



# **IBM eServer iSeries Performance and Tuning Tips for Oracle's JD Edwards EnterpriseOne 8.10 and 8.11 Web Server**

**Gerrie Fisk  
Senior iSeries System Specialist  
IBM / Oracle EnterpriseOne ICC**

**Clark Scholten  
Senior Software Engineer  
iSeries ERP Development – Oracle Support**

---

Table of contents

---

**2 EnterpriseOne Performance Management**

**2 Apache HTTP Server Performance**

**12 WebSphere Application Server Tuning**

**18 EnterpriseOne Application Tuning**

**22 iSeries Database Tuning**

**31 Appendix A – Software and Hardware Levels**

**33 Appendix B - Configuration Files**

**52 Appendix C – Special Note if upgrading from Release 8.9 or 8.10**

**EnterpriseOne Performance Management**

The key to good system performance remains the same regardless of the applications running. It is to provide sufficient system resources in a balanced configuration. A well performing system has the available processor, memory, disk, and network capacity—individually and in balance with one another— that is sufficient to handle the application workload so that no performance bottlenecks exist. Performance management, the technique used to make sure this is the case, is an ongoing, cyclical process. It consists of three key steps:

1. Measure and analyze
2. Adjust
3. Re-measure

For good system performance, these steps should be repeated on a regular basis, and the results used to predict and schedule required system upgrades as well as to resolve specific performance problems.

Oracle’s JD Edwards® EnterpriseOne is a complex application that provides Enterprise Resource Planning (ERP), Supply Chain Management (SCM), and Customer Relationship Management (CRM) capabilities using Web clients. The following sections focus on examining and optimizing the performance of the Web client environment running EnterpriseOne 8.11 with IBM® WebSphere® Application Server 5.0. Appendix A lists the specific software and hardware levels used in the tests. The recommendations throughout this paper apply to JD Edwards EnterpriseOne Release 8.10 as well.

For the best performance of the EnterpriseOne Web client environment on the IBM® eServer™ iSeries™ and eServer i5 platforms, it is important to tune parameters on the client and server. Because the user interface is a Web browser, there is little to tune on the client side. However, on the server side the following can be tuned:

- Apache HTTP Server
- WebSphere Application Server
- EnterpriseOne (the application)
- DB2® UDB for iSeries (the database)

**Apache HTTP Server Performance**

The IBM HTTP Server for iSeries (5722-DG1) product offers two HTTP servers. For WebSphere 5.0 running the EnterpriseOne 8.11 Web

client solution, J. D. Edwards requires the HTTP server (powered by Apache). The Apache HTTP Server can be started, analyzed, and tuned by using iSeries system commands or through a browser interface called the HTTP Server Administration interface. The browser interface offers ease-of-use features, such as selecting options from lists rather than typing in entries to the configuration file. It also offers extensive help text. It is the recommended user interface. To start the browser, use the iSeries command: `STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)`

To access the browser interface, start a Web browser, such as Microsoft® Internet Explorer, and enter the URL `http://<server-name>:2001` where <server-name> is the name of the iSeries server. When prompted to sign on to the server, enter a valid iSeries user profile and password.

Configuration changes can be made through the HTTP Server Administration interface or by directly editing the HTTP configuration file (using the `edtf` command). There is a configuration file for each HTTP Server instance located in the IFS directory `/www/<http-server-name>/conf`, where <http-server-name> is the name of the Apache HTTP Server instance. This directory contains a file, `httpd.conf`, which holds all of the configuration information for that instance. The information below describes how to make changes to the configuration through the browser interface and also lists the Apache configuration keywords and values that can be directly entered into the `httpd.conf` file.

### ***Measure and Analyze***

To measure the performance of the HTTP Server powered by Apache, use the general iSeries system performance tools, such as Performance Monitor and Performance Explorer. To closely monitor what is happening in the Apache HTTP Server instance, examine the access log which is enabled by default. Because the access log impacts performance, one of the performance recommendations below is to disable this logging function during normal operations. Although access logging is disabled for performance, it can be re-enabled when there is a need to debug problems.

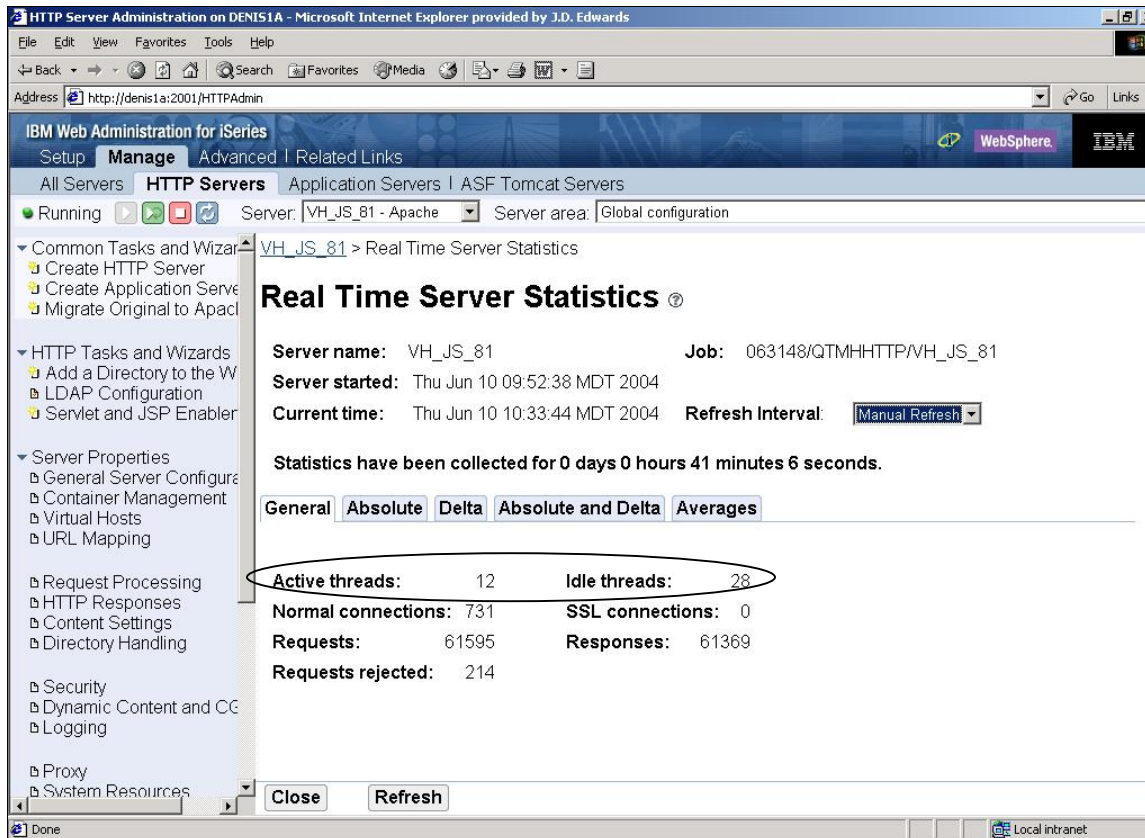


Figure 1 – Real Time Server Statistics show current activity on the Web server.

To more closely examine what is happening while the Web server is active, the Apache Server on iSeries provides real time statistics. To view these statistics, click on the “Real Time Server Statistics” link on the left hand pane, as show in <Figure 1>.

The server statistics screen shows information about the number of connections in use and the number of requests and responses. Monitor the number of active and idle threads during the time that the system is running its peak load. Adjust the number of threads accordingly, as described in Tip 3 below. Additional information is available on the “Absolute” and “Delta” tabs, such as the byte counts sent and received. Use this information to look at the communications workload running on the system during peak time periods.

### **Adjust**

The Apache Server has several settings that can be changed to improve the performance of the EnterpriseOne Web client solution. Enable these settings one at a time to ensure that each recommended change provides a benefit. Changes can be made to the Apache configuration in two ways:

- By selecting the Web server instance, through the Web interface
- By directly editing the configuration file, httpd.conf, located in the IFS (using the edtf command)

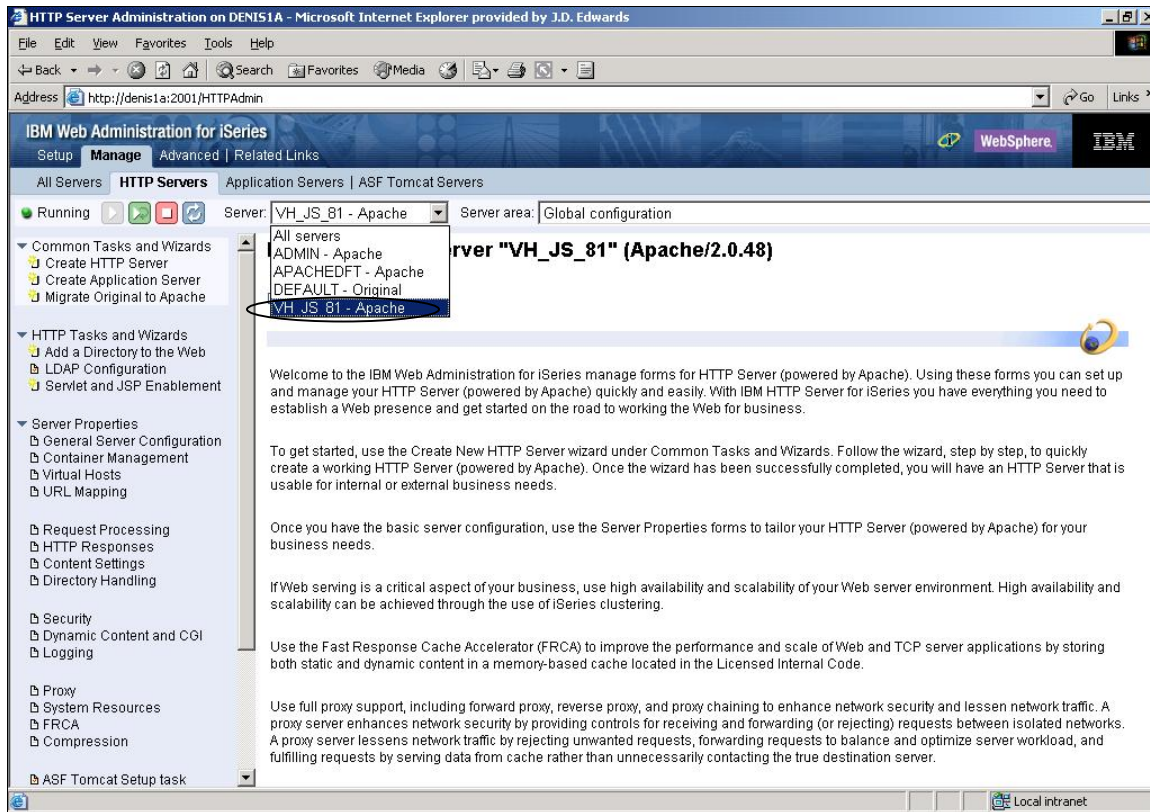


Figure 2 – Select the Apache instance

To make the changes through the Web interface, select the Apache HTTP Server instance from the list as shown in <Figure 2>. In this example, select VH\_JS\_81 from the list and make changes to the settings associated with this instance.

### 1) Minimize the logging in the production environment

As discussed earlier, logging is useful during the initial bring-up of the Web client environment and to debug problems. However, when the production environment is running and stable, disabling extraneous logging provides a performance benefit.

The Apache HTTP Server provides two different types of logging: error logging and access logging. The error log can be configured to capture various types of errors. By default it will capture all warnings, errors, and critical conditions. The recommendation is to change this setting so that only errors and critical conditions are captured in the error log. Also, since a separate error log will be created for each day that the HTTP server is running, change the configuration to have logs automatically removed after seven days, as this reduces the disk requirements for the HTTP server.

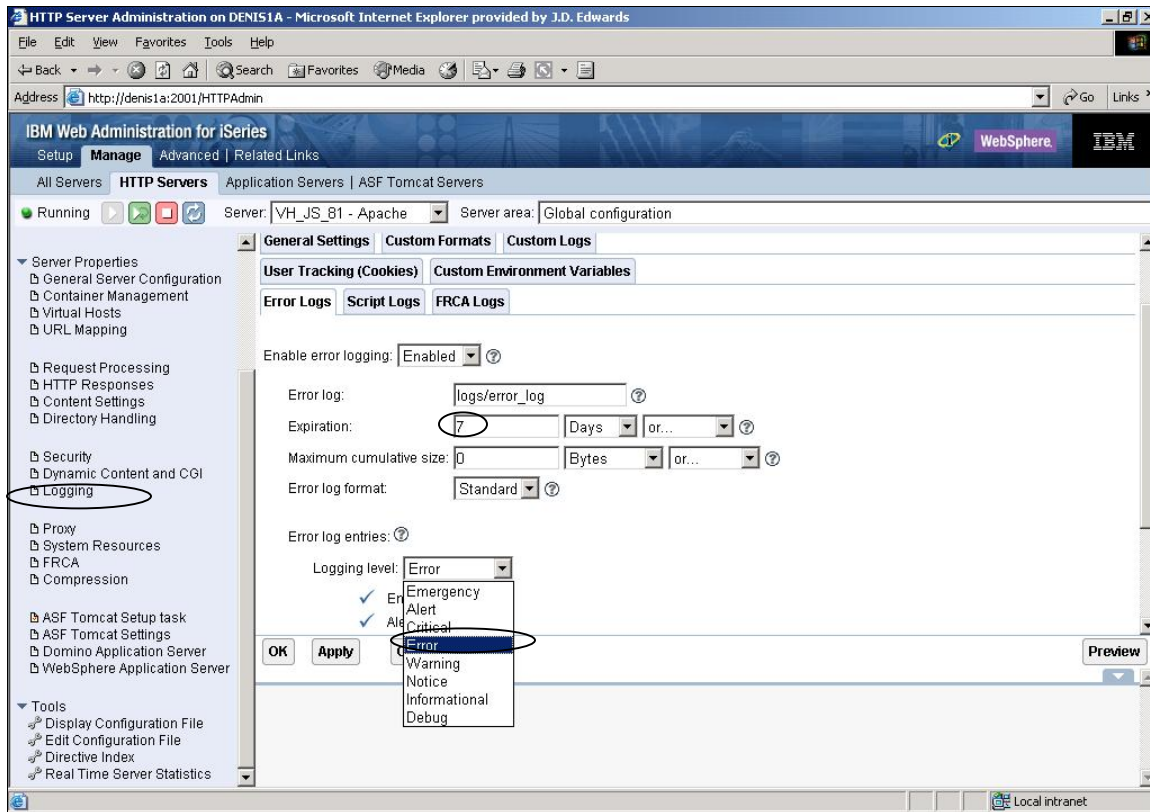


Figure 3 – Change the error log settings in Apache

One way you can change the log settings is to select “Logging” from the left pane on the Apache Administrative Server, then select “Error Logs.” As seen in <Figure 3>, on the error logs page, change the “Expiration” value to “7” days, then select the “Logging Level” pulldown list and select “Error” instead of “Warning.” Click the Apply button after making the change, otherwise the changes are not saved.

Alternatively, you can change the configuration file by editing it to add the following two lines:

```
LogLevel Error
LogMaint logs/error_log 7 0
```

The access log – as the name indicates – captures all HTTP accesses to the system. Each browser that accesses the system via a URL will be noted by an entry in the access log. Depending on the number of users connected to the system, this log can become very large. As a result, the performance recommendation is to turn off all HTTP access logging. However, the access log tracks all access to the system, and this log can be used to monitor the external systems accessing the system through the HTTP server or to determine the location of a Web attack. So, for security reasons, you may determine the need to live with any performance affect that results from keeping the HTTP access log active on your system.

One way to disable the access log is to click on the “Custom Format” tab on the “Logging” page. As seen in <Figure 4>, click the radio button on each of the log formats and then click the Remove button. After all of the log formats are removed, click the Apply button. Then click on the “Custom Logs” tab, select the entry for “logs/access\_log”, click the Remove button, and then click the Apply button.

