Oracle® Internet Procurement

Installation Guide

Release 11*i*

May 25, 2000

Part No. A85343-01

Oracle® Internet Procurement 11*i* Installation Guide.

Part No. A85343-01

Primary Author: Brian Gause

Contributing Authors: Padmini Ranganathan, Anand Lakhotia, Michel Bazinet, Vineet Gupta, Sanjay Bhasin, Andrew Yeung, Eddy So, Vesa Peltola, Fred Wong, Chacko Pulivelil

Contributors: Shilpa Kotwal

The programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be licensee's responsibility to take all appropriate fail-safe, back up, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle disclaims liability for any damages caused by such use of the Programs.

Send Us Your Comments

Oracle Internet Procurement Installation Gude, Release 11*i*

Part No. A85343-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the chapter, section, and page number (if available). You can send comments to us in the following ways:

- E-mail - appsdoc@us.oracle.com
- Postal service:

  Oracle Corporation
  Oracle Internet Procurement 11*i* Documentation
  500 Oracle Parkway, 3op6
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, and telephone number below.

If you have problems with the software, please contact your local Oracle Support Services.

Contents

1
Installing Oracle Internet Procurement 11*i*

A
Sample Apache Configuration Files

1

Installing Oracle Internet Procurement 11*i*

This chapter describes the various components of Oracle Internet Procurement 11*i* and details the steps for installing it and the underlying technology stack. This chapter covers the following topics:

This chapter should be read before beginning the installation procedure. This chapter covers only those tasks specific to Oracle Internet Procurement 11*i*.

> **Note:** Before starting the installation procedure, review documentation updates, if any, available on the MetaLink (http://www.oracle.com/support/elec_sup/index.html) under Oracle Purchasing or Internet Procurement.

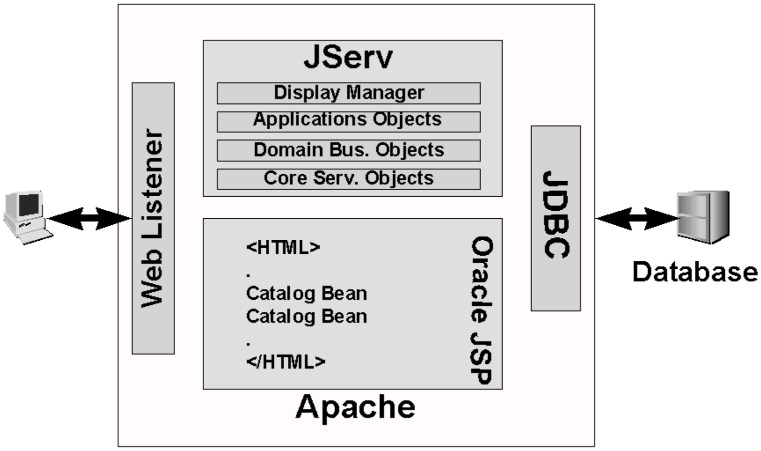
Internet Procurement Architecture

Oracle Internet Procurement 11*i* is designed to streamline the requisition process to ensure that employees can order necessary items efficiently using a simple and widely available web browser interface. Numerous tedious tasks are now automated to deliver needed items in a timely manner. The design of Internet Procurement 11*i* is built around this central requirement.

Oracle Internet Procurement 11*i* uses a 3-tier architecture comprising the database tier, an application tier, and a desk-top tier. Users connect to Oracle Internet Procurement 11*i* through any java enabled browser using a URL. The web server tier provides business logic and generates dynamic webpages.

The presentation layer of the web server, which is part of the application tier, and the business logic for Oracle Internet Procurement 11*i* are implemented as Java Servlets and JavaServer Pages (JSP) running on Apache JServ. Apache JServ, a module of Apache Web Server, implements the Java Servlet API for running server side Java.

Figure 1–1 Oracle Internet Procurement 11i Apache Server



Oracle Internet Procurement 11*i* shares the common Oracle Self-Service Web Applications signon to provide a common gateway to all Self-Service Applications. Upon first accessing Oracle Internet Procurement 11*i*, you will see the Oracle Self-Service Web Applications home page. Enter your user name and password for authentication to begin your session. You will then redirected to the Apache Web Server to begin purchasing activity.

