

Frequently Asked Questions

WebSphere Performance Monitoring & Tuning

For Webtop Version 5.3 on WebSphere 5.1.x

FAQ Version 1.0
External

FAQ1. Q. How do I monitor Webtop performance in WebSphere?	1
Enabling Performance Monitoring on WebSphere	2
Logging in and Using the Tivoli Performance Viewer	4
FAQ2. Q. Are there guidelines for tuning WebSphere with Webtop?	10
Maximum Heap Size – 1GB	10
Minimum Heap Size – 512MB	10
MaxPerm and MaxNew – ¼ of max heap size	10
Initial Thread Pool Settings	11
Monitoring Thread Pool Usage	11

This FAQ covers using the Tivoli Performance Viewer to monitor Webtop or any other WDK application. It also provides some basic start-up settings to help you start tuning your application server based on the number of users and user habits.

FAQ1. Q. How do I monitor Webtop performance in WebSphere?

A. The Tivoli Performance Viewer ships with WebSphere and is a useful tool. You can run the viewer on Windows or UNIX. The executable is found in the WebSphere home/AppServer/bin directory and is called tperfviewer.sh (UNIX) or tperfviewer.bat (Windows).

Enabling Performance Monitoring on WebSphere

1) To monitor with the viewer, performance monitoring needs to be enabled in WebSphere. Having it enabled does consume system resources. However, for standard monitoring, IBM indicates that no more than 5% of resources would be consumed.

Navigate to your application server, runtime tab, and click the link to Performance Monitoring Services shown below.

The screenshot shows the WebSphere Administrative Console interface. The browser address bar displays `http://us301:9091/admin/secure/logon.do`. The console header includes "WebSphere Application Server Version 5" and "Administrative Console". The left sidebar shows a navigation tree with "Application Servers" selected. The main content area shows the configuration for "server1".

Application Servers > server1

An application server is a server which provides services required to run enterprise applications. ⓘ

Runtime | **Configuration**

General Properties

Process Id	21974	The native operating system's process id for this server.
Cell Name	us301	The name of the cell in which this server is running.
Node Name	us301	The name of the node in which this server is running.
State	Started	The runtime execution state for this server.

OK

Additional Properties

Transaction Service	Specify settings for the Transaction Service, as well as manage active transaction locks.
Performance Monitoring Service	specify settings for performance monitoring, including enabling performance monitoring, selecting the PMI module and setting monitoring levels.
Product Information	Product information for this installation of WebSphere Application Server