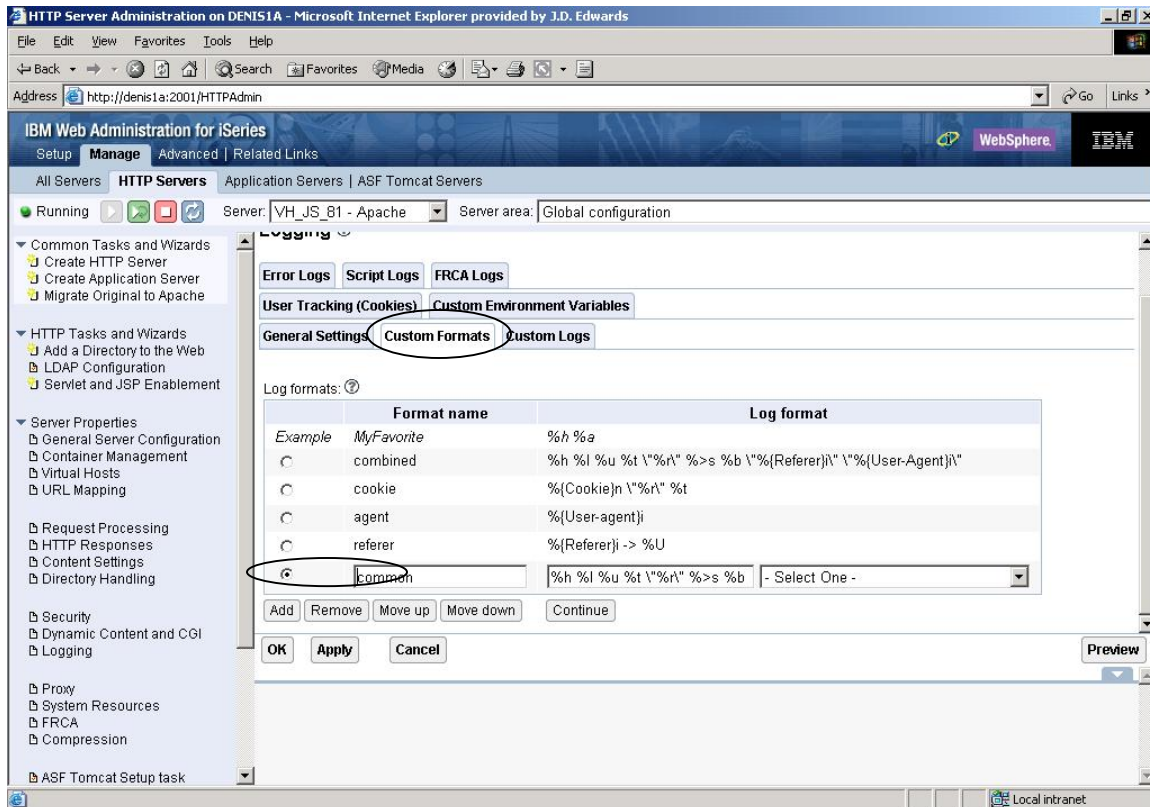


Figure 4 – Eliminate access logging for the Apache HTTP Server instance

Another method of disabling the access log is to change the httpd.conf file for this instance—comment out the following lines in the file by putting a number sign (#) in front of each line:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""" combined
LogFormat "%{Cookie}n \"%r\" %t" cookie
LogFormat "%{User-agent}i" agent
LogFormat "%{Referer}i -> %U" referer
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log combined
```

If the system has been running with access logging enabled, deleting the existing logs will clean up system storage. The logs are located in the directory /www/<http-server-name>/logs. Delete all of the files in this directory; the Apache server automatically creates new files as needed.

***2) Allow unlimited requests per connection***

When a browser on a client machine makes a connection to the Apache HTTP server, the server maintains (or keeps alive) the connection and allows multiple requests from many browsers to be made on the same connection. It is more efficient to keep the same TCP connection for as many requests as possible. By default, the HTTP Server will use the same TCP connection for 100 requests. The recommendation is to change the HTTP server setting so that an unlimited number of requests can be sent on the same connection.

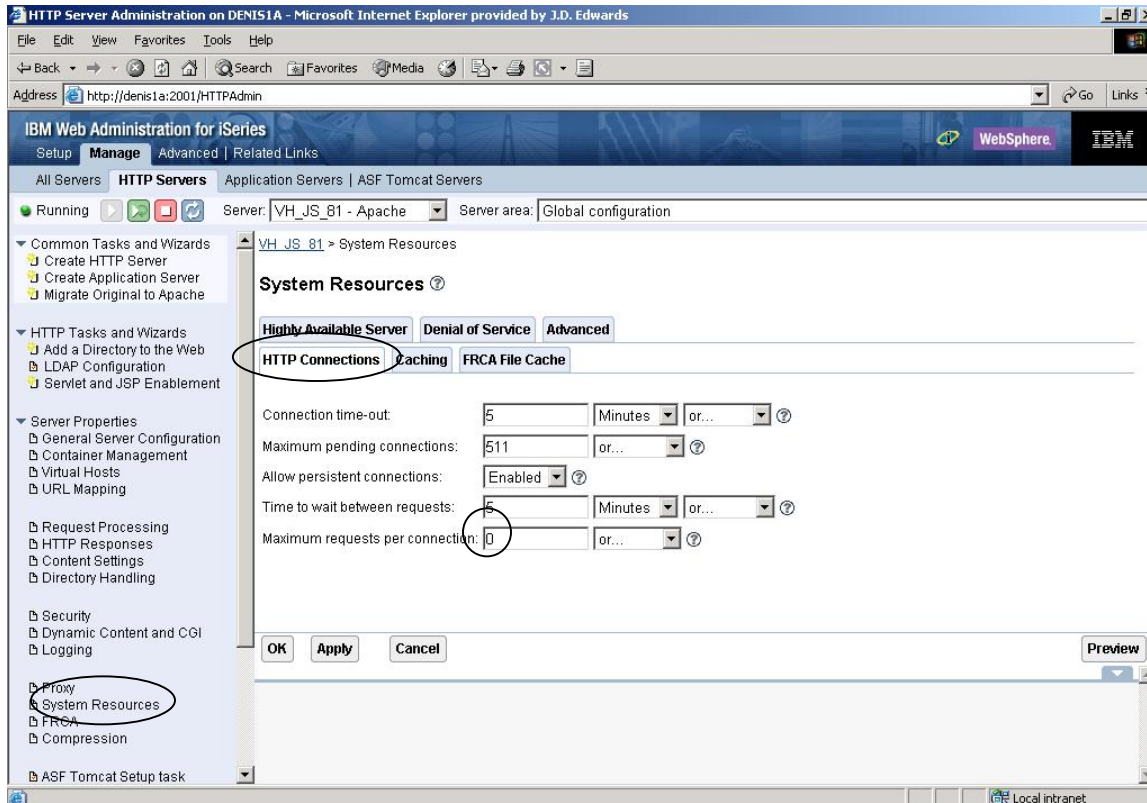


Figure 5 – Allow unlimited requests per connection

One method of changing this setting is to select “System Resources” from the left pane of the Apache Administrative Server and select the “HTTP Connections” tab as shown in <Figure 5>. Change the number in the field, “Maximum requests per connection” from 100 to 0. Though this may seem counter-intuitive, “0” indicates that the server will allow an unlimited number of requests.

Another way to change this setting is to edit the httpd.conf configuration file by adding the following lines:

```
MaxKeepAliveRequests 0  
KeepAlive On
```

### 3) Set the number of threads for each job

The Apache HTTP server job maintains a pool of threads. The number of threads is determined by a configuration setting. The thread pool is used to service all the Web browsers that connect to the server job. If all threads within the Web server job are in use and a new request arrives, an additional job will be started—with the same number of threads in its pool. The recommendation is to keep the proper number of threads available to avoid the extra overhead of starting another Web server job.

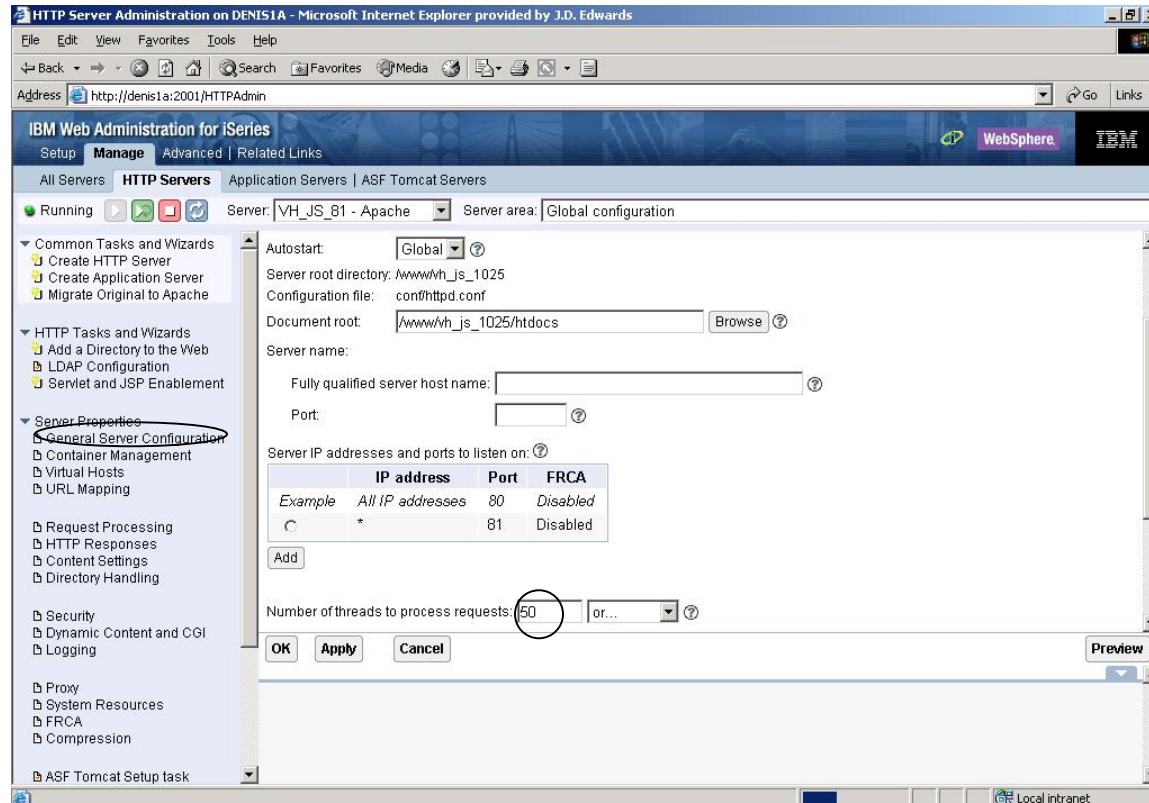


Figure 6 – Set the number of threads to handle requests in the Apache HTTP Server instance.

Adjust the number of threads that run in each of these Web server jobs by selecting “General Server Configuration” from the navigation pane on the left-hand side and then selecting the “General Settings” tab. Change the field for “Number of threads to process requests” and then click the Apply button. The default is 40, and lab tests have found that this number of threads is sufficient to adequately handle 400 users doing typical workloads through browser connections. If the system is running fewer or more users than this, change this setting accordingly. For example, if the system is being configured for 500 users, change the setting to 50 as shown in <Figure 6>.

Alternatively, you can adjust the number of threads running in each Web server job by editing the httpd.conf file to add the following line:

```
ThreadsPerChild 50
```

#### **4) Enable Compression**

Apache supports HTTP Server data compression before files are output to clients that support decompression. This compression of files to the client decreases the data volume sent over the network and improves response times. It is enabled via a DEFLATE filter, which can be configured to only compress files located in certain directories. Remote clients connected to the network at slow speeds have the greatest potential for performance gains, though locally connected clients will also see lesser performance gains. Web client's browser version must support data decompression. Internet Explorer and Netscape Navigator both support this function when running at the versions recommended by the Oracle MTR (Minimum Technical Requirement).

To enable compression for the Apache HTTP Server, first ensure that the Apache HTTP Server is at the correct version. Use the WRKPTFGRP command to ensure that V5R2 group PTF SF99098 is at level 13 or higher. The compression function was enabled with level 10; however, the GUI interface to configure the function is enabled with level 13.

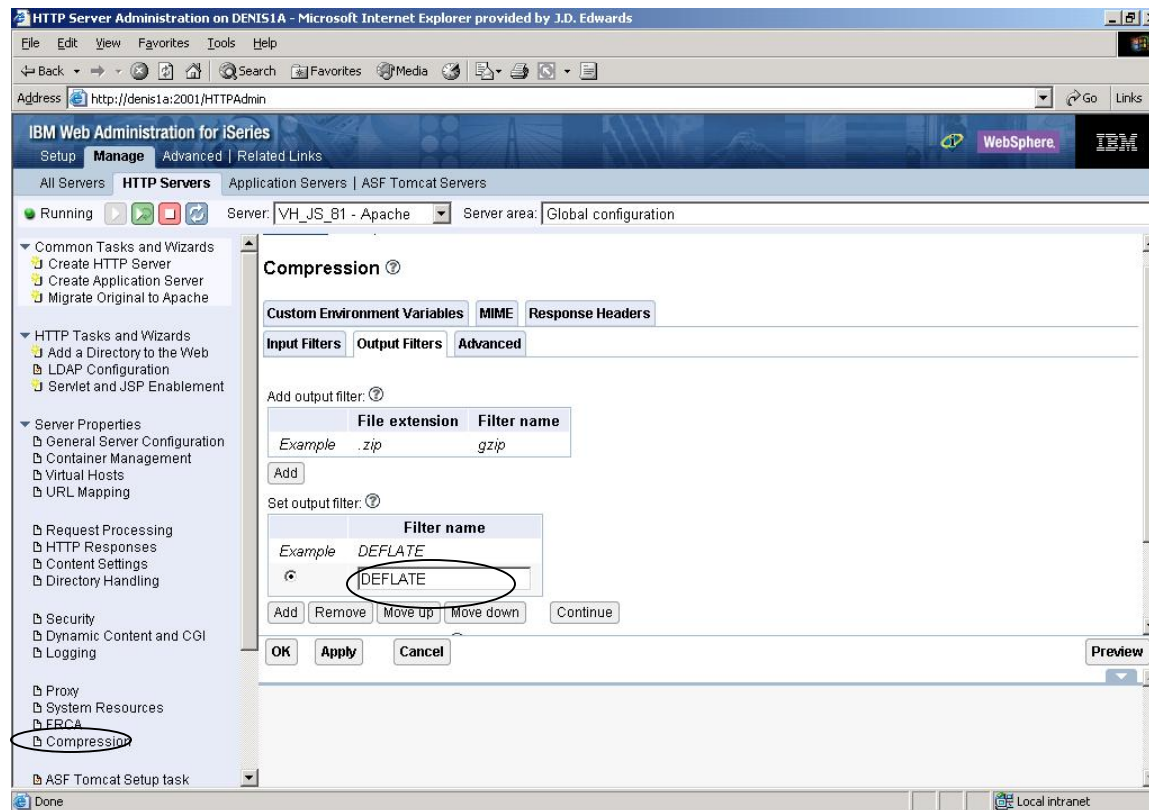


Figure 7 – Enable compression for the Apache HTTP server

There are two steps to enable compression:

1. As seen in <Figure 7>, select “Compression” from the left pane, then click the “Output Filters” tab. In the “Set output filter” section, click “Add.” enter “Deflate” as the filter name, and click “Apply.”

Alternatively, you could edit the configuration file by adding the following lines:

```
LoadModule deflate_module /QSYS.LIB/QHTTPSVR.LIB/QZSRCORE.SRVPGM
SetOutputFilter DEFLATE
```

2. Next, as shown in <Figure 8>, in the “Server area” section, select the directory path that ends with “webclient.war” that was added during the EnterpriseOne Java Application Server configuration. Click the “Add” button in the “Add output filter by MIME type” section, then enter a MIME type of “text/html” or select it from the list. You will also need to enter a filter name of “DEFLATE” and click the “Apply” button.

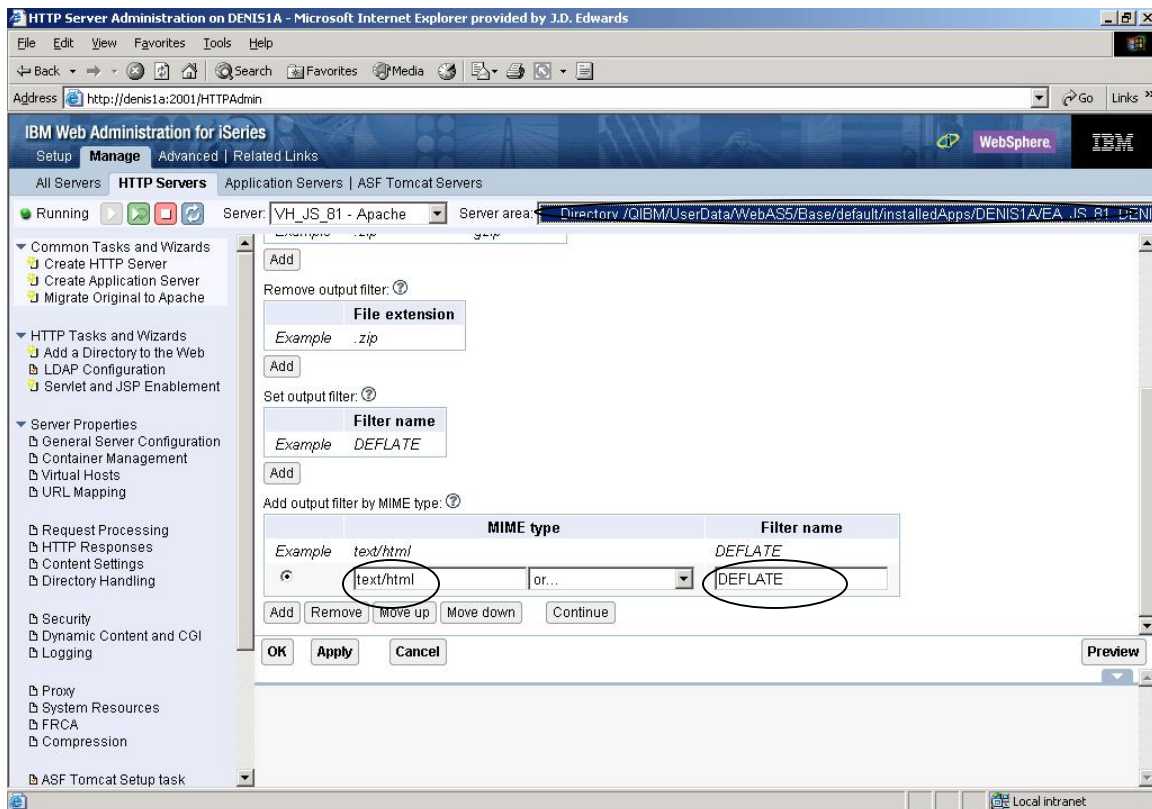


Figure 8 – Set the output filter for text in the JAS directories.

Alternatively, you could change the directory directives to include the filter statement for text by adding the line shown in bold, below, to the httpd.conf file:

```
<Directory /QIBM/UserData/WebAS5/Base/default/installedApps/<sys>/<ent-app>/webclient.war>
Order Deny,Allow
Allow From all
AddOutputFilterByType DEFLATE text/html
```

</Directory>

where <sys> is the name of the iSeries server and <ent-app> is the name of the enterprise application.

### **WebSphere Application Server Tuning**

The WebSphere Application Server 5.0 (5722-WS5) product is the recommended application server for running the EnterpriseOne 8.11 Java Application Server (referred to as JAS). This can be a bit confusing since both products have the term “Application Server” in the name. To clarify, the WebSphere Application Server is a general-purpose application server capable of running many different types of applications with the JD Edwards JAS code being only one of them. JAS is a set of Java servlets and Java Server Pages (JSPs) that are used to serve as the EnterpriseOne application interfaces to Web users.

WebSphere Application Server 5.0 on iSeries has two versions: Base and Network Deployment (ND). JAS runs with either version. Network Deployment is usually implemented for high availability utilizing horizontal clustering or scalability via vertical clustering. Lab tests conducted with ND using vertical clustering with two clones did not result in any performance improvements. Some customers with high JVM memory requirements might achieve economies of scale with vertical clustering. The iSeries servers also support WebSphere Application Server Express; however, JAS does not support this version of WebSphere.

### ***Measure and Analyze***

The WebSphere Application Server, in general, and the JAS server code, specifically, can be analyzed in a variety of ways. The familiar iSeries tools, such as Performance Monitor and Performance Explorer, can be used to analyze WebSphere performance. Additionally, OS/400 (named i5/OS™ with V5R3) has system commands for looking at the Java Virtual Machine (JVM) which is the WebSphere job that runs the JAS server code. The ANZJVM and DMPJVM CL commands can be used to look at heap usage, garbage collection, thread stacks, and object statistics. JAS has a built-in tool, the System Administration Workbench (SAW), and WebSphere has a built-in tool, the Tivoli Performance Viewer, each of which can be used to analyze specific internal resources (such as business function response times for JAS and servlet and thread statistics for WebSphere).

The System Administration Workbench is accessible via the URL:

**http://<sys>:<port>/jde/saw/sawJas.html**

where <sys> is the name of the JAS server and <port> is the port on which the JAS server is listening.

Additional details on the SAW functionality are provided in the Enterprise Server tuning section.

The Tivoli Performance Viewer is a useful tool to monitor such things as the number of threads used and the amount of memory consumed. As shown in <Figure 9>, a convenient graphical user interface (GUI) monitors WebSphere resource usage and information so you can adjust the thread settings as described in Tip 3 below.

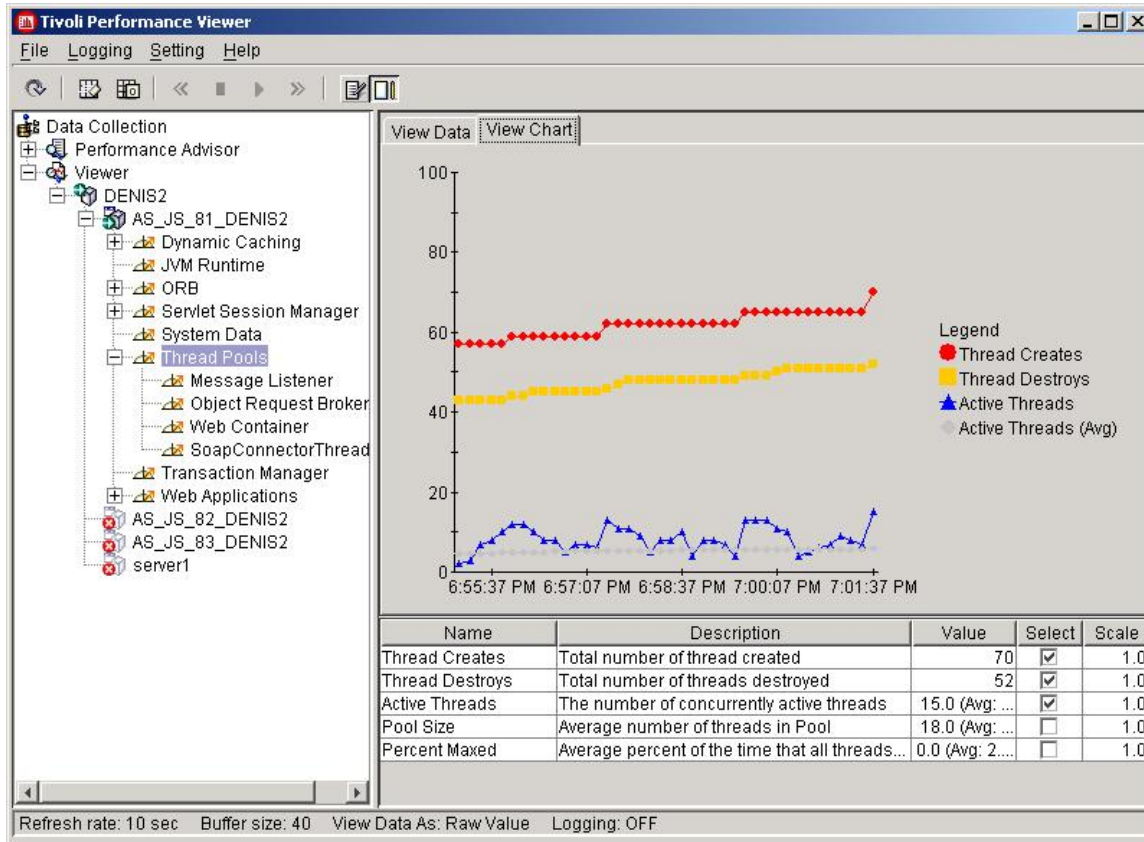


Figure 9 – Tivoli Performance Viewer

Although WebSphere 5.0 has replaced the thick client administrative console of previous WebSphere releases with a Web interface, the Tivoli Performance Viewer requires a client. To install the client, run the WebSphere Application Server installation process on the target PC, choose a custom install and only select the performance monitoring tools. If installing the Tivoli Performance Viewer from the JD Edwards Technology Foundation CDs, it can be found on the CD entitled "IBM WebSphere Application Server V 5.0 for Windows 2000 Enterprise Application Server". With that CD, run install.exe from the /windows/WAS50 directory. (There is also an install.exe in the /windows directory which is not the correct executable to run because it does not offer the "custom" install.)

To run the performance viewer tool, determine which port the application server for JAS is "listening" for RMI connections. The ports for the application are found by doing the following in the WebSphere Administrative Console: expand **Servers** > click **Application Servers** > click *Your Server Name* > click **End Points** > click **BOOTSTRAP\_ADDRESS**. Note the port listed. This is the port to use when starting Tivoli Performance Viewer. From the command line run:

```
tperviewer <sys> <port> RMI
```

where <sys> is the name of the iSeries server and <port> is the RMI port for the application server.

When a message is displayed indicating that other application servers are not listening on the specified port, click OK. While users are on the system, the viewer displays current statistics on resource usage within WebSphere, and the Performance Advisor provides suggestions on other WebSphere settings to change. The Performance Advisor may provide specific environmental suggestions which should be tested in addition to the tips described below.

### ***Adjust***

Changes to the WebSphere 5.0 configuration are made through the WebSphere Administrative Console. The WebSphere 5.0 Administrative Console uses a Web interface. The default instance of WebSphere is accessed through the following URL:

http://<sys>:9090/admin

where <sys> is the name of the iSeries server that is running WebSphere.

The administrative console presents a user id prompt, which is used to track the changes entered into the administrative console. It does not have to match an iSeries user profile.

### ***1) Adjust the heap size for the JVM***

On iSeries servers, Java performs memory management for applications running within the JVM. It allocates memory on behalf of the application in a single heap space and cleans up unused memory via the garbage collection process. WebSphere, as a Java application, has its memory controlled by the JVM. When the application server running JAS is started through WebSphere, Java allocates memory and manages it on behalf of JAS. Setting the initial heap size for the JVM controls the amount of memory used by WebSphere.

Because Java objects are cleaned up by the JVM, the iSeries JVM also uses the initial heap size as the indicator for when to perform a garbage collection. Therefore, setting the initial heap size to an appropriate value for the workload is very important. Setting the heap size too small causes frequent garbage collections, which consumes CPU resources. Setting the heap size too large causes excessive memory usage, paging, and can result in a higher cache miss ratio because the heap is so large. The initial heap size setting for JAS is based on the number of active users. The guideline derived from lab tests is 1.5 megabytes per active user. The active user count is defined as the number of concurrent users who are hitting OK or function keys two to three times per minute. The appropriate setting may vary depending on the type of applications being run as well as the number of users. A prudent minimum initial heap size is 128 MB of memory even for a very low user count.

The maximum heap size controls how large the JVM heap is allowed to grow. The recommendation is to leave this field blank, meaning the system will control the maximum heap size. In certain cases on memory-constrained systems, setting the maximum heap size may help the overall system performance.

To change the JVM settings for the application server, follow these steps: expand **Servers** > click **Application Servers** > click *Your Server Name* > click **Process Definition** > click **JVM Settings**. Changing

the JVM settings for a 200-user workload is shown in <Figure 10> where you can see that the initial heap size is set to 200 users times 1.5 megabytes (which equals 300 megabytes). A maximum heap size has not been set.

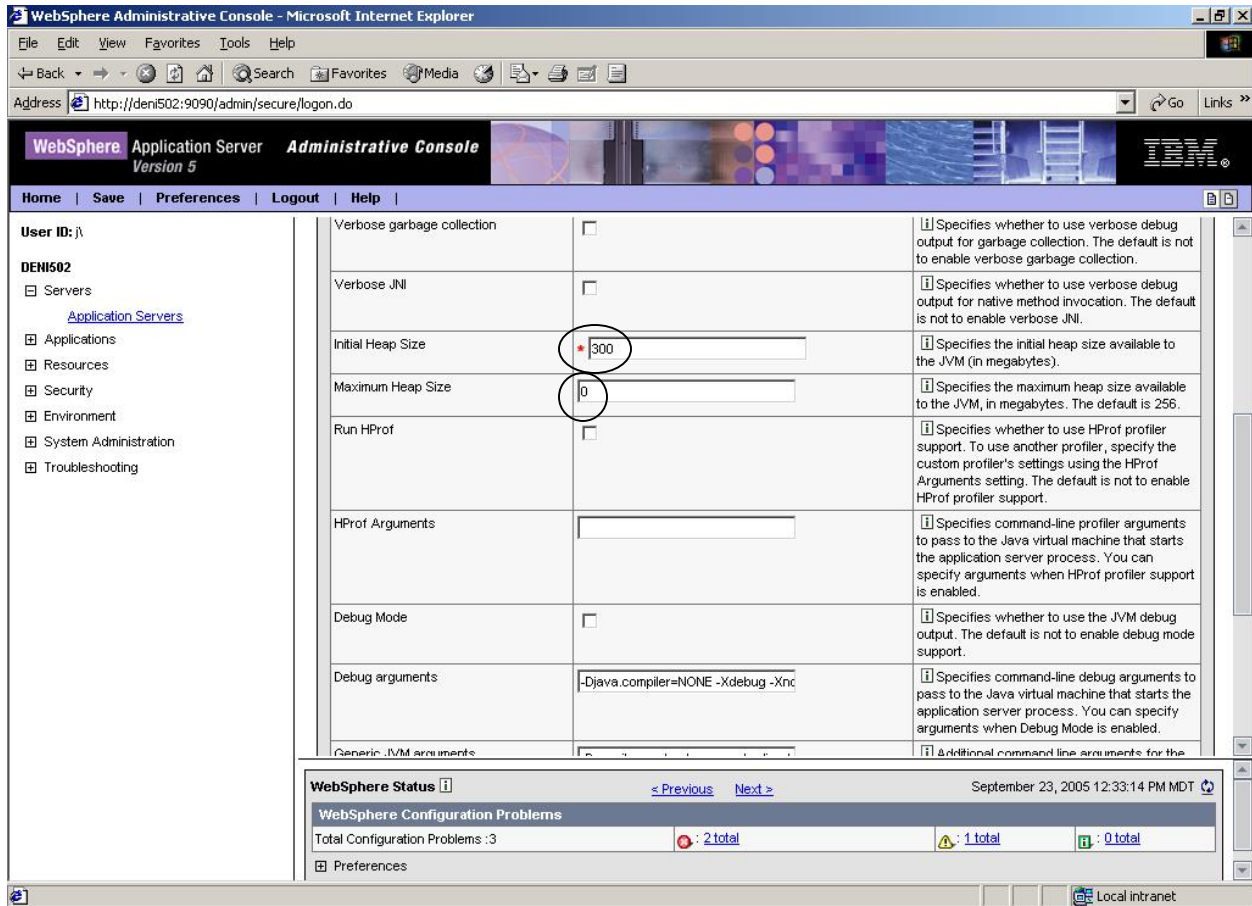


Figure 10 – Set the initial heap size for WebSphere

Heap size and garbage collection information may be checked with the DMPJVM command. This command produces a pool file that contains information on the current heap size, the number of garbage collections performed, and the duration of the last garbage collection.

**Additional Tip:** The WebSphere administrative server (JVM) automatically restarts any application server job that fails. This is accomplished by periodically checking the job (ping), and waiting for a reply. If the iSeries server is underpowered or heavily loaded, the application server job may not respond in time, causing the administrative server to inadvertently end and restart the application server job. To prevent this from happening, do the following: expand **Servers** > click **Application Servers** > click *Your Server Name* > click **Process Definition** > click **Monitoring Policy** and increase the “Ping timeout” value from the default of 60 to 300.

## ***2) Adjust WebSphere subsystem settings***

As shipped, the WebSphere subsystem runs in the base memory pool on the system. In many cases, it is helpful to divide workloads into separate shared memory pools. Putting WebSphere into a separate shared memory pool isolates it from other workloads and ensures it does not compete with other jobs for memory. Using a shared memory pool also provides flexibility because memory can be easily and dynamically moved from one pool to another. This is done automatically by the system, through the OS/400 command line, or through a scheduled command.

There are only two steps to setting up WebSphere to run in a separate pool:

1. The shared memory pool must be set up. While shared memory pools already exist on the system, no memory is allocated to them. The shared memory pool for WebSphere running JAS with 200 users requires about four times the initial heap size (300 megabytes) or 1.2 gigabytes of memory plus some overhead memory that Websphere requires outside of the heap. A minimum amount of additional memory for WebSphere is 1 GB. Caution should be used here if additional JAS servers are established because these memory requirements are additive. Also, the memory pool allows a specific number of threads to run. With 200 users plus overhead for other threads, such as JDBC connections, the recommendation is to start with 300 threads, which is referred to as the activity level. Allowing the system to dynamically adjust the paging characteristics of the storage pool for optimum performance is also recommended. Setting the paging option to \*CALC allows the system to make adjustments as necessary. To change a shared memory pool \*SHRPOOL1 for WebSphere issue the following command:

```
CHGSHRPOOL POOL(*SHRPOOL1) SIZE(x) ACTLVL(y) PAGING(*CALC) TEXT('WebSphere  
Share Pool')
```

where x is the total memory allocated and y is the max threads based on the user count.

2. The subsystem must be changed to make use of the shared memory pool. That is, the WebSphere subsystem must be able to use \*SHRPOOL1 and will run with a pool ID of 2. If the subsystem is to run exclusively in the shared pool, run the command:

```
CHGSBSD SBSD(QEJBAS5/QEJBAS5) POOLS((1 *SHRPOOL1))
```

If the subsystem is to continue using memory in \*base, as well as to use memory in the new pool, change the subsystem by running the command:

```
CHGSBSD SBSD(QEJBAS5/QEJBAS5) POOLS((2 *SHRPOOL1))
```

Next, each routing entry that needs to point to the new pool must be changed:

Enter WRKSBS on the command line, then take option 5 to display the subsystem description, then option 7 to work with routing entries. Note the entries to be changed and enter:

```
CHGRTGE SBSD(QEJBAS5) SEQNBR(999) POOLID(2) for each entry to be changed.
```

The subsystem needs to be restarted after these changes are made. Then use the command WRKSYSSTS to look at the new pool. Validate the activity level setting for the shared memory pool by using the WRKSYSSTS command, pressing the F11 key, and verifying that the “wait to ineligible” value (Wait->Inel) remains at zero. Note: iSeries Navigator contains a graphical interface to look at the shared pools.

### ***3) Adjust the number of threads running in the JVM***

WebSphere maintains a pool of threads that processes requests from Web clients. Two things limit the pool: a minimum number of threads always started and a maximum number of threads that the pool can grow. Set the minimum size to handle the average load on the system and set the maximum size to handle the peak load. A good place to start is to set the minimum number of threads to 10% of the number of users and the maximum number of threads to equal the actual number of users.

The pool settings limit the number of users that can be handled concurrently. Setting the minimum too high causes additional memory usage. Setting the maximum too low causes users to wait, increasing response times.

The thread settings can be manipulated by doing the following: expand **Servers** > click **Application Servers** > click *Your Server Name* > click **Web Container** > click **Thread Pool**. Change the “Minimum Size” to the average workload and the “Maximum Size” to the peak workload plus one (necessary overhead for JAS). Check the “Is Growable” checkbox so that thread allocations beyond the maximum are allowed.

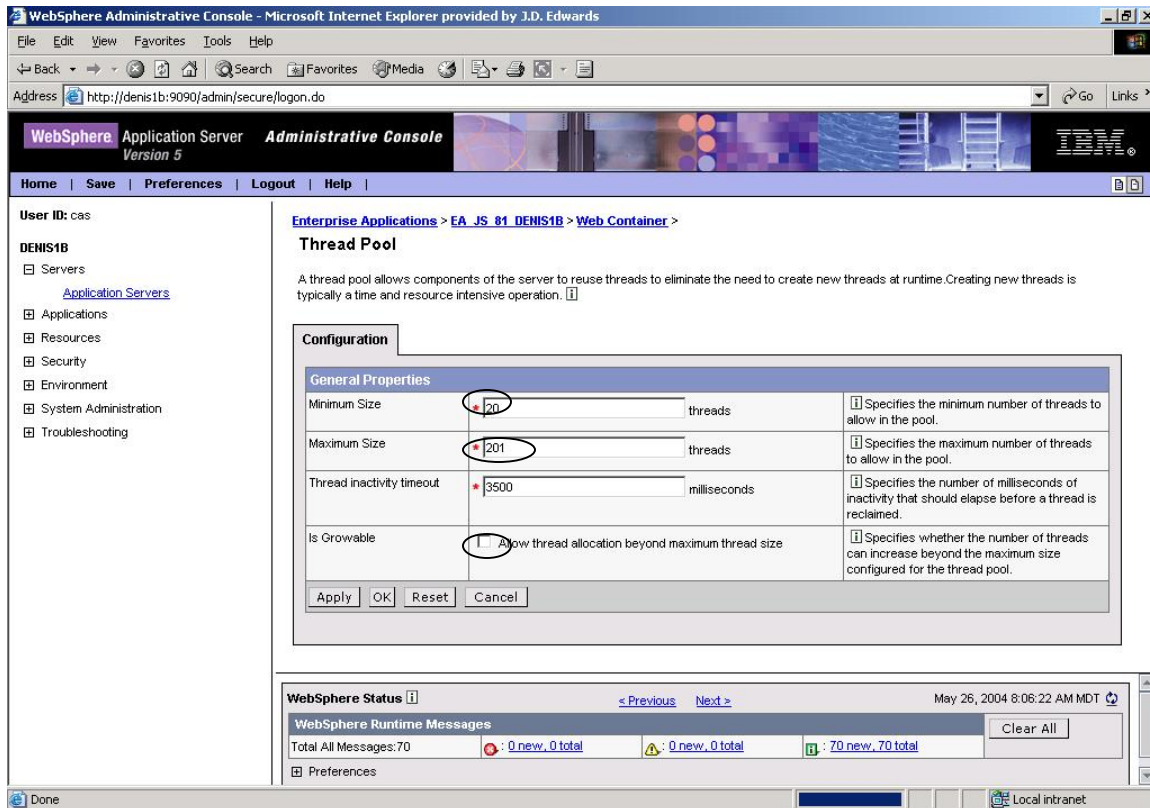


Figure 11 – Set the number of threads for WebSphere

As an example, <Figure 11> shows the recommended values for a maximum workload of 200 users. The minimum size is set to 10% of the total number of users (or 20 threads). The maximum is set to 200+1 (or 201 threads). Also, the checkbox “Allow thread allocation beyond maximum size” needs to be checked.

**Additional Tip:** Timestamps within the WebSphere and JAS logs are output in Greenwich Mean Time (GMT) by default. To change the timestamps to the current time zone, change the Java system settings. From the OS/400 command line, enter the following command:

```
edtf '/qibm/userdata/java400/systemdefault.properties'
```

Enter the following line to change the time zone:

```
os400.user.timezone=xxx
```

where xxx is the correct time zone, such as MST for Mountain Standard Time

### **EnterpriseOne Application Tuning**

JD Edwards EnterpriseOne provides a complete suite of modular, integrated industry-specific business applications. The applications run via a Web interface, or a Windows interface for a development user. EnterpriseOne is flexible, permitting the database, application logic, and user interface code to run entirely on one system or each of these can run on separate systems. Although some of the tuning recommendations apply to the Windows interface, the information in the following section is intended for, and applies specifically to, the EnterpriseOne Web interface.

#### ***Measure and Analyze***

Similar to the Apache HTTP Server and WebSphere Application Server, EnterpriseOne has built-in tools for monitoring the performance of the user's interaction via the Web interface. The SAW tool allows an administrator to monitor the performance of the JAS server and its interactions with the other components of the application and system. SAW for JAS provides various views for monitoring:

- System Summary – shows a high-level view of the heap, including number of users and business functions.
- User List – shows the number of users signed on and active.
- CallObject Info – lists the business functions that have been run and statistics about the calls.
- Work with Log files – contains the logs and ini files.
- Environment – lists the various operating system and Java properties.
- Config – shows some of the jas.ini configuration settings.
- JDENET pool – lists all of the connections to the enterprise server.
- Work with JDBCj – contains details of the JDBC connections and caches the JAS server maintains.
- Virtual Clients – lists all virtual client sessions and activity within them.
- Outstanding Requests – shows any requests currently waiting for the enterprise server to return.
- Thread List – shows all threads running within JAS and their current activity.

Each of these views is useful for tracking various aspects of the JAS server and its configuration. For example, the System Summary view shows how many users are currently active on the system and how much memory they are using. The other views provide additional relevant information. Access SAW for JAS servers with the following URL:

`http://<sys>:<port>/jde/saw/sawJAS.html`

where <sys> is the name of the JAS server and <port> is the port on which the JAS server is listening.

The application server portion of EnterpriseOne also contains a SAW function with useful information about the jobs running on the enterprise server. Access it by running the SAW command from an OS/400 command line. (The system library, such as E811SYS, must be in the library list for this command to work properly). A Web version of the enterprise SAW function is also available by using the Web URL:

`http://<sys>:<port>/jde/saw/sawEnt.html`

When prompted for host and port, enter the name of the enterprise server for the host and the port of the enterprise server (6013 by default).

SAW for enterprise servers also contains multiple views. The server summary view displays information about each of the server jobs, identifies which jobs have a backlog of requests, and allows tracing to be turned on dynamically.

### ***Adjust***

The JD Edwards EnterpriseOne enterprise server provides configurable parameters through the jde.ini file in the IFS. On the iSeries command line, enter the command: `edtf '/E811sys/ini/jde.ini'`

In the tuning suggestions that follow, this is referred to as the jde.ini file.

Additionally, the EnterpriseOne JAS server provides configurable parameters through the jas.ini file in the IFS. On the iSeries command line, enter the command:

`edtf '/QIBM/UserData/WebAS5/Base/default/installedApps/<sys>/<ent-app>/webclient.war/WEB-INF/jas.ini'`

where <sys> is the name of the iSeries server and <ent-app> is the name of the enterprise application. In the tuning suggestions that follow, this is referred to as the jas.ini file.

### ***1) Kernel Settings***

The EnterpriseOne enterprise server makes use of various jobs to accomplish work for a user. Security kernels authenticate the credentials for Web users. Call object kernels run business functions on behalf of Web users. The JAS server communicates with the enterprise server's network kernels. The jde.ini file provides two types of settings related to these jobs. The first setting is the number of jobs to prestart when EnterpriseOne services are started. Additional jobs are started if a request arrives and all jobs are currently active—but only up to the second setting, which is the maximum number of jobs allowed to run.

Call object kernel jobs are capable of handling the requests for multiple users. Because each job caches information and uses system resources, starting too many jobs is not efficient. However, the presence of too few call object kernels causes users to wait for business functions to run and increases response times. Lab tests have shown that having 10-15 users allocated to each call object kernel provides a good balance of

resource utilization and response time. Probably, 15 users is a good place to start. Use a tool such as SAW for enterprise servers to provide feedback to fine-tune this setting iteratively.

Starting an optimal number of jobs is true for security and network kernels as well. One security kernel job for every 100 users and one network job for every 20 call object kernel jobs (with a minimum of two jobs) provides the best performance. These settings are changed in the JDE INI file.

Network kernel jobs:

```
[JDENET]
maxNetProcesses=x (default is 1)
```

Security kernel jobs:

```
[JDENET_KERNEL_DEF4]
maxNumberOfProcesses=x (default is 1)
numberOfAutoStartProcess=x (default is 0)
```

Call object kernel jobs:

```
[JDENET_KERNEL_DEF6]
maxNumberOfProcesses=x (default is 10)
numberOfAutoStartProcess=x (default is 0)
```

## ***2) Check kernel health***

While the enterprise server is running, it passes messages between the network jobs and the kernel jobs. By default, every 10th time it sends or receives a message, the network job verifies that each call object kernel is running. Increasing this setting reduces the amount of overhead within the enterprise server and improves overall response time; the trade-off is better performance versus prompt knowledge of a kernel problem.

Monitoring kernel jobs is important, but with a heavy transaction load and a large number of call object kernel jobs, consider increasing this number to have it check for kernel health less frequently. Alternatively, consider reducing this number while testing new modifications.

In the jde.ini file add a new line to the [JDENET] section. Add a line with the setting checkKrnlHealth and enter a value. During normal, stable operations, setting the network job to check every 5000 or 10000 times it sends or receives a message is a reasonable place to start:

```
[JDENET]
checkKrnlHealth=x
```

### ***3) Maximum users for the Java Server***

The maximum users setting in the JAS acts as a threshold. Once the maximum number of users has been reached, no additional users are allowed to login to EnterpriseOne until another active user logs out.

Set the MAXUser value equal to the maximum number of users to be concurrently connected to the JAS. In the jas.ini file's OWWEB section, replace the "x" to set the maximum number of users expected:

```
[OWWEB]
MAXUser=x
```

Go into Web SAW under the User List pull-down to verify that the MAXUser setting is high enough by comparing it to the number of users connected during peak activity.

### ***4) JDENET Connections for the Java Server***

The connections between the EnterpriseOne Java server and the enterprise server are TCP/IP-based and are referred to as JDENET connections. The number of JDENET connections should be the same as the number of users being run on the Java server. To increase the number of connections, add a line with a maxPoolSize to the [JDENET] section of the jas.ini file and replace the "x" with the user count:

```
[JDENET]
maxPoolSize=x
```

Go into Web SAW under the JDENet Pool pull-down to verify the setting. During peak activity, monitor the waiting column. This number should usually be "0" to ensure that no latency problems exist for JDENET connections.

## **iSeries Database Tuning**

DB2 Universal Database (UDB) for iSeries is an advanced, 64-bit Relational Database Management System (RDBMS). The tight integration of DB2 UDB with OS/400 provides unique attributes to this RDBMS. Single Level Store and the object-based operating system, OS/400, minimize the effort required to manage the database, while maintaining its mainframe-like reliability and security. Automation of many of the common Database Administrator (DBA) tasks required by other RDBMS's is a cornerstone of DB2 UDB for iSeries lower Total Cost of Ownership (TCO). Because DB2 UDB for iSeries is part of OS/400, many of the tools used to analyze the database are also part of OS/400's command set.

The EnterpriseOne environment performs the majority of its database access through three interfaces:

- ODBC (Open Database Connectivity)
- JDBC (Java Database Connectivity)
- EDRS (Extended Dynamic Remote Support)

Connection Type	Used by	Job(s)	Subsystem
ODBC	Developer fat client	QZDASOINIT	QUSRWRK
Toolbox JDBC	JAS running on machine other than database server	QZDASOINIT	QUSRWRK
Native JDBC	JAS running on database server	QSQSRVR	QSYSWRK
Local EDRS	EnterpriseOne application server on same system as database	EDRSQL interface (no DB job visible in wrkactjob)	
Remote EDRS	EnterpriseOne application server on different system than database	QXDARECVR (one job for each connection) QXDAEDRSQL (1 job)	QSYSWRK

*Table 1 – Database connection types*

Each of these interfaces is used by different parts of EnterpriseOne, and <Table 1> shows the different database interfaces, the parts of EnterpriseOne that use those interfaces, and the jobs and subsystems that execute the SQL requests for these interfaces.

***Measure and Analyze***

The iSeries Performance Monitor and Performance Explorer utilities can be used to monitor the overall system and database performance. To look more closely at the database component itself, use the database monitor tool. The database monitor can be started through iSeries system commands or through iSeries Navigator. The recommendation is to use the iSeries Navigator interface.

After starting iSeries Navigator, expand the system, expand the “Databases” section, and then expand the local directory (typically this will be the system name) as shown in <Figure 12>. Right-click “SQL Performance Monitors”, select “New” and then “Detailed” to start the database monitor. In the new window, on the “General” tab, give the monitor job a name, and a library for storing the data, and on the “Monitored Jobs” tab, select the “All” radio button (unless specific database jobs have been identified). Then click OK. The monitor job will be displayed in the right pane of the window.

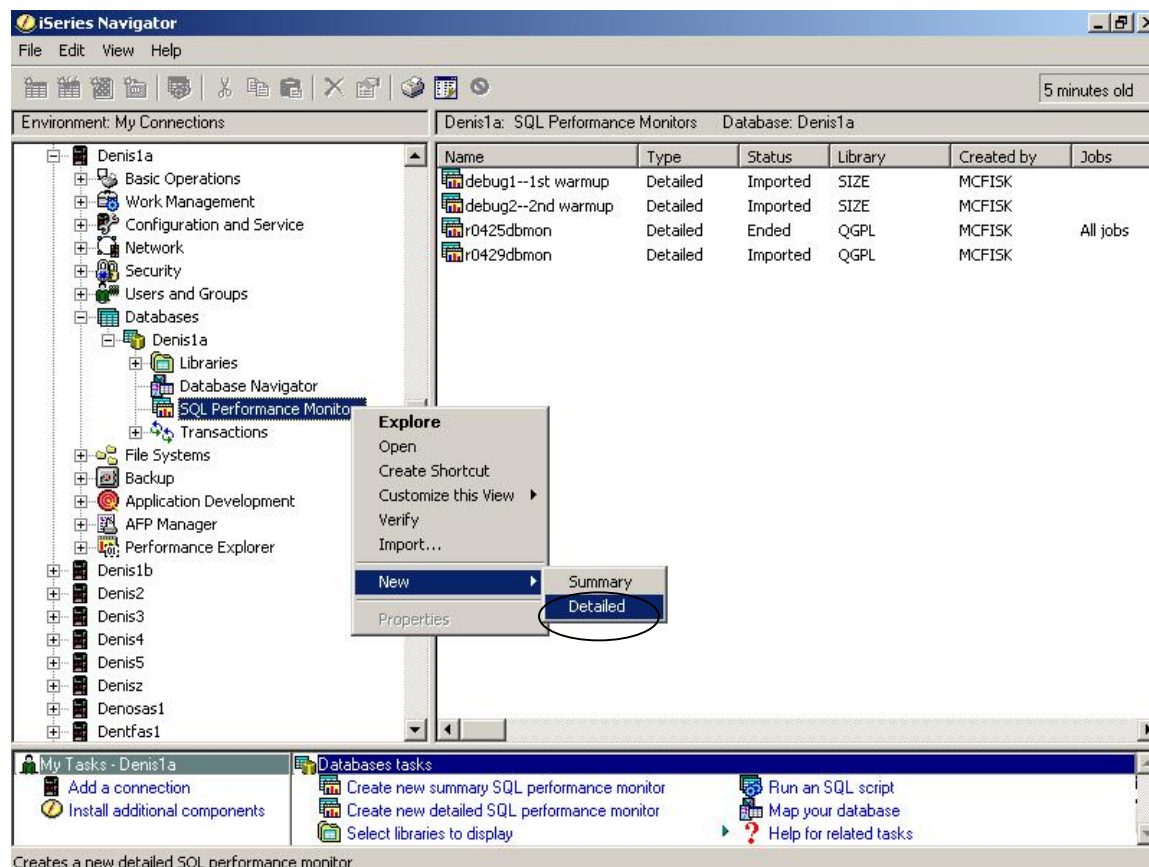


Figure 12 – Start the SQL performance monitor through iSeries Navigator.

The monitor will impact the end-user performance while it is running; therefore, it should be run for a short period of time (10-20 minutes). Capture a time period when the system is running the workload of greatest concern, wait for the time period, then right-click the job in the right pane and select “End”.

Alternatively, the following commands can be used to start and stop the database monitor using iSeries system commands:

```
STRDBMON OUTFILE(QGPL/R0429DBMON) JOB(*ALL) TYPE(*DETAIL) COMMENT('April 29 DBMON information')  
ENDDBMON JOB(*ALL)
```

Database monitor information captured in this way can be imported into the iSeries Navigator interface. Right-click “SQL Performance Monitors” as shown in <Figure 1> and select import. Enter the name of the monitor and the library containing the data.

### Adjust

DB2 UDB on iSeries has several settings that can be changed to improve the performance of the EnterpriseOne Web client environment. Some of these are operating system settings, while others are settings within the Web client configuration files.

***1) Improve database access performance by creating additional indexes***

Using the information collected by the database monitor as (described in the Analyze and Measure section above), iSeries Navigator can perform various queries on this information. These queries can be used to identify many things: temporary indexes that the system is creating for SQL statements, other SQL statements that require sorting, or long running queries. In addition, iSeries Navigator has a function to explain how specific SQL statements are being executed and another function to automatically create a new index to improve the performance of a query.

Indexes created through the iSeries Navigator interface should only be used temporarily to test the index and see if it improves the performance of the query. Indexes that help performance should be deleted and then recreated through the EnterpriseOne Table Design Utility. Using EnterpriseOne to create indexes ensures that they are maintained by the application and will not be lost when the application is upgraded. NOTE: Additional information on database monitor is found in the Redbook entitled, *Using AS/400 Database Monitor and Visual Explain to Identify and Tune SQL Queries*, <http://www.redbooks.ibm.com/redpapers/pdfs/redp0502.pdf>

***2) Tune the number of database connections for the system***

OS/400 maintains prestarted jobs for database connections. A prestart job starts before a work request is received, either when the subsystem starts or as a result of the Start Prestart Jobs (STRPJ) command. Prestart jobs start from a prestart job entry in the subsystem description. The prestart job entry specifies properties such as what program to run in the prestart job, the memory pool in which the prestart job runs, the number of jobs to start, the threshold at which new jobs will be started, the number of times the job is reused, and the number of additional jobs to start.

The EnterpriseOne Web client environment makes use of database prestart jobs. If a single system is running with the WebSphere application server, the EnterpriseOne application server, and the EnterpriseOne database (All-In-One or AIO), then QSQRVR prestart jobs are used. If the WebSphere application server is on a separate system (Virtual Three-Tier or V3T), QZDASOINIT prestart jobs are used. All Windows clients use ODBC, which requires QZDASOINIT jobs. Regardless of the type of environment and the type of prestart job, the process of tuning the number of database connections for the system is consistent.

The system maintains the prestart jobs and starts additional jobs, if necessary. In addition, every five minutes, OS/400 determines whether there is an excessive number of prestart jobs that are not being used. If there is, the operating system gradually starts ending the extra jobs until they are reduced to the number of jobs specified in the initial start setting. Starting and stopping these jobs can cause extra activity as the system attempts to maintain the proper number of jobs. To see the current settings on the database system, use the following command and select option 10: DSPSBSD QUSRWRK

This will display the various prestart jobs for the subsystem, then use option 5 for the program QZDASOINIT. (For an AIO environment, substitute QSYSWRK for the subsystem name and QSQSRVR for the job name in this example.) The parameter, Maximum number of uses, should be left at 200 (the shipped value). Allowing the QZDASOINIT jobs to be reused 200 times minimizes the number of jobs starting and stopping (specified with the MAXUSE parameter).

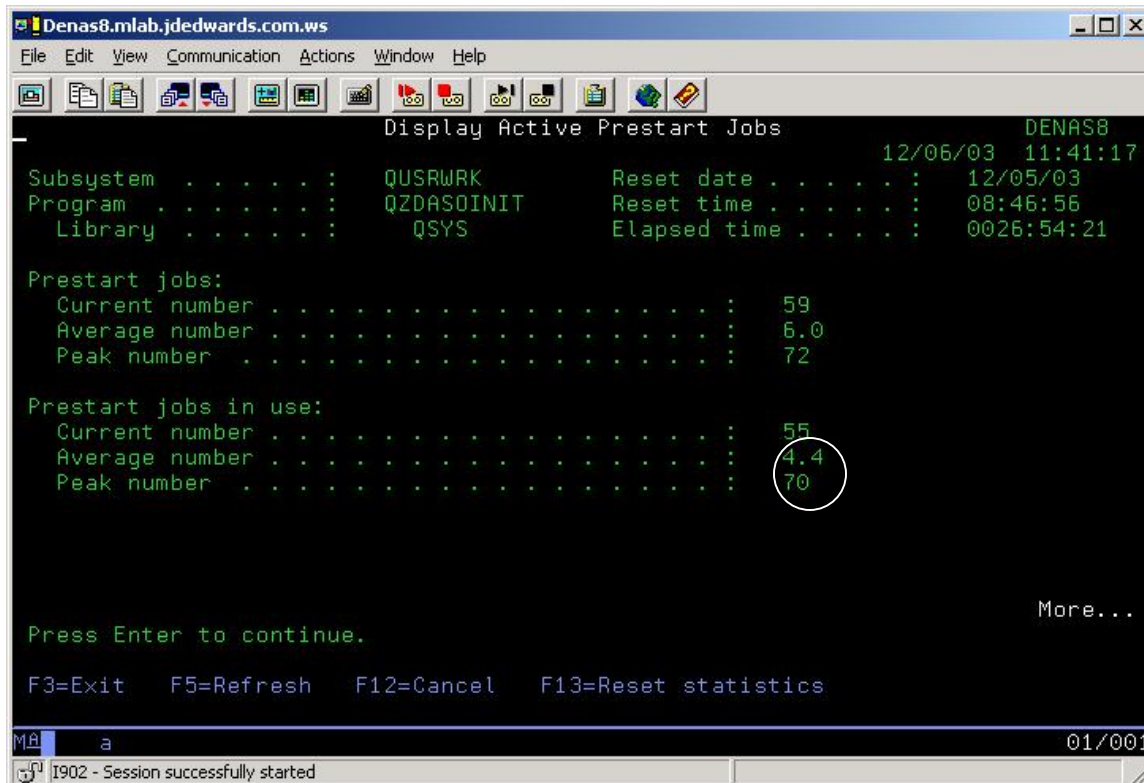


Figure 13 – Use the DSPACTPJ command to determine the average and peak number of jobs

Determine the number of prestart jobs currently being used during peak periods. Use the command DSPACTPJ as shown in <Figure 13> to show the current and peak number of prestart jobs. The command to use is: DSPACTPJ SBS(QUSRWRK) PGM(QZDASOINIT)

The key number is the peak number of prestart jobs in use. Use this number to make changes to the prestart job. By setting the initial number of jobs to start so that it is equal to the peak number of jobs, the additional work of starting new jobs is avoided. The jobs will be started when the subsystem starts and be available as the connection requests are made. Use the CHGPJE command to make the necessary changes for the environment by entering the command and pressing F4 to enter additional parameters as shown in <Figure 14>. Alternatively, enter the complete command:

CHGPJE SBS(QUSRWRK) PGM(QZDASOINIT) INLJOBS(70) THRESHOLD(5) ADLJOBS(10)



***3) Tune the number of database connections for the Web client environment***

The EnterpriseOne JAS uses JDBC connections to handle its database work. A JDBC connection is made for a specific iSeries user profile with a given set of properties. EnterpriseOne makes use of proxy profiles that allow many EnterpriseOne users to access the iSeries server through a common user profile. Because these proxy user profiles are used, EnterpriseOne creates pools of JDBC connections that many users can share. A connection pool is started and maintained by JAS for every connection type, where a connection type is the proxy profile and a set of connection properties.

The settings for the JDBC connections are maintained in the JDBj.ini file, which is located in the WEB-INF directory. Typically, this will be:

```
/QIBM/UserData/WebAS5/Base/Default/InstalledApps/<sys>/<ent-app>/webclient.war/WEB-INF
```

where <sys> is the name of the iSeries system and <ent-app> is the name of the enterprise application installed in the WebSphere Application Server (or JVM).

To determine the number of JDBC connections required for the pool, look at the Web SAW utility provided with the EnterpriseOne Web client. Go to the screen "Work with JDBJ", and under View Statistics "Connection Pool", locate the field, # of entries (connections). Pay particular attention to the number of connections being used relative to the initial setting in the JDBj.ini file.

Similar to the prestart jobs, set the initial number of connections to match the number of connections being used. This provides the best performance. Make changes to the [CONNECTION POOL] section of the JDBj.ini file.

```
[CONNECTION POOL]
minConnection=5
maxConnection=x
initialConnection=y
poolGrowth=10
```

where "x" is equal to the number of Web clients and "y" is equal to the number of entries (connections) displayed in the Web SAW utility column, "Number of entries," on the Connection Pools Statistics screen. If the max connections value of "x" is reached, the next user will not be denied, but response times begin to slow.

The test environment in the lab showed that one connection for every 10 users is a good number for the initial connections setting and that a maximum connections setting of one connection for every user is also a good number. A minimum of five connections and pool growth of 10 connections were also used in the lab tests. These settings are a good place to start for tuning an environment. Then make adjustments by observing the Web SAW utility.

A consideration is the use of the Multiple Application Framework (MAF), which allows each user to open multiple windows and potentially run multiple applications simultaneously. It is difficult to predict the impact of

MAF on an environment as only a percentage of the users may utilize this function. It is safe to assume that the maximum number of connections should increase with the use of MAF.

If the EnterpriseOne environment is not making use of proxy profiles and each user is mapped to its own iSeries server user profile, the recommendations are considerably different. In this case, set the initial number of connections to "1," minimum connections to "1," pool growth to "1" and maximum connections to "10."

```
[CONNECTION POOL]
minConnection=1
maxConnection=10
initialConnection=1
poolGrowth=1
```

Because of MAF, the maximum number of connections is set to "10," which allows for a single user to have 10 applications open, all utilizing JDBC connections simultaneously.

#### ***4) Optimize the use of SQL Packages***

SQL packages are OS/400 objects that contain both the control structures and the access plans that are necessary to process SQL statements on the application server when running a distributed program, such as EnterpriseOne. Because SQL packages are shared resources, the information built when a statement is prepared is available to all the users of the package. This saves time, especially in an EnterpriseOne Web client environment where many of the users execute the same or similar queries.

All the types of database interfaces used by EnterpriseOne make use of SQL packages with the exception of the Native JDBC driver (used with AIO environments). There are two important considerations when using SQL packages in an EnterpriseOne Web client environment:

- Knowing when to delete SQL packages
- Ensuring SQL packages are being used effectively

Packages must be deleted when the underlying metadata for statements stored in the package has been changed. If a table, view, stored procedure, or other SQL object is altered, information in the package is not updated. Therefore, EnterpriseOne SQL packages should be deleted whenever significant changes have been made to the database, operating system, or hardware. Examples of significant changes are those that may cause a large amount of access plan rebuilds, such as applying a database group PTF or upgrading the operating system. Because SQL packages are recreated when the EnterpriseOne application is run, there is little harm in deleting them other than the first user's performance will be a little slower.

The JAS code creates a separate SQL package in each library it accesses with JDBC. It is more efficient to use only one SQL package for all JDBC access. This setting is in the JDBj.ini file in the [JDBj-RUNTIME

PROPERTIES] section. Add a new setting to force all JDBC access to use a single library for storing SQL packages:

```
[JDBJ-RUNTIME PROPERTIES]  
as400PackageLibrary=QRECOVERY
```

The above setting adds the JAS SQL packages to the same library already used by the EnterpriseOne kernel jobs. The SQL package JDBJxxx (where xxx is a system-generated suffix, typically BBA or BAA) is initially created during the Java serialized object generation. Once the installation is completed, delete this SQL package, because it contains installation-specific information that is not necessary for normal Web client sessions.

As stated earlier, EnterpriseOne kernel jobs use SQL packages created in the QRECOVERY library. Each time the enterprise server is ended and restarted, new SQL packages are built and associated with the new kernel jobs. Old SQL packages associated with kernel jobs that have ended are never used again and can be deleted to save disk space. When enterprise services are down, use the following commands to delete all the SQL packages discussed:

```
DLTSQLPKG SQLPKG(QRECOVERY/O*)  
DLTSQLPKG SQLPKG(QRECOVERY/T*)  
DLTSQLPKG SQLPKG(QRECOVERY/JDBJ*)
```

Never delete SQL packages starting with the letter "Q." These SQL packages are used by the system.

**Additional Tip:** OS/400 V5R1 (and higher) supports a call to the job termination signal handler for a job (if one is enabled) when you run the ENDJOB OPTION (\*IMMED) command. If you must use the ENDSBS OPTION(\*IMMED) command, it is recommended that you enable this support and set the amount of time available for handling the job termination signal to an appropriate value. To set the value, create a data area that specifies the number of seconds for the handler to wait. It is recommended that you initially set this value to 600 seconds if you use the \*IMMED option to end the QEJBAS5 subsystem or QEJBASND5 subsystem:

```
CRTDTAARA DTAARA(QSYS/QENDJOB LMT) TYPE(*DEC) LEN(5 0) VALUE(600)
```

The maximum value allowed is 3600. The minimum value allowed is 30.

**Appendix A – Software and Hardware Levels**

All tests were performed on i5/OS V5R3. It is highly recommended that a production environment be kept current with the JD Edwards informational APAR.

The link below is for the JD Edwards EnterpriseOne product and contains links to the informational APARs for V5R2 and V5R3: [ibm.com/servers/enable/site/events/jde\\_links.html](http://ibm.com/servers/enable/site/events/jde_links.html)

Tests were conducted with both an All-In-One (AIO) and a Virtual Three Tier (V3T) environment. <Figure 15> below shows the V3T environment and the components that ran on each system.

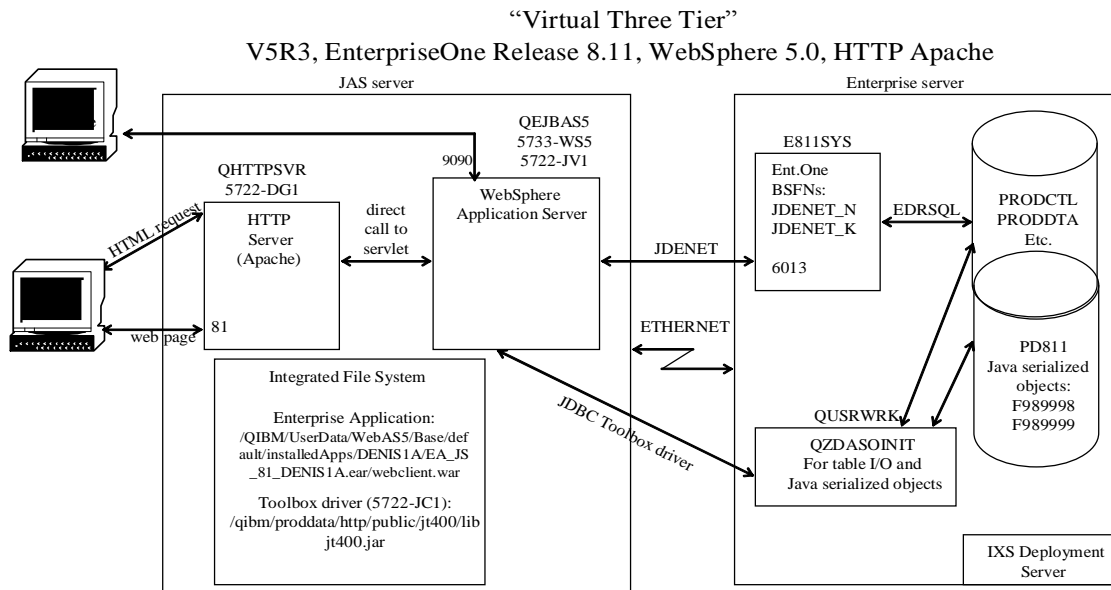


Figure 15 – Virtual Three Tier (V3T) configuration

The tables below show the hardware used during the tests to make the recommendations in this chapter. Tests were run on 2 Model 570s feature 8961. The following is the enterprise server (which was not partitioned):

Hardware for test environment	Description	Number or Amount
Processors	6000 cpw	2
Disk Drives, Raided	30 arms (70 GB drives)	1.7 TB
Memory		32 GB
I/O Card	100 Mb Ethernet	1

*Table 2 – iSeries Hardware Testing Environment (enterprise server)*

The Java server was an LPAR partition on a separate Model 570 system:

Hardware for test environment	Description	Number or Amount
Processors	6000 cpw	2
Disk Drives, Raided	4 arms (70 GB drives)	211 GB
Memory		12 GB
I/O Card	100 Mb Ethernet	1

*Table 3 – iSeries Hardware Testing Environment (web server)*

The table below shows the software levels used during the tests to make the recommendations in this document.

Software for test environment (WRKPTFGRP):	Description:	Service Pack:
OS/400	V5R3	Cum C5102530
Database	SF99503	Service Pack #5
EnterpriseOne	8.11 with 8.94	GA
HTTP	SF99099	Service Pack #6
WebSphere Advanced Base 5.0.2	SF99269	Service Pack #10
Java™	SF99269	Service Pack #6

*Table 4 – iSeries Software Testing Environment*

## **Appendix B – Configuration Files**

Following are the configuration files for HTTP and the EnterpriseOne Java and enterprise servers. (The WebSphere configuration settings are only edited through the Administrative Console, which does not offer a file listing.) The V3T environment is configured for 500 users.

### *HTTPD.CONF*

```
WebSpherePluginConfig /QIBM/UserData/WebAS5/Base/default/config/cells/plugin-cfg.xml
LoadModule ibm_app_server_http_module /QSYS.LIB/QEJBAS5.LIB/QSVTIHSAH.SRVPGM
# Configuration originally created by Create HTTP Server wizard on Thu Aug 11 15:51:30 MDT 2005
Alias /jde /QIBM/UserData/WebAS5/Base/default/installedapps/DENI507B/EA_JS_81_DENI507B.ear/webclient.war
Listen *:84
DocumentRoot /www/vh_jas_84/htdocs
Options -ExecCGI -FollowSymLinks -SymLinksIfOwnerMatch -Includes -IncludesNoExec -Indexes -MultiViews
#LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
#LogFormat "%{Cookie}n \"%r\" %t" cookie
#LogFormat "%{User-agent}i" agent
#LogFormat "%{Referer}i -> %U" referer
#LogFormat "%h %l %u %t \"%r\" %>s %b" common
#CustomLog logs/access_log combined
MaxKeepAliveRequests 0
KeepAlive On
ThreadsPerChild 50
LogLevel Error
LogMaint logs/access_log 7 0
LogMaint logs/error_log 7 0
SetEnvIf "User-Agent" "Mozilla/2" nokeepalive
SetEnvIf "User-Agent" "JDK/1\0" force-response-1.0
SetEnvIf "User-Agent" "Java/1\0" force-response-1.0
SetEnvIf "User-Agent" "RealPlayer 4\0" force-response-1.0
SetEnvIf "User-Agent" "MSIE 4\0b2;" nokeepalive
SetEnvIf "User-Agent" "MSIE 4\0b2;" force-response-1.0
<Directory />
    Order Deny,Allow
    Deny From all
</Directory>
<Directory /www/vh_jas_84/htdocs>
    Order Allow,Deny
    Allow From all
</Directory>
<Directory /QIBM/UserData/WebAS5/Base/default/installedapps/DENI507B/EA_JS_81_DENI507B.ear/webclient.war>
```

## IBM eServer iSeries Performance and Tuning Tips for the JD Edwards EnterpriseOne Web Server

### Page 34

```
Order Deny,Allow
Allow from all
</Directory>
<Directory /QIBM/UserData/WebAS5/Base/default/installedapps/DENI507B/EA_JS_81_DENI507B.ear/webclient.war/WEB-INF>
Order Deny,Allow
Deny from all
</Directory>
```

```
JAS.INI
# Please refer to the JAS Installation Guide for detailed information on
# this file.
#
#
[SECURITY]
DataSource=SY811
SecurityServer=DENI502
UseLogonCookie=FALSE
CookieLifeTime=7
SSOEnabled=FALSE
SSOUserIDOnly=FALSE
# set password only if SSOUserIDOnly is set to TRUE
Password=

[OWWEB]
PathCodes=('PD811')
MO QUEUE=/QIBM/UserData/WebAS5/Base/default/installedApps/DENI507A/EA_JS_84_DENI507A.ear/webclient.war/moqueue
MAXUser=1000
UseMOWinNTShare=FALSE
SystemDateFormat=MDE
SystemDateSeparator=/

# The HelpPath key determines where JAS looks for help files.
# Syntax: "http://[machine]/[path]".
# Note, if you specify "/jde/owhelp/", this machine is assumed.
HelpPath=/jde/owhelp/

# Library Name for the AS/400 STRJOURNAL stored procedure

[CACHE]
# These intervals are in milliseconds. For example, 60000ms = 1 minute
UserSession=2400000
CacheCheck=60000
```

```
[SERVER COMPONENTS]
#
# Do not modify this section
#
com.jdedwards.jas.UserManager
com.jdedwards.jas.JDBCProxy
com.jdedwards.jas.JDEORB
com.jdedwards.jas.DDValidation
com.jdedwards.jas.security.SecurityBroker
com.jdedwards.jas.UDCJDBC
com.jdedwards.jas.JDEUDCText
com.jdedwards.jas.JDEUpdates
com.jdedwards.jas.JDEQueries
com.jdedwards.jas.JDEOWDirect
com.jdedwards.jas.ServerQuery
com.jdedwards.jas.JDESignon
com.jdedwards.runtime.virtual.MAFComponentLoader
```

# IBM eServer iSeries Performance and Tuning Tips for the JD Edwards EnterpriseOne Web Server

## Page 35

### [JDENET]

# Enterprise Server JDENET port  
serviceNameConnect=6013

# JAS Server JDENET Listening port  
serviceNameListen=6013

# Maximum number of dedicated kernel types  
maxKernelRanges=1  
maxPoolSize=1000  
# Time out value for requests sent to the Enterprise Server  
enterpriseServerTimeout=900000

# Directory for JDENET temporary file  
tempFileDir=/tmp/InstallShield

# Clustered logical datasource retry interval  
ClusterRetryInterval=60000

### # Dedicated Kernel Definitions

[JDENET\_KERNEL\_DEF1]  
kernelName=XAPI Kernel  
processClassName=com.jdedwards.runtime.virtual.xapi.PublishedMessageProcessor  
startMessageRange=15001  
endMessageRange=15250  
maxNumberOfThread=3

### [SERVER]

# Enterprise Server and port providing Glossary Text information  
glossaryTextServer=DENI502:6013

# Encoding scheme  
# 1252- English and Western European  
# 932- Japanese  
# 950- Traditional Chinese  
# 936- Simplified Chinese  
# 949- Korean  
codePage=1252

### [LOGS]

# The STDERR and STDOUT keys must point to the same directory that was supplied for the  
# STDERR and STDOUT fields when defining the WebSphere Application Server.  
# This is usually the INTERNET directory under B7333.  
#

stderr=/QIBM/UserData/WebAS5/Base/default/logs/AS\_JS\_84\_DENI507A/SystemErr.log  
stdout=/QIBM/UserData/WebAS5/Base/default/logs/AS\_JS\_84\_DENI507A/SystemOut.log  
#stderr=/QIBM/UserData/WebAS5/Base/default/installedApps/DENI507A/EA\_JS\_84\_DENI507A.ear/webclient.war/logs/stderr.txt  
#stdout=/QIBM/UserData/WebAS5/Base/default/installedApps/DENI507A/EA\_JS\_84\_DENI507A.ear/webclient.war/logs/stdout.txt

### [JAS INSTANCE]

# This is for JAS redirector only  
# RoundRobin will be used as default if there is no entry in [JAS WEIGHT].  
# If redirect based on JAS instance weight, then each instance's weight must be specified,  
# otherwise, weight 0 will be assumed, which means the server will not get  
# any redirected request  
# example: JAS1=http://127.0.0.1/jde  
# Note: in this example, "JAS1" is the key, if weight will be given,  
# this key must be used in [JAS WEIGHT] section to match this JAS instance

### [JAS WEIGHT]

# The key must match the key in [JAS INSTANCE] section  
# example: JAS1=2  
# The weight is ratio based, for example:  
# JAS1=2  
# JAS2=1



## IBM eServer iSeries Performance and Tuning Tips for the JD Edwards EnterpriseOne Web Server

### Page 36

# That means JAS1 will take as twice load as JAS2

#### [REDIRECTOR]

# interval unit is milliseconds

Interval=5000

# This is the location that Redirector will redirect to, if no server is available.

# If it is empty, redirect will display a simple page tells user that no server is available

# example: NoServerAvailableURL=http://servername/path/noserver.html

NoServerAvailableURL=

#### [TRANSACTIONS]

# Configuration settings to customize transaction processing within JAS.

# ReapInterval : Interval in minutes of how often the TransactionReaper

# should check for abandoned transactions.

# Valid values: Integer from 1 to n

# Default: 5 minutes

# MaxAge : Maximum age in minutes of a transaction. Transactions

# older than this will be rolled back by the

# TransactionReaper.

# Valid values: Integer from 1 to n

# Default: 10 minutes

#

ReapInterval=5

MaxAge=10

#### [ERPINTERACTIVITY]

#interactivity level. valid values are: LOW, MEDIUM, and HIGH

InteractivityLevel=HIGH

# whether multiple browsers fetch is enabled--must be FALSE for lab testing

MultipleBrowserEnabled=FALSE

# the maximum number of open browsers for each user

MaxOpenBrowsers=5

# the number of records fetched for each database fetch before getting a system warning

DBFetchLimitBeforeWarning=2000

#### *jdelog.properties*

# This is a sample configuration file for root configuration

jdelog.rootLogger=ERROR,JDELOG,JASLOG

jdelog.loggerFactory=com.jdedwards.base.logging.log4j.JdeLoggerFactory

jdelog.reloadInterval=60

# These are the settings for file handlers.

jdelog.handler.JDELOG=com.jdedwards.base.logging.log4j.FileHandler

jdelog.handler.JDELOG.File=/QIBM/UserData/WebAS5/Base/default/installedApps/DENI507A/EA\_JS\_84\_DENI507A.ear/webclient.war/ogs/jderoot.log

jdelog.handler.JDELOG.Level=ERROR

jdelog.handler.JDELOG.Append=FALSE

jdelog.handler.JDELOG.MaxBackupIndex=1

jdelog.handler.JDELOG.MaxFileSize=10MB

jdelog.handler.JDELOG.format=com.jdedwards.base.logging.log4j.DefaultFormat

# These are the settings for file handlers.

jdelog.handler.JASLOG=com.jdedwards.base.logging.log4j.FileHandler

jdelog.handler.JASLOG.File=/QIBM/UserData/WebAS5/Base/default/installedApps/DENI507A/EA\_JS\_84\_DENI507A.ear/webclient.war/ogs/jas.log

jdelog.handler.JASLOG.Level=ERROR

jdelog.handler.JASLOG.Append=TRUE

jdelog.handler.JASLOG.MaxBackupIndex=1

jdelog.handler.JASLOG.MaxFileSize=10MB

jdelog.handler.JASLOG.format=com.jdedwards.base.logging.log4j.DefaultFormat

#This is added for debug logging. Please uncomment following to start debug.

#jdelog.Debug=DEBUG, jasdebug



# IBM eServer iSeries Performance and Tuning Tips for the JD Edwards EnterpriseOne Web Server

## Page 37

```
#jdelog.handler.jasdebug=com.jdedwards.base.logging.log4j.FileHandler
#jdelog.handler.jasdebug.File=/QIBM/UserData/WebAS5/Base/default/installedApps/DENI507A/EA_JS_84_DENI507A.ear/webclient.war
/logs/jasdebug.log
#jdelog.handler.jasdebug.Level=DEBUG
```

### JDBJ.INI

```
=====
#          jdbj.ini - JDBj configuration
#
# This file contains configuration information for JDBj, which provides
# general database access capabilities on behalf of J.D. Edwards Java
# solutions.
#
# IMPORTANT: All keys in this file are case sensitive.
=====
```

```
=====
#          ERP specific information
#
# These sections apply only when accessing ERP data. These sections are
# ignored for standalone database access.
=====
```

```
#-----
# Bootstrap session
#
# The user, password, role information is optional. It is used by the
# the JDBJ Unit tests and JDBExplorer only if these parameters are not
# specified on the command line. The environment is required field otherwise
# JAS cannot signon to the HTML client.
#
# NOTES: * This information in this section corresponds to the information
# specified in the [DB SYSTEM SETTINGS] section of this installation's
# JDE.INI file.
#-----
```

#### [JDBj-BOOTSTRAP SESSION]

```
user=PSFT
password=PSFT
environment=PD811
role=*ALL
```

```
#-----
# Bootstrap physical data source
#
# This defines the data source where the OCM and some other system
# tables reside. JDBj uses this at bootstrap time and later to look up
# OCM entries on demand. If it is not valid, then no ERP database
# access is possible.
#
```

# Here is a description of each field in this section:

```
#-----
# Field      Description          Applies to
#-----
# name       The data source name. This is not that All
#            important for bootstrap connections,
#            but it does show up in error messages
#            and the log.
#-----
# databaseType The database type. One of the All
#            following:
#
#            I = AS/400
```