System Requirements

Oracle Applications Release 11*i*

Oracle Internet Procurement 11*i* requires Oracle Applications Release 11*i*, with Maintenance Pack 11.5.1 or greater. For more information on installation procedures, refer to Installing Oracle Applications, Release 11*i*.

Supported Platforms

The following table lists the platforms supported by Oracle Internet Procurement 11*i* and the corresponding directory names:

Table 1–1 Platform Directories

Platform	Subdirectory
IBM RS 6000 (AIX)	aix
Compaq Tru64 UNIX	dig_unix
HP 9000 Series HP-UX	hp_ux11
Windows NT (Intel)	intel_nt
Sun Solaris x86 (Intel)	solx86
Sun Solaris (SPARC)	solaris
Linux	linux

For directory layout of the Oracle Internet Procurement 11*i* CD-ROM, please refer to the *readme.txt* file in the root directory of the CD-ROM.

Installation Procedures

Before You Begin

- Verify that Oracle Applications Release 11.5.1 or higher is fully installed and operational. For more information on installation procedures, see Oracle Applications Installation Guide, Release 11*i*.
- Determine if you are upgrading from Self-Service Purchasing 4 to Oracle Internet Procurement 11*i*. If yes, the upgrade procedures are explained in the

following sections. If you are not upgrading, **do not** apply the upgrade patch (p1305768.zip).

For information on how to use AutoPatch, see Maintaining Oracle Applications, Release 11*i*. The scripts for upgrading from Self-Service Purchasing 4 to Oracle Internet Procurement 11*i* are discussed in detail in the Oracle Internet Procurement 11*i* Implementation Manual.

Required Patches

The following patches must be applied to your Oracle Applications Release 11*i* environment. If the patch for your platform is not on the CD, contact Oracle Support or download the patch using Metalink.

- p1310958.zip
- p1313614.zip
- p1305768.zip (use only if you are upgrading from Self-Service Purchasing 4 to Oracle Internet Procurement 11*i*)
- p1291293.zip
- p1291372.zip

These patches are located in the 115 directory under the corresponding platform subdirectory as stated in the preceding table listing supported platforms.

◗◗ To apply the patches:

1. If you are using Vision Demo Release 11*i* Rapid Install, connect to SQL*Plus using the apps account and perform the following manual step:

```
SQL> delete from ICX_POR_DESCRIPTORS_TL;
SQL> commit;
```

2. Copy patches p1310958.zip, p1313614.zip, p1305768.zip, p1291293.zip, and p1291372.zip, to the administration server, concurrent processing server(s), forms server(s), and web server(s).
3. Unzip the patch files.

Note: For detailed steps on applying a patch, see its README file. Remember to check the log files for errors.

---

4. Use AutoPatch to apply patch drivers from the 1310958 directory. You must apply patch drivers in the following order on the specified server(s).

   - c1310958.drv (all servers).

5. Use AutoPatch to apply patch drivers from the 1313614 directory. You must apply patch drivers in the following order on the specified server(s).

   - c1313614.drv (all servers).
   - d1313614.drv (administration server only).
   - g1313614.drv (forms server).

Note:

- Apply patch 1305768 only if you are upgrading from Self-Service Purchasing Version 4 to Oracle Internet Procurement 11*i*. If you apply this patch to a new installation of Oracle Internet Procurement 11*i* (not an upgrade from Self-Service Purchasing 4) the patch will fail.
- Before applying this patch to an Oracle 8*i* database, the RPC portion of the Net8 listener needs to be setup properly. Otherwise, you will not be able to create InterMedia indexes. For more information, please refer to the Oracle8i Components Installation / Setup Guide, InterMedia Installation section.
- If you are not upgrading to Oracle Internet Procurement 11*i*, skip step 6.

---

6. Use AutoPatch to apply patch drivers from the 1305768 directory. You must apply patch drivers in the following order on the specified server(s).

   - c1305768.drv (all servers)

Preupgrade Step

Oracle Internet Procurement 11*i* requires that all suppliers defined in the catalog are also defined as suppliers in the PO_VENDORS table of Oracle Applications. If suppliers in the catalog are not registered as suppliers in Oracle Applications, you must use the Suppliers window in Oracle Applications to register them before starting the upgrade process.

Supplier names are case-sensitive. The supplier name in PO_VENDORS should match the supplier name in the catalog.