WebSphere Status ⓘ [< Previous](#) [Next >](#) June 26, 2006 5:01:59 PM PDT ↻

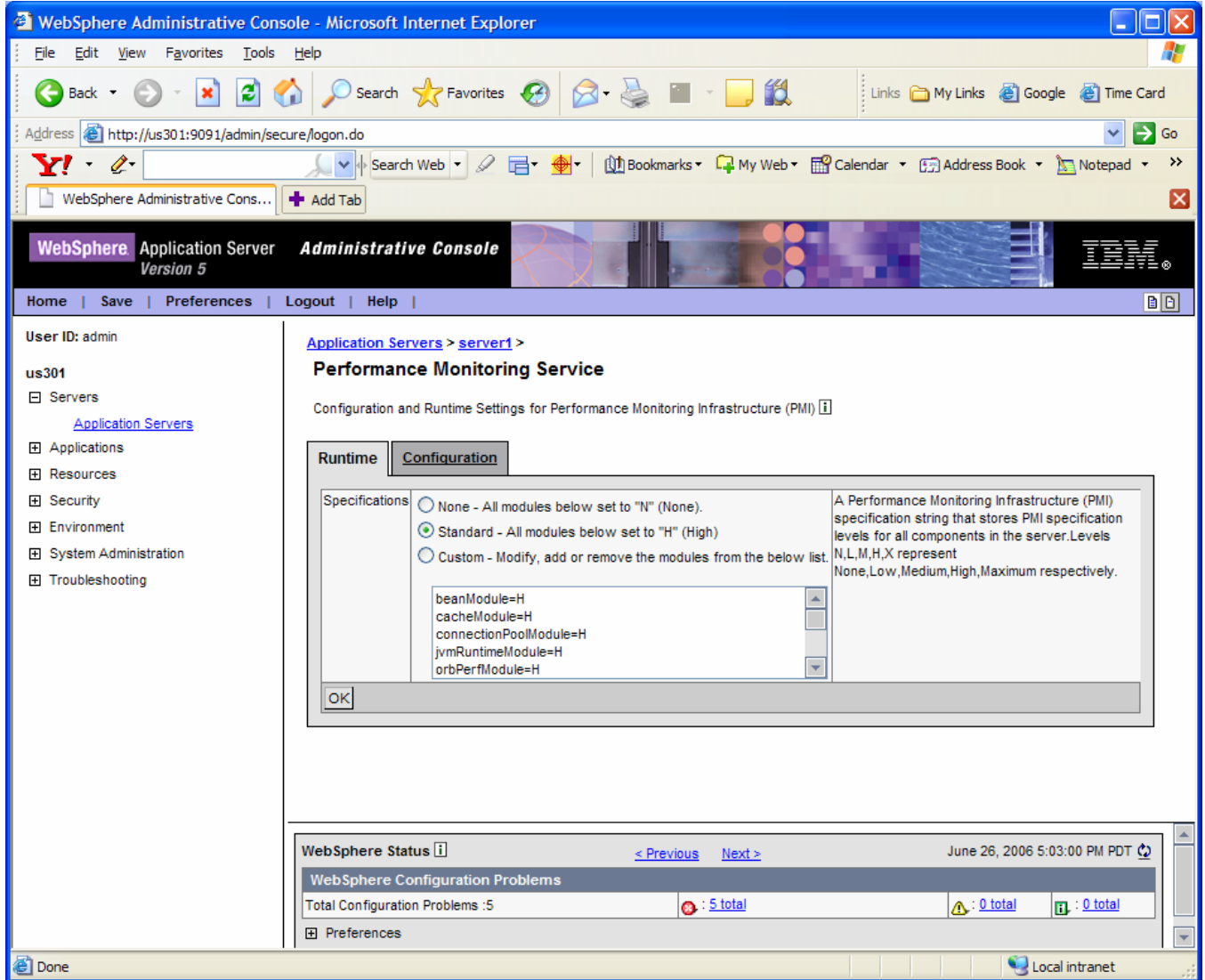
WebSphere Configuration Problems

Total Configuration Problems :0	🔴 : 0 total	🟡 : 0 total	🟢 : 0 total
---------------------------------	-------------	-------------	-------------

Preferences

Address bar: `http://us301:9091/admin/com.ibm.ws.console.servermanagement.forwardCmd.do?forwardName=PMIService.config.view&sfname=si`