```
#          O = Oracle
#          S = SQL Server
#          W = UDB
#          M = MSDE
# -----
# server    The server (i.e. the host), or      AS/400,
#           "localhost".                      SQL Server
#                                           MSDE
# -----
# database  The database.                    Oracle,UDB
# -----
# serverPort The server port.                SQL Server
# -----
# physicalDatabase The physical database (used as library AS/400,
#                 qualifier for AS/400).      SQL Server,
#                                           MSDE
# -----
# owner     The owner.                      Oracle,
#                                           SQL Server,
#                                           UDB,
#                                           MSDE
# -----
# lob       Supports lobs.                  Oracle,
#                                           AS/400
# -----
# unicode   Performs unicode conversion.     SQL Server
#                                           MSDE
# -----
# databaseInstance The instance name of server. (JDELocal) MSDE
# -----
# translateOn AS400 translate on (default is FASLE) AS400
# -----
#
# NOTES: * This information in this section should match the
#         corresponding information specified in the
#         [DB SYSTEM SETTINGS] section of this installation's JDE.INI
#         file.
# -----

[JDBj-BOOTSTRAP DATA SOURCE]
name=SY811
databaseType=l
server=DENI502
serverPort=
database=
physicalDatabase=SY811
owner=
lob=false
unicode=true
databaseInstance=
translateOn=

# -----
# Bootstrap spec data source (Optional)
#
# This defines the data source where the serialized spec tables reside.
# JDBj uses this at bootstrap time and after to look up serialized spec
# objects on demand. JDBj first looks for the name specified in this
# section in the Data Source Master table (F98611). If it finds a data
# source there, it uses it to read the serialized spec tables. If it
# does not find the data source in the Data Source Master table, then
# it initializes a new data source based on the rest of the fields in
# this section.
#
```

# IBM eServer iSeries Performance and Tuning Tips for the JD Edwards EnterpriseOne Web Server

## Page 39

```
# If this section is not specified, then JDBj uses the OCM to find the
# serialized spec tables. If it is not valid, then no ERP database
# access is possible.
#
# See the description for "Bootstrap physical data source" above for
# information about which fields must be filled in for a given database
# platform.
#-----

;[JDBj-SPEC DATA SOURCE]
;name=
;databaseType=
;server=
;serverPort=
;database=
;physicalDatabase=
;owner=
;lob=false
;unicode=true
;user=
;password=

#=====
#           END - ERP specific information
#
# The rest of the sections apply when accessing ERP and standalone data.
#=====

#-----
# JDBC drivers
#
# This defines JDBC drivers and JDBC specific settings.
#-----

[JDBj-JDBC DRIVERS]
ORACLE=oracle.jdbc.driver.OracleDriver
AS400=com.ibm.as400.access.AS400JDBCdriver
SQLSERVER=com.microsoft.jdbc.sqlserver.SQLServerDriver
UDB=COM.ibm.db2.jdbc.app.DB2Driver

[JDBj-ORACLE]
tns=

[JDBj-LOGS]
jdbcTrace=false

[JDBj-SERVER]
dbsConversionTolerant=true

#-----
# Connection pool settings
#-----

[JDBj-CONNECTION POOL]
minConnection=5
maxConnection=500
initialConnection=45
poolGrowth=10
    ; The minimum number of connections to a data source. When closing old connections,
    ; the value set by this parameter will remain in the pool regardless of use.
    ;
    ; Valid values: 0 or greater
    ; Default value: 5

#minConnection=8
```

# IBM eServer iSeries Performance and Tuning Tips for the JD Edwards EnterpriseOne Web Server

## Page 40

```
; The maximum number of connections to a data source for all users. Additional
; database connection requests beyond this value will be queued for the next available
; connection.
;
; Valid values: 1 or greater (must be equal to or greater than "minConnection")
; Default value: 50

#maxConnection=750

; The number of connections to initialize upon pool creation.
;
; Valid values: a value equal to or greater than "minConnection" AND equal to or
; less than "maxConnection"
; Default value: 5

#initialConnection=75

; The increment of connections the system will create when a request for a connection
; cannot be satisfied with the current allocation as long as the maxConnection value
; has not been reached.
;
; Valid values: The rule of thumb is (poolGrowth >= maxConnection/10)
; Default value: 5

#poolGrowth=10

; Amount of time in milliseconds that a connection remains idle before it is flagged
; to be cleaned up. Keeping this value high may or may not add to performance. Reducing
; the value may reduce memory consumption.
;
; Valid values: 0 or greater
; Default value: 1800000 (30 minutes)

connectionTimeout=1800000

; Amount of time in milliseconds that the pool cleaner is set to clean data sources
; that have a validation string. When cleaning occurs, any connections which have been
; flagged as idle will be cleaned up. The amount of time between a connection
; becoming idle and being cleaned up can be as long as the connectionTimeout value plus
; the cleanPoolInterval value.
;
; Valid values: 0 or greater
; Default value: 600000 (10 minutes)

cleanPoolInterval=600000

; The maximum number of JDBC prepared statements cached on each JDBC connection.
;
; Valid values: 0 or greater
; Default value: 50

maxSize=50

; The number of JDBC statements purged from a JDBC connection if the maxSize limit is
; reached. The statements purged will be the statements in the cache which have not
; been used for the longest amount of time.
;
; Valid values: The rule of thumb is (cachePurgeSize >= maxSize/10 + 2)
; Default value: 5

cachePurgeSize=5

#-----
# JDBj Runtime properties
#
# These are all optional. They correspond to system runtime properties
```

# IBM eServer iSeries Performance and Tuning Tips for the JD Edwards EnterpriseOne Web Server

## Page 41

# applicable to JDBj logical connections.

#

# NOTES: \* In each case, the values listed are the defaults. The  
# template default entries are commented out. Make sure  
# and remove the pound sign (#) when changing these values.

#-----

### [JDBj-RUNTIME PROPERTIES]

; AS400 Database ASP property. Use this setting if you are running your iSeries  
; server with multiple user databases or IASP (Independent Auxiliary Storage Pool)  
; and have your EnterpriseOne tables located on the IASP. The IASP name can be found  
; by using the command WRKRDBDIRE. This entry is not needed if you only have \*LOCAL  
; configured on your system in the WRKRDBDIRE Remote Location name.  
;  
; Valid values: Any valid string

# AS400DatabaseASP=null

; AS400 package name property. Overrides the default sql package  
; name on the AS400 with the specified name.  
;  
; Valid values: Any valid string

# as400Package=JDBj

; AS400 package library name property. Overrides the default sql package  
; library name on the AS400 with the specified name.  
;  
; Valid values: Any valid string

# as400PackageLibrary=JDBj-Library

as400PackageLibrary=QRECOVERY

; AS400 extended dynamic property. Overrides the default extended dynamic  
; setting with the specified setting. If specified to a false, AS400 database  
; will not use packages. This will have a negative impact on performance.  
;  
; Valid values: true, false

# as400ExtendedDynamic=true

; Data cache entry size threshold property. Sets the maximum  
; size (in rows) for a single entry in the data cache. If a  
; potential entry exceeds this threshold, then it will not be  
; cached. -1 indicates that all eligible entries should be  
; cached.  
;  
; Valid values: -1 or greater

# dataCacheEntrySizeThreshold=500

; Data cache entry threshold property. Sets the maximum size (in  
; entries) for the data cache. If the data cache exceeds this  
; size, then the least recently used entries are removed to  
; make room for the new entry. -1 indicates that there is no  
; maximum.  
;  
; Valid values: -1 or greater

# dataCacheEntryThreshold=100

; Data verification property. Enable or disable the data verification  
; between actual data and EnterpriseOne specification for insert and  
; update operations.  
; The verification is performed on the following types:  
; - EVDT\_CHAR



```
; - EVDT_STRING
; - EVDT_NID
; - EVDT_TEXT
; - EVDT_VARSTRING
; - EVDT_VARCHAR
; - EVDT_MATH_NUMERIC
;
; This property will have a negative impact on performance.
;
; Valid values: true or false

# dataVerification=false

; Oracle NCHAR Trimming property. Indicates if the Oracle NCHAR
; Trimming logic is turned on or off. True means the returned NCHAR
; data is intact. False means always right trim the returned
; NCHAR data.
;
; Valid values: true or false

# disableOracleNcharTrimming=false

; Force single data source property. Indicates whether JDBj
; should force operations involving multiple database objects to
; run against a single data source. If this is true, JDBj will
; only resolve the physical data source for the first database
; object involved in a given operation. Otherwise, JDBj will
; resolve physical data sources for all database objects and
; throw an exception if a single operation spans multiple data
; sources.
;
; Valid values: true, false

# forceSingleDataSource=false

; Result set timeout property. Sets the the number of
; milliseconds for the result set timeout. A result set will
; timeout if it has not had any operations for a period of time.
; -1 means that result sets will not timeout.
;
; Valid values: -1 or greater

# resultSetTimeout=60000

; Retry interval property. Sets the interval (in milliseconds)
; to wait before retrying idempotent database operations. -1
; means not to retry, 0 means retry immediately.
;
; Valid values: -1 or greater

# retryInterval=0

; Retry maximum property. Sets the number of times to retry
; idempotent database operations. -1 means to retry
; indefinitely.
;
; Valid values: -1 or greater

# retryMaximum=5

; OCM cache purge property. Sets the number of
; milliseconds after which inactive OCM cache entries are
; purged. 0 means to never purge OCM cache entries.
;
; Valid values: 0 or greater
```

## IBM eServer iSeries Performance and Tuning Tips for the JD Edwards EnterpriseOne Web Server

### Page 43

# ocmCachePurge=3600000

; Personal Spec Versions property. Indicates if user's personal  
; version can be read. This is only enforced during a read and not  
; during a spec generation. If it set to a true, the personal version  
; will be returned if it exists, otherwise the public version will be  
; returned. If the property is set to a false, then the personal  
; version will not even be read and we will default to the public  
; version  
;  
; Valid values: true , false

# personalSpecVersions=false

; Service cache purge property. Sets the number of  
; milliseconds after which inactive service cache entries are  
; purged. 0 means to never purge service cache entries.  
;  
; Valid values: 0 or greater

# securityCachePurge=3600000

; Service cache purge property. Sets the number of  
; milliseconds after which inactive service cache entries are  
; purged. 0 means to never purge service cache entries.  
;  
; Valid values: 0 or greater

# serviceCachePurge=3600000

; Spec cache purge property. Sets the number of  
; milliseconds after which inactive spec cache entries are  
; purged. 0 means to never purge spec cache entries.  
;  
; Valid values: 0 or greater

# specCachePurge=3600000

; Spec consistency check property. Sets the level of spec  
; consistency checking. Set this property to a higher level to  
; improve exception messages relating to spec inconsistency  
; problems. Set this to a lower level when it is not needed  
; since it may degrade performance.  
;  
; Valid values: full, minimal, none

# specConsistencyCheck=minimal

; Spec generate on demand property. Indicates whether JDBJ  
; should generate specs on demand if it can not find them  
; in the persistent cache.  
;  
; Valid values: true, false

# specGenerateOnDemand=false

; Transaction isolation property. Sets the isolation level to be  
; used for transactions.  
;  
; Valid values: default, none, read uncommitted, read committed,  
; repeatable read, serializable

# transactionIsolation=default

; Transaction timeout property. Sets the number of milliseconds



```
; for the transaction timeout. A transaction will timeout if it
; has not had any operations for a period of time. If this
; timeout is set to a -1 then the transaction timeout feature
; will be disabled. (This currently only happens when JDBJ is
; run as part of a JAS transaction.)
;
; Valid values: -1 or greater

# transactionTimeout=120000

; Trigger auto fetch property. Indicates whether JDBJ should
; automatically fetch rows to be changed in order to pass them
; to ERP triggers. This is needed for certain triggers to
; work properly.
;
; Valid values: none, single, all

# triggerAutoFetch=none

; Updatable result set timeout property. Sets the the number of
; milliseconds for the updatable result set timeout. An
; updatable result set will timeout if it has not had any
; operations for a period of time. -1 indicates that updatable
; result sets should have the same value as RESULT_SET_TIMEOUT.
;
; Valid values: -1 or greater

# updatableResultSetTimeout=-1

; Usage execution threshold property. Sets the expected maximum
; time (in milliseconds) for a single database operation to
; execute. If a single database operation takes longer than this
; threshold, then the statement and time will be logged as part
; of usage tracking.
;
; Valid values: 0 or greater

# usageExecutionThreshold=20000

; Usage fetch size threshold property. Sets the expected maximum
; fetch size (in rows) for a single fetch to return. If a single
; fetch returns more than than this threshold, then the actual
; fetch size will be logged as part of usage tracking. -1
; indicates that any fetch size is valid.
;
; Valid values: -1 or greater

# usageFetchSizeThreshold=500

; Usage result set open threshold property. Sets the expected
; maximum time (in milliseconds) for a result set to remain
; open. If a result set is left open longer than this threshold,
; then the result set and time will be logged as part of usage
; tracking. This does not automatically close the result set. A
; usage result set open threshold value of -1 indicates that no
; such usage tracking will occur.
;
; Valid values: -1 or greater

# usageResultSetOpenThreshold=60000

; Usage tracking property. Enables or disables usage tracking.
; This provides additional tracking and logging to use during
; testing. Usage tracking does incur a performance penalty so it
; should be disabled for production code.
;
```

# IBM eServer iSeries Performance and Tuning Tips for the JD Edwards EnterpriseOne Web Server

## Page 45

```
; Valid values: true, false

# usageTracking=false

; Usage tracking output property. Specifies the destination of
; usage tracking output.
;
; Valid values: log, exception

# usageTrackingOutput=log

; Usage transaction active threshold property. Sets the expected
; maximum time (in milliseconds) for a transaction to remain
; active. If a transaction is left active longer than this
; threshold, then the transaction and time will be logged as
; part of usage tracking. This does not automatically rollback
; the transaction. A usage transaction active threshold value of
; -1 indicates that no such usage tracking will occur.
;
; Valid values: -1 or greater

# usageTransactionActiveThreshold=120000

#=====
#                               END OF FILE
#=====

jde.ini
; OneWorld initialization file INI(JDE)
; AS/400 specific version - E811

[JDE_CG]
CLASSPATH=/E811SYS/classes/xalan.jar:/E811SYS/classes/xerces.jar:/E811SYS/classes/kernel.jar:/E811SYS/classes/XTS.jar:/E811S
YS/classes/log4j.jar:/E811SYS/classes

[DEBUG]
Output=NONE
DebugFile=/PSE811/JDEDEBUG
JobFile=/PSE811/JDE.LOG
JDETSFile=/PSE811/JDETS.LOG
LogErrors=1
ClientLog=0
KeepLogs=1
RunBatchDelay=0
TAMTraceLevel=0

[JDEMAIL]
Rule1=90|OPT|MAILSERVER=mail.jdedwards.com
Rule2=100|DEFAULT|OWMON=OWMON@jdedwards.com
Rule3=110|DEFAULT|JDE_SYSTEM=JDE_System@jdedwards.com
Rule4=120|DEFAULT|WORKFLOW_SYSTEM=Workflow@jdedwards.com
Rule5=130|OPT|MERGELOCAL=1
Rule6=140|OPT|UPDATELOCAL=0
ClientType=Windows

[SVR]
EnvironmentName=PD811
LibraryListName=PD811
SpecPath=specfile
ApplicationPathAddendum=