To identify catalog suppliers not defined in PO_VENDORS:

- Run *icxgetsu.sql* script against the APPS schema on the database. This script is stored in the $ICX_TOP/patch/115/sql directory on UNIX systems and in %ICX_TOP%\patch\115\sql directory for Windows NT systems. To run the *icxgetsu.sql* script, start a SQL*Plus session and connect to the APPS schema. Execute the *icxgetsu.sql* script using the following command:

  ```
  SQL> START icxgetsu.sql
  ```

  Running the *icxgetsu.sql* script will list all the suppliers defined in the catalog, but not defined in the PO_VENDORS table. You must now log into Oracle Applications and create them as suppliers from the Supplier Window.

Apply the following remaining patch driver:

- d1305768.drv (administration server only)

7. Use AutoPatch to apply patch drivers from the 1291293 directory. You must apply patch drivers in the following order on the specified server(s).
   - c1291293.drv (all servers).
   - d1291293.drv (administration server only).
8. Use AutoPatch to apply patch drivers from the 1291372 directory. You must apply patch drivers in the following order on the specified server(s).
   - c1291372.drv (all servers).
9. Run the AD Administration utility (adadmin) on the administration server.

   a. Compile the APPS schema. From the Maintain Applications Database Objects menu, select Compile APPS schema(s).

   b. Validate the APPS schema. From the Maintain Applications Database Objects menu, select Validate APPS schema(s).

   c. If you have installed Multiple Reporting Currencies (MRC), select the Maintain Multiple Reporting Currency schema(s) option from the Maintain Applications Database Objects menu.

10. Run the AD Administration utility (adadmin) on each server. Generate message files for all products. To generate message files, from the Maintain Applications Files menu, select Generate message files.

    > **Note:** Perform this step only if you are upgrading from Self-Service Purchasing Version 4 to Internet Procurement 11*i*. If you are not upgrading to Oracle Internet Procurement 11*i*, skip this step.

11. This step updates the access URL for Update Requisition and View Requisition functions used in the PO Requisition Approval Workflow Notification.

    a. Construct the URL for Self-Service Purchasing 4 using the following syntax:

    ```
    http://<hostmachine>.<domain>:<port>/<agent>/plsql
    ```

    where port is your the OAS port and agent is the OAS PL/SQL agent name. For example:

    ```
    http://hostmachine1.yourserver.com:8000/ssp4/plsql/
    ```

    b. Construct the URL for Oracle Internet Procurement 11*i* using the following syntax:

    ```
    http://<hostmachine>.<domain>:<port>/<DAD>
    ```

    where port is your WebDB 2.5 port and DAD is the WebDB 2.5 DAD. Rapid Install will default the WebDB 2.5 DAD to the Oracle SID. For example:

    ```
    http://hostmachine2.yourserver.com:8010/ssp5/
    ```

    c. Run *pontfupg.sql* script against the APPS schema on the database.

    This script is available in the $ICX_TOP/patch/115/sql directory.

    - To run the *pontfupg.sql* script, start a SQL*Plus session.
    - Execute the *pontfupg.sql* script using the following command:

      ```
      SQL> START pontfupg.sql <Self-Service Purchasing 4 URL> <Oracle
      Internet Procurement 11i URL>
      ```

      where Self-Service Purchasing 4 URL and Oracle Internet Procurement 11*i* URL are URLs constructed in steps a and b.

Note: You may skip step ***d*** if you are not using Confirm Receipt Workflow.

d. Update the access URL for Confirm Receipt Notifications by running the *pocrnotif.sql* script against the APPS schema on the database.

This script is available in the $ICX_TOP/patch/115/sql directory.

- To run the *pocrnotif.sql* script, start a SQL*Plus session.
- Execute the *pocrnotif.sql* script using the following command:

```
SQL> START pocrnotif.sql <Self-Service Purchasing 4 URL> <Oracle
Internet Procurement 11i URL>
```

where Self-Service Purchasing 4 URL and Oracle Internet Procurement 11*i* URL are URLs constructed in steps a and b.

Apache Configuration

Note: COMMON_TOP/util/apache/1.3.9/Apache/Apache is referred to as the Apache home directory. From this point onwards, it is referred to as APACHE_TOP when an environment variable can be used. It is referred to as <location of apache_top> when the path must be hard coded.

Verifying Apache Port Configuration

Verifying Apache involves starting and testing it to be sure it is working properly.

