of 3 Total 12

__ b. Click the link for the **TUNE5042W: Enable servlet caching for better performance** message to see the advice details.



General Properties

Message

TUNE5042W: Enable servlet caching for better performance.

Severity

Config

Description

Servlet caching is not enabled.

User Action

To enable servlet caching in the administrative console, click Servers > Application servers > server_name > Web container settings > Web container and select Enable servlet caching under the Configuration tab. Click Apply or OK. You must restart your Application Server.

Detail

Currently, servlet caching is disabled.

Information

In your lab environment, typically the processor utilization is low, and so you do not see much relevant performance advice. However, you see the configuration advice to turn on servlet caching with descriptions as to how to use the console to turn on this feature.

It is a good practice to run a representative workload against your application. Look at the performance advisor after the workload is run.

You can use the IBM Rational Performance Tester to create load tests and run them against your applications. You can read more about Rational Performance Tester and download a trial version at the following website:
`http://www.ibm.com/software/awdtools/tester/performance`

Section 6: Using request metrics

Request metrics log the time that is spent at major components of the application server, such as the web server plug-in, web container, EJB container, and more.

The request metric architecture differs from the Performance Monitoring Infrastructure (PMI). PMI provides information about average system resource usage, with no correlation between the data across different WebSphere components.

The request metrics tool tracks each individual transaction within WebSphere Application Server, recording the response time of the major components. Some of these response times include time in the web server or in the Enterprise JavaBeans (EJB) container. The collected information can be saved to log files or forwarded to an Application Response Measurement (ARM) agent.

__ 1. Using the administrative console, turn on request metrics by selecting **Monitoring and Tuning > Request metrics**.

- __ a. Under **General Properties**, select **Prepare Servers for request metrics collection**.
- __ b. Under **Components to be instrumented**, select **ALL**.
- __ c. Set **Trace level** to **Debug**.