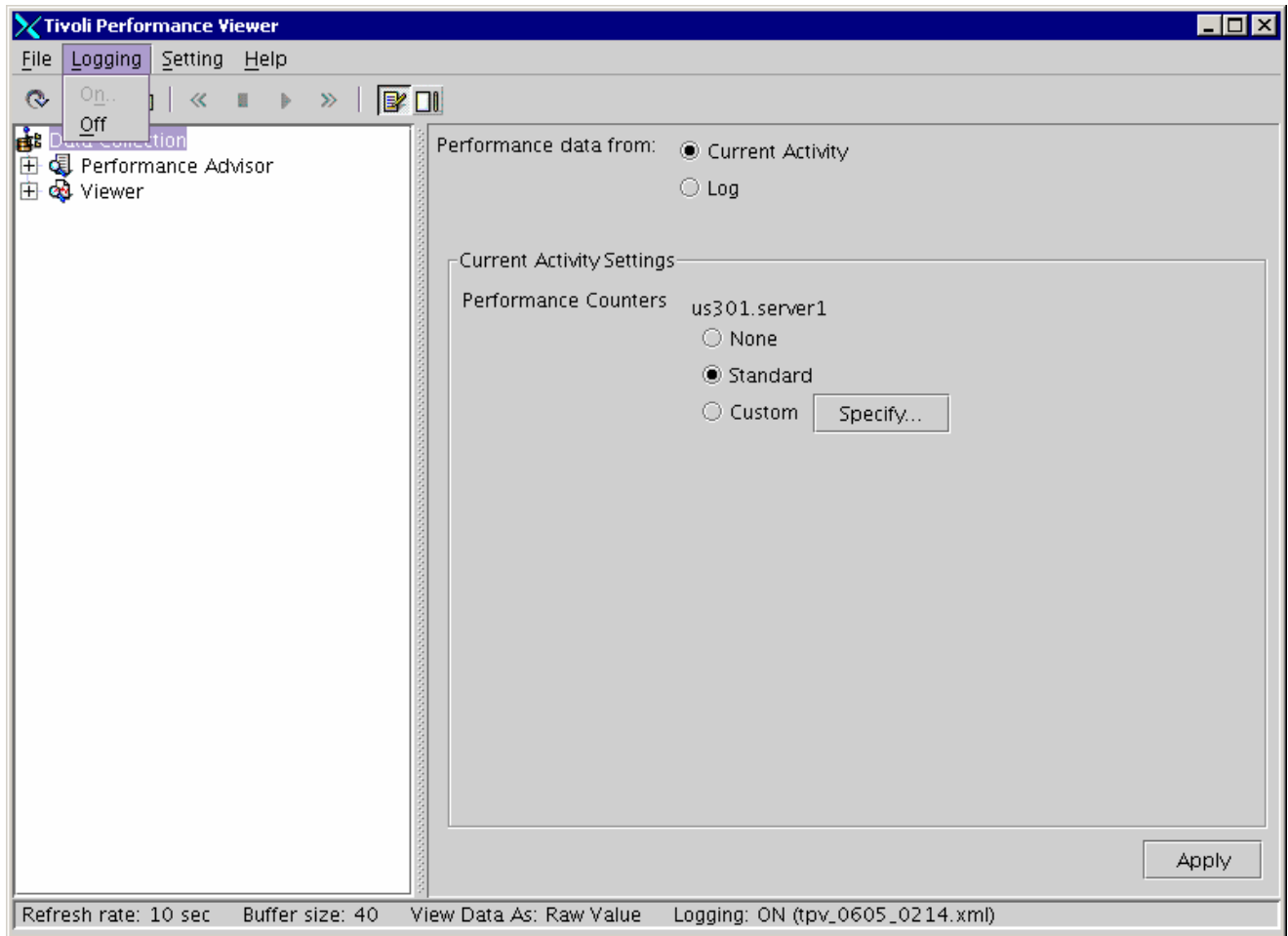
Enable Standard performance monitoring as shown below. Be sure to save your configuration