[INSTALL]
;POSTSCRIPT_ONLY=1
DefaultSystem=E811SYS
```



## IBM eServer iSeries Performance and Tuning Tips for the JD Edwards EnterpriseOne Web Server

### Page 46

B9=  
Double\_Byte=0  
LocalCodeSet=US\_EBCDIC

[JDEIPC]  
maxNumberOfResources=1000  
startIPCKeyValue=2101  
avgResourceNameLength=15  
maxMsgqEntries=1024  
maxMsgqBytes=65536  
ipcTrace=0

[JDENET]  
serviceNameListen=6013  
serviceNameConnect=6013  
maxNetProcesses=3  
maxNetConnections=800  
netShutdownInterval=15  
maxKernelProcesses=612  
maxKernelRanges=24  
netTrace=0  
enablePredefinedPorts=0  
checkKrnIHealth=10000

[JDENET\_KERNEL\_DEF1]  
krnlName=JDENET RESERVED KERNEL  
dispatchDLLName=JDENET  
dispatchDLLFunction=JDENET\_DispatchMessage  
maxNumberOfProcesses=1  
numberOfAutoStartProcesses=0

[JDENET\_KERNEL\_DEF2]  
krnlName=UBE KERNEL  
dispatchDLLName=JDEKRNL  
dispatchDLLFunction=JDEK\_DispatchUBEMessage  
maxNumberOfProcesses=1  
numberOfAutoStartProcesses=0

[JDENET\_KERNEL\_DEF3]  
krnlName=REPLICATION KERNEL  
dispatchDLLName=JDEKRNL  
dispatchDLLFunction=DispatchRepMessage  
maxNumberOfProcesses=1  
numberOfAutoStartProcesses=0

[JDENET\_KERNEL\_DEF4]  
krnlName=SECURITY KERNEL  
dispatchDLLName=JDEKRNL  
dispatchDLLFunction=JDEK\_DispatchSecurity  
maxNumberOfProcesses=2  
numberOfAutoStartProcesses=2

[JDENET\_KERNEL\_DEF5]  
krnlName=LOCK MANAGER KERNEL  
dispatchDLLName=JDEKRNL  
dispatchDLLFunction=TM\_DispatchTransactionManager  
maxNumberOfProcesses=1  
numberOfAutoStartProcesses=0

[JDENET\_KERNEL\_DEF6]  
krnlName=CALL OBJECT KERNEL  
dispatchDLLName=XMLCALLOBJ  
dispatchDLLFunction=XMLCallObjectDispatch  
maxNumberOfProcesses=34  
numberOfAutoStartProcesses=34

```
[JDENET_KERNEL_DEF7]
krnlName=JDBNET KERNEL
dispatchDLLName=JDEKRNL
dispatchDLLFunction=JDEK_DispatchJDBNETMessage
maxNumberOfProcesses=1
numberOfAutoStartProcesses=0
```

```
[JDENET_KERNEL_DEF9]
krnlName=SAW KERNEL
dispatchDLLName=JDESAW
dispatchDLLFunction=JDEK_DispatchSAWMessage
maxNumberOfProcesses=1
numberOfAutoStartProcesses=0
```

```
[JDENET_KERNEL_DEF10]
krnlName=SCHEDULER KERNEL
dispatchDLLName=JDEKRNL
dispatchDLLFunction=JDEK_DispatchScheduler
maxNumberOfProcesses=1
numberOfAutoStartProcesses=0
```

```
[JDENET_KERNEL_DEF11]
krnlName=PACKAGE BUILD KERNEL
dispatchDLLName=JDEKRNL
dispatchDLLFunction=JDEK_DispatchPkgBuildMessage
maxNumberOfProcesses=1
numberOfAutoStartProcesses=0
```

```
[JDENET_KERNEL_DEF12]
krnlName=UBE SUBSYSTEM KERNEL
dispatchDLLName=JDEKRNL
dispatchDLLFunction=JDEK_DispatchUBESBSMessage
maxNumberOfProcesses=1
numberOfAutoStartProcesses=0
```

```
[JDENET_KERNEL_DEF13]
krnlName=WORK FLOW KERNEL
dispatchDLLName=WORKFLOW
dispatchDLLFunction=JDEK_DispatchWFServerProcess
maxNumberOfProcesses=0
numberOfAutoStartProcesses=0
```

```
[JDENET_KERNEL_DEF14]
krnlName=QUEUE KERNEL
dispatchDLLName=JDEKRNL
dispatchDLLFunction=DispatchQueueMessage
maxNumberOfProcesses=1
numberOfAutoStartProcesses=0
```

```
[JDENET_KERNEL_DEF15]
krnlName=XML TRANS KERNEL
dispatchDLLName=XMLTRANS
dispatchDLLFunction=XMLTransactionDispatch
maxNumberOfProcesses=1
numberOfAutoStartProcesses=0
```

```
[JDENET_KERNEL_DEF16]
krnlName=XML LIST KERNEL
dispatchDLLName=XMLLIST
dispatchDLLFunction=XMLListDispatch
maxNumberOfProcesses=1
numberOfAutoStartProcesses=0
```

```
[JDENET_KERNEL_DEF19]
krnlName=EVN KERNEL
dispatchDLLName=JDEIE
```

## IBM eServer iSeries Performance and Tuning Tips for the JD Edwards EnterpriseOne Web Server

### Page 48

```
dispatchDLLFunction=JDEK_DispatchITMessage  
maxNumberOfProcesses=0  
numberOfAutoStartProcesses=0
```

```
[JDENET_KERNEL_DEF20]  
krnlName=IEO KERNEL  
dispatchDLLName=JDEIEO  
dispatchDLLFunction=JDEK_DispatchIEMessage  
maxNumberOfProcesses=0  
numberOfAutoStartProcesses=0
```

```
[JDENET_KERNEL_DEF22]  
krnlName=XML DISPATCH KERNEL  
dispatchDLLName=XMLDSPATCH  
dispatchDLLFunction=XMLDispatch  
maxNumberOfProcesses=1  
numberOfAutoStartProcesses=0
```

```
[JDENET_KERNEL_DEF23]  
krnlName=XTS KERNEL  
dispatchDLLName=XTSKRNL  
dispatchDLLFunction=JDEK_DispatchXTSMessage  
maxNumberOfProcesses=1  
numberOfAutoStartProcesses=0
```

```
[JDENET_KERNEL_DEF24]  
krnlName=XML SERVICE KERNEL  
dispatchDLLName=XMLSERVICE  
dispatchDLLFunction=XMLServiceDispatch  
maxNumberOfProcesses=1  
numberOfAutoStartProcesses=0
```

```
[DB SYSTEM SETTINGS]  
Version=43  
Default User=PSFT  
Default Pwd=PSFT  
Default Role=*ALL  
Default Env=PD811  
Default PathCode=PD811  
Base Datasource=DENI502 - 811 Server Map  
Object Owner=  
Server=DENI502  
Database=DENI502 - 811 Server Map  
Load Library=  
Decimal Shift=Y  
Julian Dates=Y  
Use Owner=N  
Secured=Y  
Type=l  
Library=  
Library List=  
DatabaseName2=SVM811  
DatabaseInstance=  
ServerPort=  
JDBNETUse=N  
UnicodeFlag=Y  
LOBFlag=Y  
DatabaseProgramMax=-1  
DatabaseProgramInitial=10  
DatabaseProgramThreshold=3  
DatabaseProgramAdditional=10  
DatabaseProgramCheckIntervalSeconds=20  
#DatabaseProgramCheckIntervalSeconds=10 double this to avoid manual commit failures  
Default Journal=OW_JRNL  
Default Journal LIBRARY=OWJRNL  
Default Journal Receiver=OW_JRNL000
```



# IBM eServer iSeries Performance and Tuning Tips for the JD Edwards EnterpriseOne Web Server

## Page 49

Default Journal Receiver LIBRARY=OWJRNL  
SQL Package Library=2

[NETWORK QUEUE SETTINGS]  
DefaultPrinterOUTQ=QGPL/ONEWORLD\_A  
QKActive=1  
QKOnIdle=300

[BSFN BUILD]  
BuildArea=/OneWorld/Packages  
OptimizationFlags=(40)  
DebugFlags=\*ALL  
InliningFlags=Y  
DefineFlags=JDENV\_AS400MUTEX PRODUCTION\_VERSION JDBDB2400 AS400V3R6  
CompilerFlags=\*EXPMAC \*NOSHOWINC  
CompileOutput=\*PRINT  
OSReleaseLevel=\*CURRENT  
LinkFlags=\*DUPPROC \*DUPVAR \*UNRSLVREF  
LinkLibraries=JDEUNICODE JDELIB JDEKRNL OWVER V\_VERIFY XERCESWRAP  
SimultaneousBuilds=0

[AS400]  
CRTCMOD=CRTCMOD MODULE(%s/%s) SRCFILE(%s/%s) SRCMBR(%s) OUTPUT(\*PRINT) DBGVIEW(\*NONE) OPTIMIZE(40)  
CRTCMOD2=DEFINE(JDENV\_AS400MUTEX PRODUCTION\_VERSION NO\_SIGNALS) TGTRLS(\*CURRENT)  
CRTSRVPGM=CRTSRVPGM SRVPGM(%s/%s) MODULE(%s/\*ALL) BNDSRVPGM(JDEUNICODE JDELIB JDEKRNL OWVER  
V\_VERIFY XERCESWRAP) EXPORT(\*ALL) OPTION(\*DUPPROC \*DUPVAR \*UNRSLVREF) ALWLIBUPD(\*YES)  
TGTRLS(\*CURRENT)  
CRTDBPGM1=CRTPGM PGM(%3s/%s) MODULE(DBDRVAG DBDRV DBDRV\_AC DBDRV\_CC DBDRV\_CN  
CRTDBPGM2=DBDRV\_CH DBDRV\_CP DBDRV\_DT DBDRV\_RQ DBDRVSQL DBMONCTL DBDRVCLI  
CRTDBPGM3=DBSQL DBSQL\_A DBSQL\_D DBSQL\_I DBSQL\_M DBSQL\_S DBSQL\_U DBSQL\_X)  
CRTDBPGM4=BNDSRVPGM(JDEUNICODE JDEKRNL JDELIB JDEIPC QBFCPRCED) ACTGRP(%s) OPTION(\*DUPPROC  
CRTDBPGM5=\*DUPVAR) ALWLIBUPD(\*YES) AUT(\*ALL) TGTRLS(\*CURRENT)  
PrintUBEJoblog=FALSE  
PrintUBEJoblogOnError=FALSE  
useAS400Security=TRUE

[SERVER ENVIRONMENT MAP]  
PSFTPLAN=PS811

[UBE]  
UBEDebugLevel=0

[SECURITY]  
DataSource=System - 811  
User=PSFT  
Password=PSFT  
Default Role=\*ALL  
DefaultEnvironment=PD811  
SecurityServer=DENI502  
ServerPswdFile=TRUE  
History=0

[LOCK MANAGER]  
Server=  
AvailableService=NONE  
RequestedService=None

[TCENGINE]  
TraceLevel=1  
StopAfterRow=0  
ForceRowByRow=0

[INTEROPERABILITY]  
FilteredEvents=\*NONE  
RegisteredEvents=



# IBM eServer iSeries Performance and Tuning Tips for the JD Edwards EnterpriseOne Web Server

## Page 50

```
[SAMPLE_EVENT]
DS1=DXXXXXXXXX
DS2=DYYYYYYYYY
DS3=DZZZZZZZZ
```

```
[JDEITDRV]
DrvCount=4
Drv1=RT:RTDRV
Drv2=WF:WFDRV
Drv3=Z:ZDRV
Drv4=JDENET:JDETRDRV
```

```
[LRENGINE]
System=/E811SYS/output
Repository_Size=20
Disk_Monitor=YES
```

```
[WORLD ENVIRONMENT MAP]
```

```
[WORKFLOW]
; Workflow Runtime Interop, inbound (server only)
WRIUser=xxx
WRIPassword=xxx
WRIEnvironment=xxx
WRIRole=*ALL
; Workflow Runtime Interop, outbound
PrimaryEvnServer=xxx
SecondaryEvnServer=xxx
```

```
[JAS PREFERENCE]
JasServer=
Port=80
Servlet=/jde/servlet/html.login
```

```
:[TRANSPORTS]
;TransportName1=xxxTransportExeNamexxx
;TransportParams1=xxxTransportParametersxxx
```

```
[XAPI]
XMLDirectory=PSE811
```

```
[XMLLookupInfo]
XMLRequestType1=list
XMLKernelMessageRange1=5257
XMLKernelHostName1=local
XMLKernelPort1=0
```

```
XMLRequestType2=callmethod
XMLKernelMessageRange2=920
XMLKernelHostName2=local
XMLKernelPort2=0
```

```
XMLRequestType3=trans
XMLKernelMessageRange3=5001
XMLKernelHostName3=local
XMLKernelPort3=0
```

```
XMLRequestType4=JDEMSGWFINTEROP
XMLKernelMessageRange4=4003
XMLKernelHostName4=local
XMLKernelPort4=0
XMLKernelReply4=0
```

```
XMLRequestType5=xapicalmethod
XMLKernelMessageRange5=14251
XMLKernelHostName5=local
```

## IBM eServer iSeries Performance and Tuning Tips for the JD Edwards EnterpriseOne Web Server Page 51

XMLKernelPort5=0

XMLRequestType6=realTimeEvent  
XMLKernelMessageRange6=14251  
XMLKernelHostName6=local  
XMLKernelPort6=0  
XMLKernelReply6=0

[XTSRepository]  
XTSRepositoryPath=/E811SYS/XtsRepository  
XSL-JDE-BIRDS-REQUEST=birds.xsl  
XSL-JDE-V2\_2\_NATIVE-REQUEST=v2toNative.xsl  
XSL-JDE-V1\_NATIVE-REQUEST=ml.xsl  
XSL-JDE-V1\_NATIVE-RESPONSE=lm.xsl

[XTS]  
XTSTemplateSelector1=com.jdedwards.xts.xtsm.XTSMJDETemplateSelector  
XTSTraceLevel=2  
XTSDelay=0

[MTR VALIDATION]  
EnterpriseDiskInMB=3000

**Appendix C – Special Note if Upgrading from Release 8.9 or 8.10**

The first version of this document was originally developed for Release 8.9. This current version has updated the tips based on Release 8.11 testing. For those customers who have already implemented the original Release 8.9 tuning tips, the only significant modification needed for tuning Release 8.11 is to adjust the initial heap size downward. Please review 1) Adjust the heap size for the JVM section, under the WebSphere Application Server Tuning heading. Release 8.11 works best with a smaller initial heap size, 1.5 MB memory per active user, whereas the recommendation for Release 8.9 had been 3.3 MB of memory per active user. All of the other tuning tips have remained the same for Release 8.11.

**Contact information**

Gerrie Fisk  
Senior iSeries System Specialist  
IBM / Oracle EnterpriseOne International Competency Center  
[mcfisk@us.ibm.com](mailto:mcfisk@us.ibm.com)  
303-334-2073

Clark Scholten  
Senior Software Engineer  
iSeries ERP Development – Oracle Support  
[scholten@us.ibm.com](mailto:scholten@us.ibm.com)  
303-334-8720

Boyd Fenton  
Competency Center Manager  
IBM / Oracle International Competency Center  
[bafento@us.ibm.com](mailto:bafento@us.ibm.com)  
303-334-2524

**For more information**

For an overview of IBM iSeries performance management tools read the IBM RedPaper “IBM Performance Management Tools” published September, 19, 2005. More details are available at <http://www.redbooks.ibm.com/redpapers/abstracts/redp4026.html>

For additional information on integrated, collaborative enterprise solutions from IBM and JD Edwards EnterpriseOne, call 1 888 426-5505 or visit [ibm.com/solutions/businesssolutions/oracle](http://ibm.com/solutions/businesssolutions/oracle)



©Copyright IBM Corporation 2004

IBM Corporation  
Integrated Marketing Communications  
Server Group  
Route 100  
Somers, NY 10589  
U.S.A.  
ibm.com

Visit [www.ibm.com/pc/safecomputing](http://www.ibm.com/pc/safecomputing) periodically for the latest information on safe and effective computing. Warranty Information: For a copy of applicable product warranties, write to: Warranty Information, P.O. Box 12195, RTP, NC 27709, Attn: Dept. JDJA/B203. IBM makes no representation or warranty regarding third-party products or services including those designated as ServerProven or ClusterProven.

This publication was developed for products and services offered in the United States. IBM may not offer the products, services or features discussed in this document in other countries. Information is subject to change without notice. Consult your local IBM representative for information on offerings available in your area.

The examples given in this paper are hypothetical examples of how a customer can use the products described herein and examples of potential cost or efficiency savings are not based on any actual case study. There is no guarantee of comparable results. Many factors determine the sizing requirements and performance of a systems architecture. IBM assumes no liability for the methodology used for determining the configurations recommended in this document nor for the results it provides. Any performance data contained in this presentation was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements quoted in this presentation may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Some measurements quoted in this presentation may have been estimated through extrapolation. Actual results may vary. Users of this presentation should verify the applicable data for their specific environment.

Information in this presentation concerning non-IBM products was obtained from the suppliers of these products, published announcement material or other publicly available sources. IBM has not tested these products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Intel and Xeon are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

IBM, the IBM logo, the e-business logo, eServer, iSeries, OS/400, i5/OS, WebSphere, and DB2 are trademarks or registered trademarks of IBM Corporation in the United States, other countries, or both.

Oracle, JD Edwards, PeopleSoft, are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Other company, product and service names may be trademarks or service marks of others.