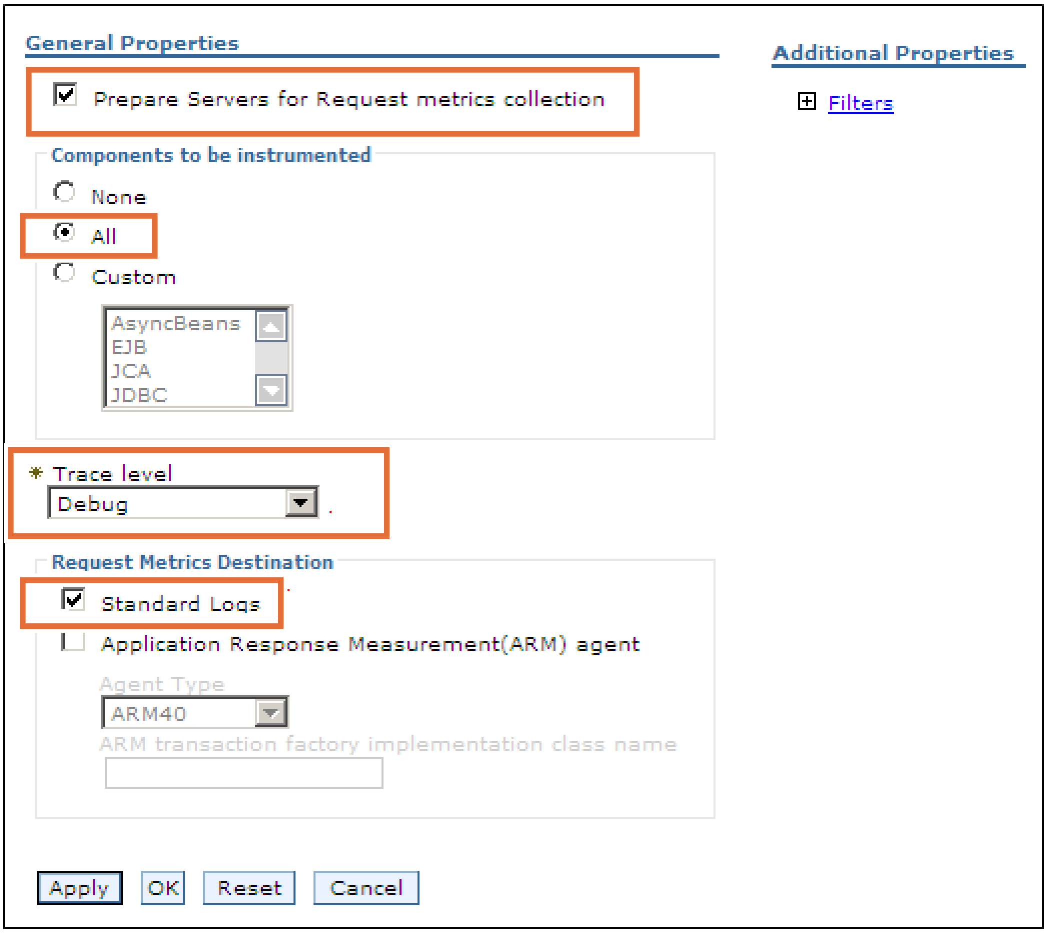
Information

Setting the trace level to **Debug** provides detailed instrumentation data, including response times for all intra-process servlet and Enterprise JavaBeans (EJB) calls. This trace level provides a fine level of detail on each method call.

Setting the trace level to **Hops** generates instrumentation information about process boundaries only (for example, a servlet request that comes from a browser or a web server, and a JDBC request that goes to a database).

You can also filter requests so that only specific incoming requests result in request metrics that are logged. Such filtering can keep the logs from being overloaded with request metrics for every request.

__ d. Under **Request Metrics Destination**, select **Standard Logs**.



General Properties

☑ Prepare Servers for Request metrics collection

Components to be instrumented

○ None

● All

○ Custom

* Trace level: Debug

Request Metrics Destination

☑ Standard Logs

□ Application Response Measurement(ARM) agent

Agent Type: ARM40

ARM transaction factory implementation class name

__ e. Click **OK**.

__ f. **Save** directly to the master configuration and log out of the administrative console.

__ g. Restart **server1**.

__ 2. Open a web browser and run the PlantsByWebSphere application by entering the following address:

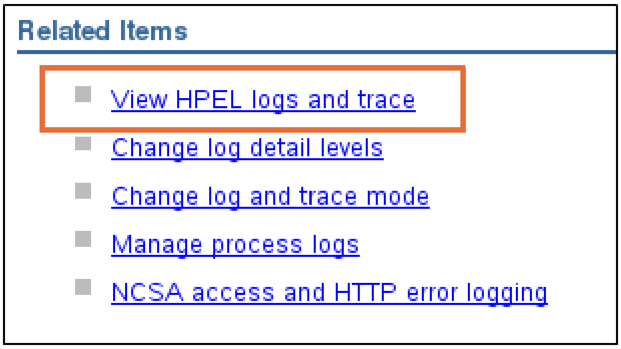
`http://localhost:9080/PlantsByWebSphere`

__ a. Go through the site, look at some plants, buy something, and check out.

__ 3. Now view the standard JVM logs for server1. If HPEL is not turned on for server1, you can go to the Problem determination lab exercise for the steps on how to turn on HPEL. Alternatively, you can open the `SystemOut.log` file with a text editor and search for the PMRM codes. The remaining steps assume that HPEL is turned on for server1.

__ a. In the administrative console, go to **Troubleshooting > Logs and trace > server1**.

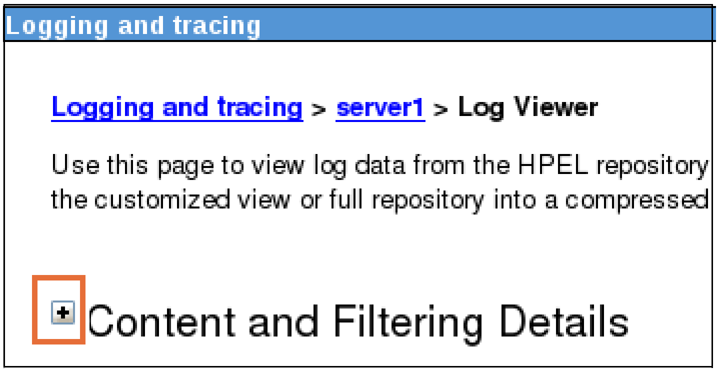
__ b. On the Logging and tracing panel, scroll down and click the link **View HPEL logs and trace**.