changes after clicking OK.



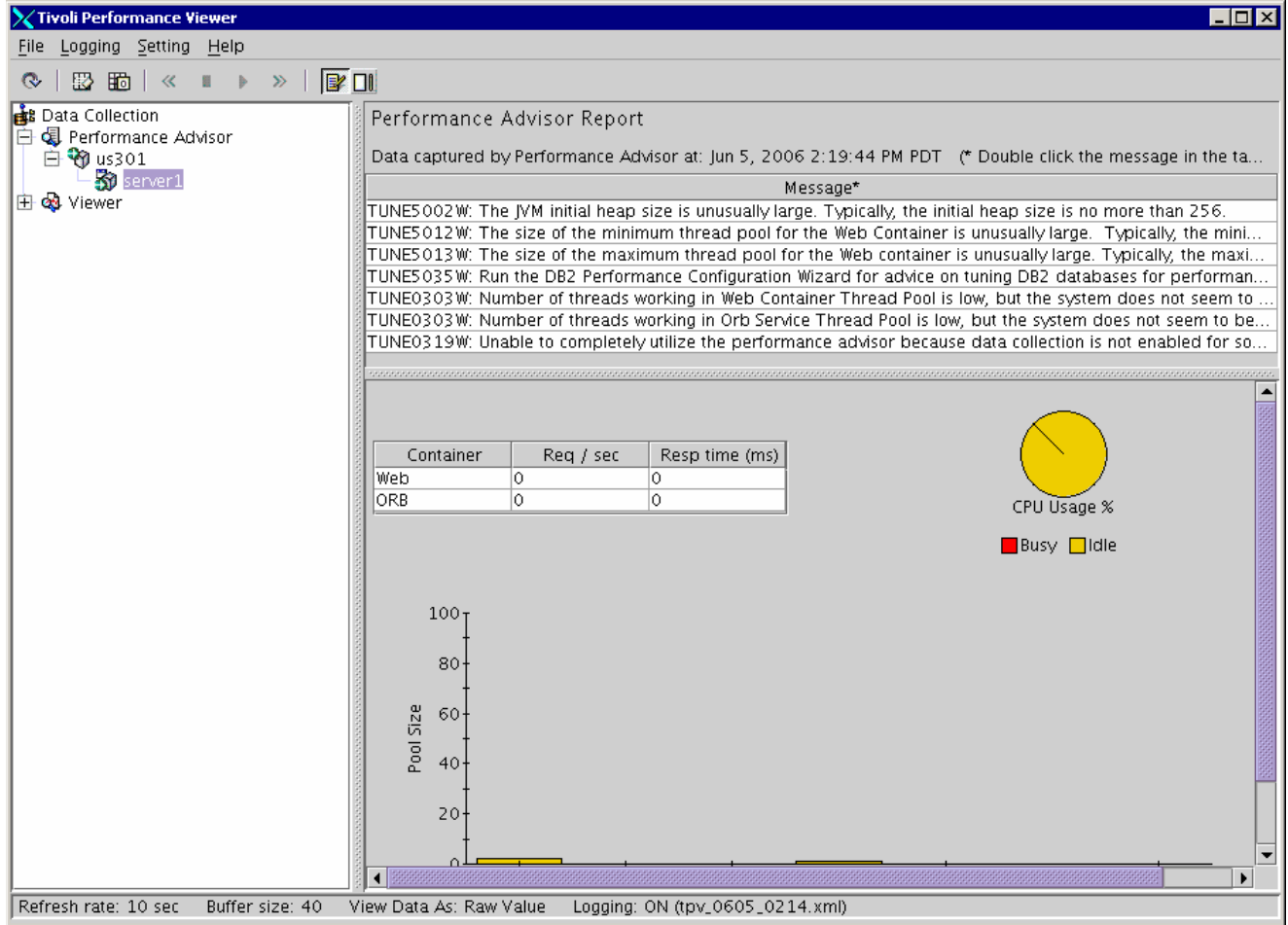
Logging in and Using the Tivoli Performance Viewer

The performance viewer runs on Windows or UNIX. As noted early, the executable is found in the WebSphere home/AppServer/bin directory. If running on UNIX, you execute the shell `tperfviewer.sh`. If running on Windows, you execute `tperfviewer.bat`.

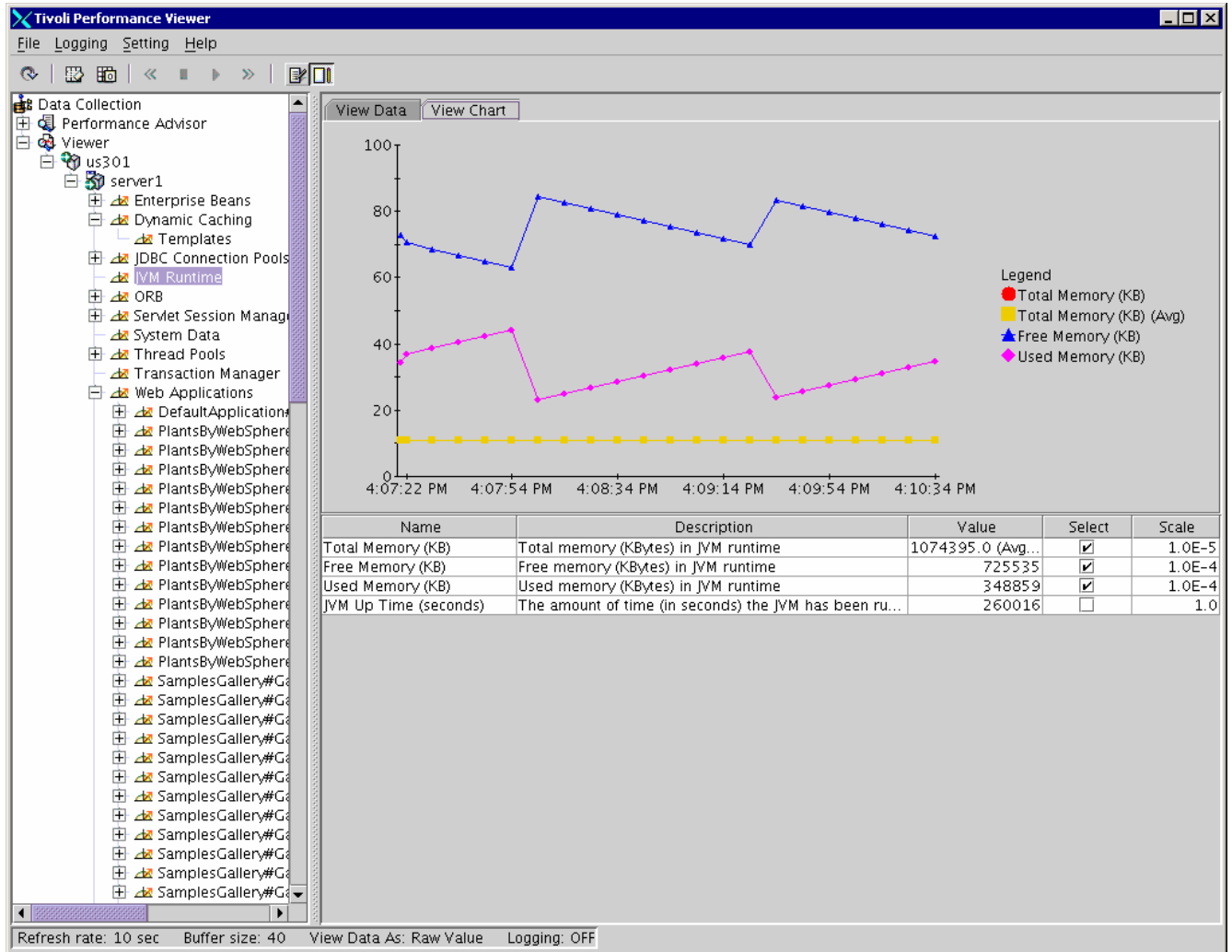
1) Logging Monitor Activity - logging is enabled via the drop down menu (see screen shot below). When logging is enabled, it creates a unique *.xml file in the WebSphere home/AppServer/logs directory. You can customize which counters it logs, but typically the default of *standard* suffices.