Related Items

- View HPEL logs and trace
- Change log detail levels
- Change log and trace mode
- Manage process logs
- NCSA access and HTTP error logging

__ c. You are now in the Log Viewer. Expand the **Content and Filtering Details** section.



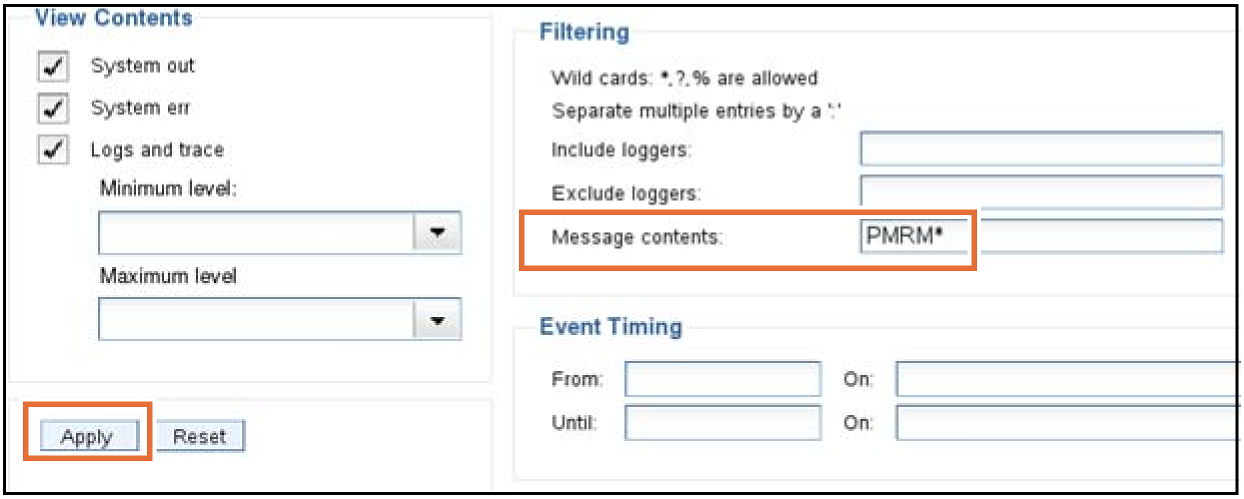
Logging and tracing

Logging and tracing > server1 > Log Viewer

Use this page to view log data from the HPEL repository the customized view or full repository into a compressed

⊞ Content and Filtering Details

__ d. Scroll down to the **Filtering** box and type `PMRM*` in the **Message contents** field.



View Contents

☑ System out

☑ System err

☑ Logs and trace

Minimum level:

Maximum level

Filtering

Wild cards: *,?,% are allowed

Separate multiple entries by a ';'

Include loggers:

Exclude loggers:

Message contents: PMRM*

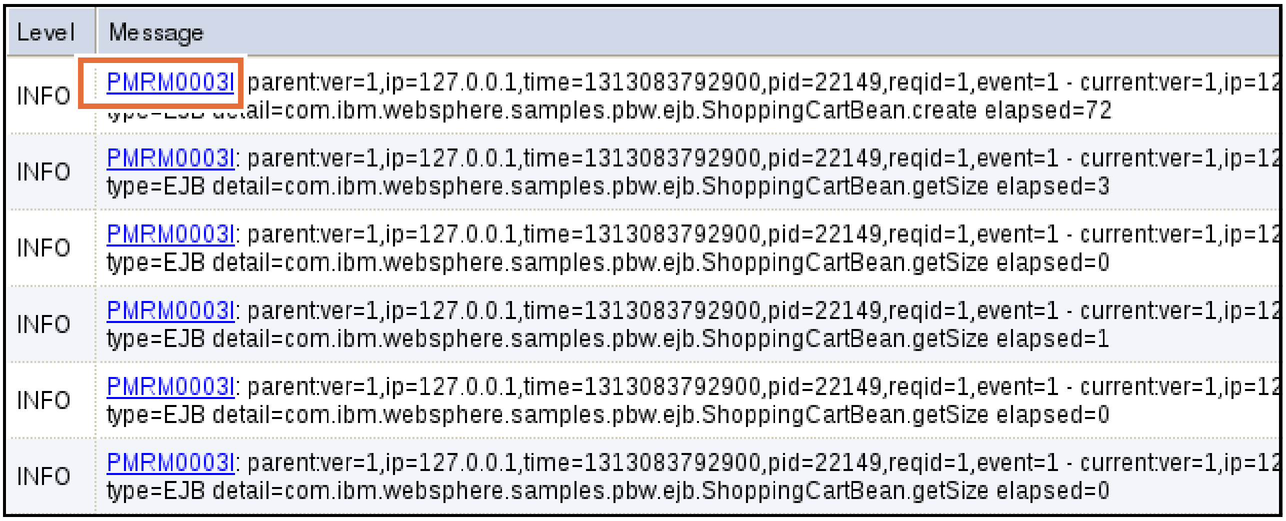
Event Timing

From: On:

Until: On:

__ e. Click **Apply**.

__ f. Now only the request metrics messages are shown in the log view. Click any of the **PMRM0003I** codes to view details of the message.



Level	Message
INFO	PMRM0003I parent:ver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 - current:ver=1,ip=12 ail=com.ibm.websphere.samples.pbw.ejb.ShoppingCartBean.create elapsed=72
INFO	PMRM0003I: parent:ver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 - current:ver=1,ip=12 type=EJB detail=com.ibm.websphere.samples.pbw.ejb.ShoppingCartBean.getSize elapsed=3
INFO	PMRM0003I: parent:ver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 - current:ver=1,ip=12 type=EJB detail=com.ibm.websphere.samples.pbw.ejb.ShoppingCartBean.getSize elapsed=0
INFO	PMRM0003I: parent:ver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 - current:ver=1,ip=12 type=EJB detail=com.ibm.websphere.samples.pbw.ejb.ShoppingCartBean.getSize elapsed=1
INFO	PMRM0003I: parent:ver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 - current:ver=1,ip=12 type=EJB detail=com.ibm.websphere.samples.pbw.ejb.ShoppingCartBean.getSize elapsed=0
INFO	PMRM0003I: parent:ver=1,ip=127.0.0.1,time=1313083792900,pid=22149,reqid=1,event=1 - current:ver=1,ip=12 type=EJB detail=com.ibm.websphere.samples.pbw.ejb.ShoppingCartBean.getSize elapsed=0

__ g. Scroll through the request metrics records. In addition to the Shopping Cart bean, you also see metrics for PlantsByWebSphere (PBW) servlets and JSPs. Look at the different event types. You see `type=URI`, `type=EJB`, `type=JDBC`, and more, reflecting the request flow.

Information

The **time** and **pid** fields are the start time and ID of the application server process. *Type* and *detail* are the description of the type of operation that is timed and its name. Most important is the measured *elapsed* time in milliseconds, which includes all suboperations that this operation calls. The *reqid* is a unique ID assigned to the request by request metrics.

Two correlators are shown, a parent correlator and a current correlator, representing the upstream request and the current operation. A correlator consists of the comma-separated fields that follow the “parent” or “current” keyword. If the parent and current correlators are the same, as in this case where all field values are the same (including *reqid=1*), then the record represents an operation that occurs as it enters WebSphere Application Server.

__ 4. Disable **Request Metrics**.

__ a. Select **Monitoring and Tuning > Request metrics**.

__ b. Clear **Prepare Servers for Request metrics collection**.

__ c. Click **OK**.

__ d. **Save** directly to the master configuration and log out of the administrative console.

__ e. Restart **server1**.

End of exercise

Exercise review and wrap-up

In this exercise, you learned how to turn on different levels of PMI statistics for an application server, monitor an application server with Tivoli Performance Viewer, and configure user settings. You generated some load on applications and examined summary reports and performance modules in Tivoli Performance Viewer. Additionally, you viewed performance messages from the Tivoli Performance Viewer advisor.

You learned how to turn on and configure the Request Metrics tool, and viewed Request Metrics messages in the standard logs of an application server.