2) One feature of the viewer is the Performance Advisor. It provides advice that should be taken in the proper context. For example, it suggests the initial heap size be no more than 256mb. In reality, this depends on your application and expected heap usage and shouldn't be taken in absolute terms. See screen shot below.

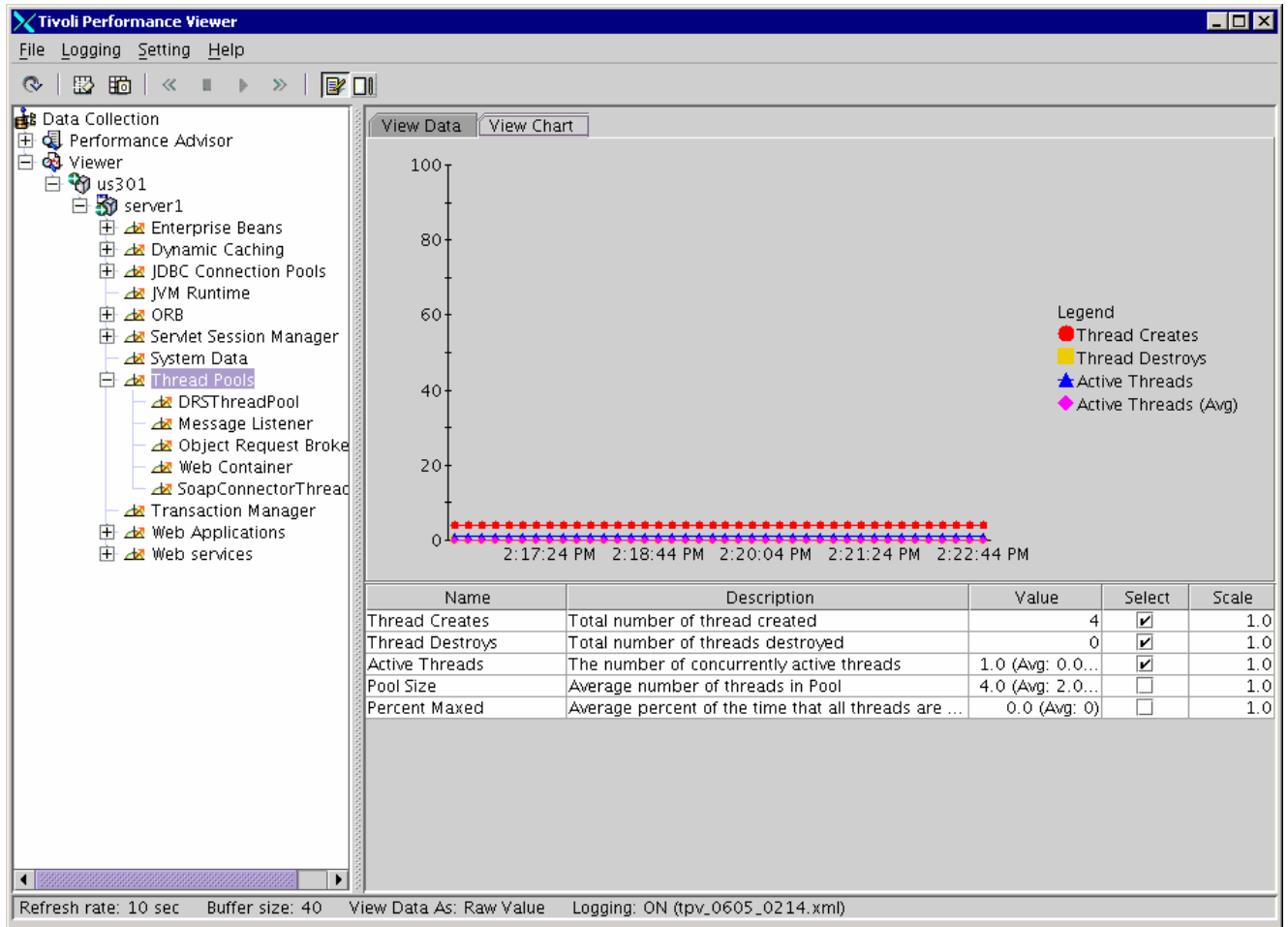


3) JVM Heap – you can view JVM heap usage and total size. Average heap usage along with the min and max heap used is important information for helping you decide how to configure it for your application. See screen shot below.



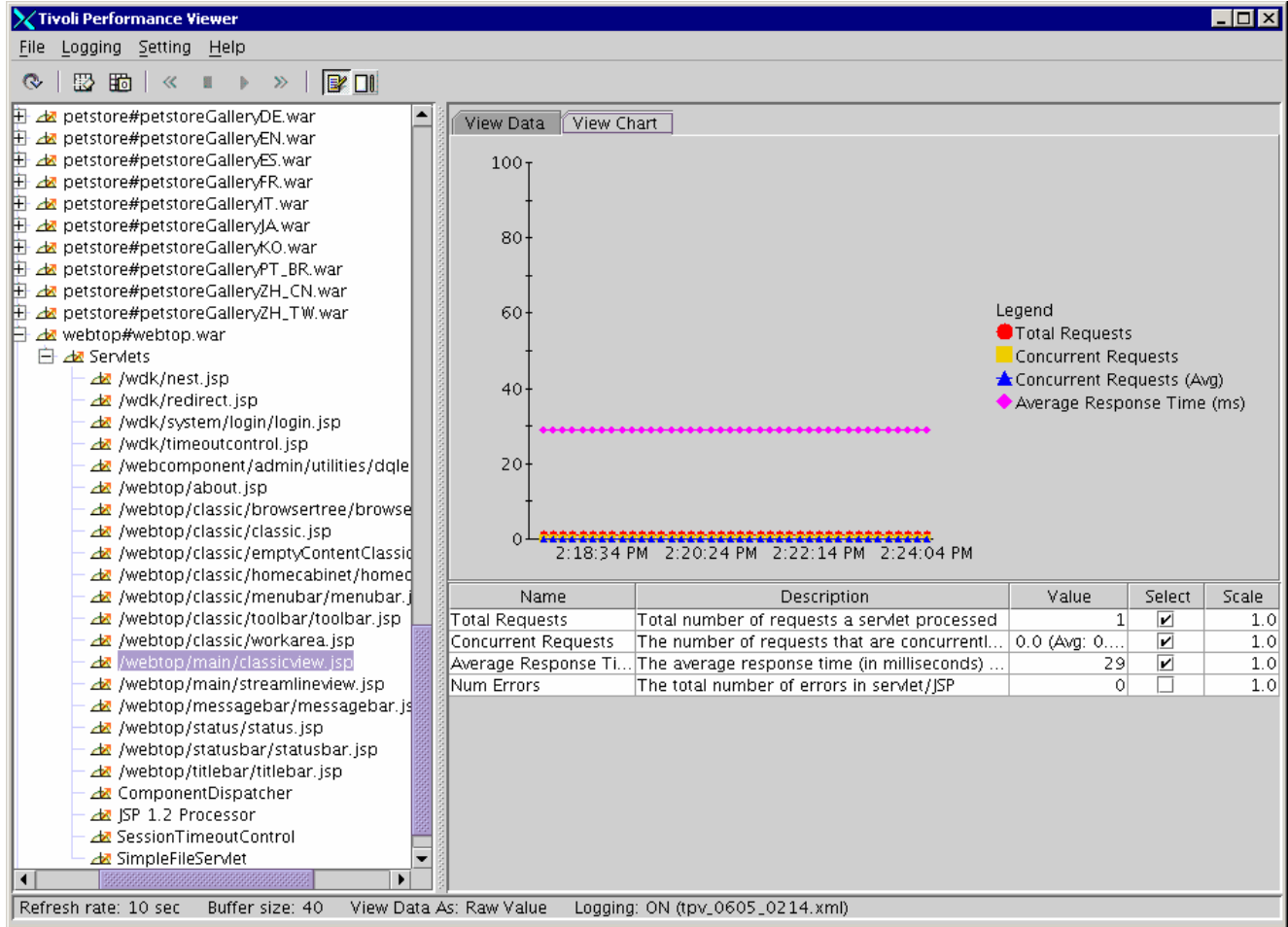
For more information on sizing your heap properly, please refer to FAQ2 in this document.

3) Thread Pools - It nicely monitors thread pools on the App Server. See screen shot below.

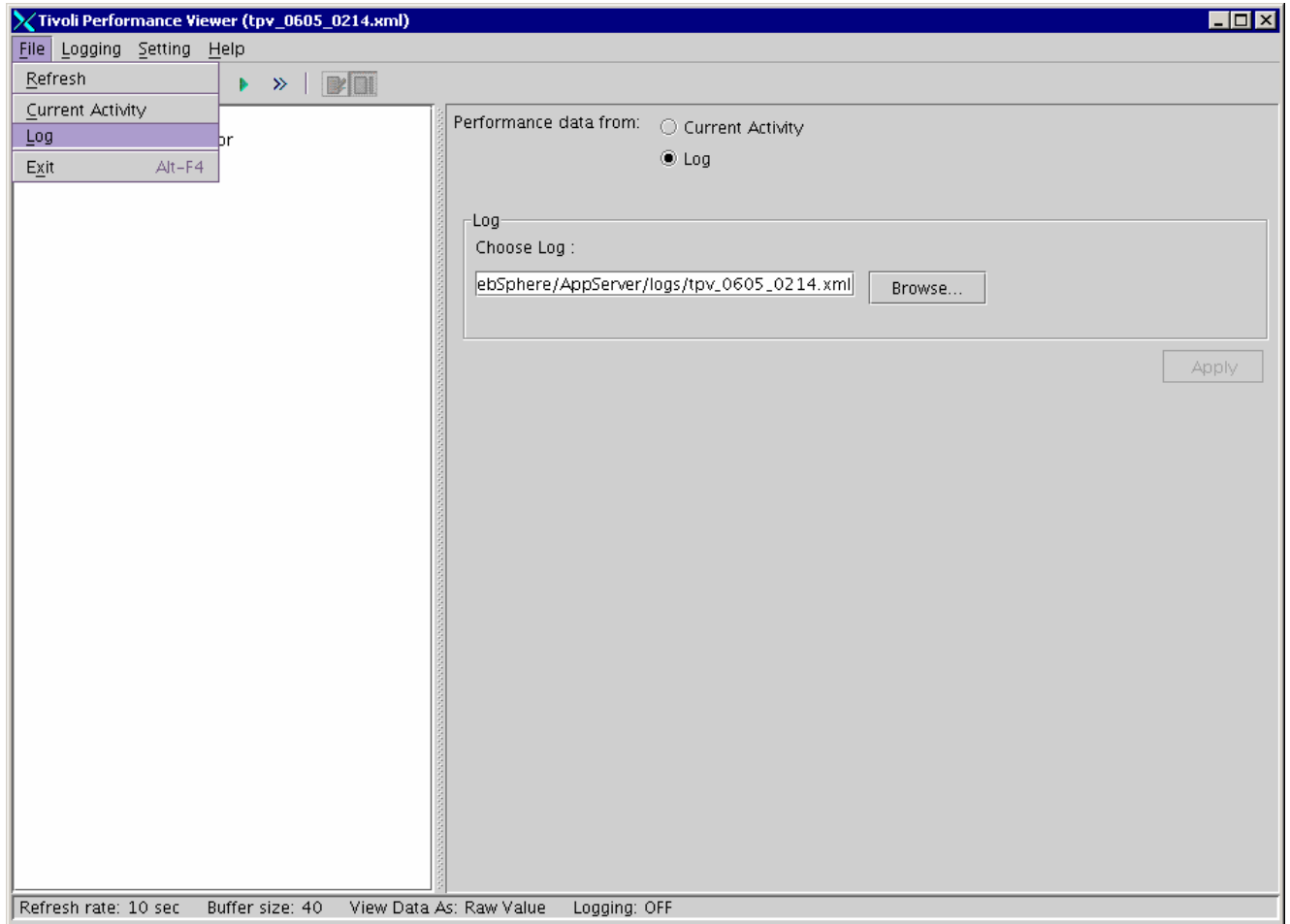


For more Information on configuring your thread pool optimally for your environment and user habits, please refer to FAQ2 in this document.

4) Webtop servlets - you can monitor Webtop servlets (total requests, response times, errors, etc.) See screen shot below.



5) Logging – the logging feature is very useful. You can record a PerfViewer session and play it back at recorded speed or faster. To open a log file for playback, go to drop down menu FILE, then click Log. Navigate to the file by clicking the browse button. Remember, all log files are stored by default in the WebSphere home/AppServer/logs directory. See screen shot below.



FAQ2. Q. Are there guidelines for tuning WebSphere with Webtop?

A. Tuning your WebSphere environment (Heap size, thread pool, etc.) is a bit of an art. Much depends on user habits, hardware limitations, and software versions.

What follows are general guidelines. They are starting points that may not be optimal for your environment. As always, test before implementing any configuration in a production environment.

In this document it is assumed we are working with 200 – 300 typical Documentum users per JVM, and that this JVM is adequately supported with today's typical CPU and memory configurations. If you have different user targets or different user habits, adjust your starting point accordingly.

Maximum Heap Size – 1GB

Generally, we've found little value going over 1GB of heap size. Typically the application is slower as managing a large heap impacts performance. 1GB is a good starting point with our assumed user count.

Minimum Heap Size – 512MB

This setting is more flexible. But, by starting out at a higher number, WebSphere doesn't have to allocate more memory as often once you are up and running.

You may also find that performance is better when the min and max heap sizes are identical. However, be careful if you have extended periods of light user activity. During those periods, garbage collection (GC) could be more expensive given a heap that is much larger than needed.

How many JVMs

Adding more users may require more JVMs. If you find your heap is over 80% utilized at peak usage and 1GB or larger, you should consider adding another JVM to compensate as opposed to increasing your heap size.

MaxPerm and MaxNew – ¼ of max heap size

MaxPerm region (default 64MB) holds class data and long-lived resources and is often undersized. Also, MaxNew region is for short-lived objects. In some version of the JDK it helps to also set this to no more than ¼ of max heap size. See references at bottom for more details.

Initial Thread Pool Settings

Proper thread pool settings vary widely by the user base's habits and the operating environment. Use this rough formula to arrive at a starting thread pool per JVM: (Nbr of active users doing content transfer * 2) + (Nbr of other active users)

Given our assumptions the formula might look as follows: $(200 * 2) + 50 = 450$ threads. As a result, you might want to set your initial Min and Max thread count as follows:

Minimum: 200
Maximum: 500

Note: WebSphere threads are tied closer to the number of users as compared to other application servers, i.e., BEA WebLogic. This results in relatively large thread pools on WebSphere.

Monitoring Thread Pool Usage

Monitor thread pool usage in Tivoli. In particular, you should monitor Percent Maxed statistic which tells you the average percent of time ALL threads are in use. If this ever rises above 10%, consider adding more threads to cover periods of peak system use.

[Reference Section]

- [IBM WebSphere Application Server 5.1.x - Enabling Performance Monitoring Services](#)
- [IBM WebSphere Application Server 5.1.x - Monitoring Performance with Tivoli Performance Viewer](#)
- [IBM WebSphere Application Server 5.1.x - Java virtual machine settings](#)
- [IBM WebSphere Application Server 5.1.x - Thread pool settings](#)