◗◗ To verify Apache port configuration:

1. Start Apache Web Server using the appropriate command for your platform:

   `$COMMON_TOP/admin/scripts/adapcctl.sh start` (UNIX)

   `adapcctl.cmd` (Windows NT console). *adapcctl.cmd* is located in the %APACHE_TOP% directory.

2. Test Apache by executing the following command:

   ```
   http://<local_host.domain>:port_number
   ```

For example, *http://sspbox.yourserver.com:8000* where *sspbox* is the local host name, *yourserver.com* is the domain and *8000* is the Apache Listener port number.

The Oracle Applications Rapid Install Release 11*i* portal will display.

If you do not see this page, check the error log. The error log is available in the $APACHE_TOP/log directory.

3. Stop Apache using the following command:

   `$COMMON_TOP/admin/scripts/adapcctl.sh stop` (UNIX)

   `adapcctl.cmd stop` (Windows NT console)

Apache Configuration for Oracle Internet Procurement

You will be making several changes to the configuration files while configuring Apache for Internet Procurement 11*i*. We recommend that you backup the configuration files so that you can easily see the changes you have made.

◗◗ To configure Apache for Internet Procurement 11*i*:

Modify the *httpd.conf* file to configure the Mod Proxy module. This file is available in the $APACHE_TOP/conf/ directory.

1. Using a text editor, open the *httpd.conf* file.
2. Setup the Mod Proxy Module by adding the following to the Proxy Server Directives section of the *httpd.conf* file.

```
<IfModule mod_proxy.c>
ProxyPass /<DAD>/ http://<hostname.domain:port>/<DAD>/
</IfModule>
```

   where:

   - <hostname> and <domain> are the name and domain of the host machine where WebDB 2.5 resides,
   - <port> is the WebDB 2.5 listener port, and
   - <DAD> is the WebDB 2.5 DAD.

3. Save and close the file.

◗◗ To set up jserv.conf file:

You must modify the JServ configuration file (*jserv.conf*) to specify a port number for ApJServDefaultPort. The *jserv.conf* file is available in the COMMON_TOP/util/apache/1.3.9/Apache/Jserv/etc/ directory on UNIX platforms, and in the COMMON_TOP\util\apache\1.3.9\Apache\Jserv\etc directory on Windows NT.

1. Using a text editor, open *jserv.conf* file.
2. Verify that the port number specified as the value for ApJServDefaultPort is unique and free of conflict. Modify the port number if necessary.
3. Save and close the file.

◗◗ To set up jserv.properties file for Internet Procurement 11*i*:

You must configure JServ properties file (*jserv.properties*) to make it work with Internet Procurement 11*i*. *jserv.properties* can be found in COMMON_TOP/util/apache/1.3.9/Apache/Jserv/etc/ directory on UNIX platforms, and in COMMON_TOP\util\apache\1.3.9\Apache\Jserv\etc on Windows NT.

1. Using a text editor, open *jserv.properties* file.
2. Verify that the following entry exists in the Java Virtual Machine interpreter section:

   ```
   wrapper.bin=<COMMON_TOP>/util/apache/1.3.9/Apache/jdk/bin/java
   ```

3. Verify that the value specified for Port matches the ApJServDefaultPort value in *jserv.conf* file.
4. Save and close the file.

Note: Verify that the port number you have set as `ApJServDefaultPort <port>` in *jserv.conf file* and `port <port>` in *jserv.properties* file are identical. This port number here must not be used for any other purpose on the same server.

◗◗ To modify zone.properties file:

You must modify the *zone.properties* file so that it points to the correct location for Oracle Internet Procurement 11*i*. The *zone.properties* file is available in the COMMON_TOP/util/apache/1.3.9/Apache/Jserv/etc/ directory on UNIX platforms, and in COMMON_TOP\util\apache\1.3.9\Apache\Jserv\etc directory on Windows NT.

1. Using a text editor, open *the zone.properties* file.
2. Set Enable servlet class autoloading flag to false.

   ```
   autoreload.classes=false
   ```

3. Set Enable servlet resourced autoloading flag to false.

   ```
   autoreload.file=false
   ```

4. Add the following to the Startup Servlets section:

   ```
   servlets.startup=oracle.apps.icx.common.InitSystemProperties
   ```

5. Under the Servlet Init Parameters section, add the following to specify the path of *ssp_init.txt* file. This file includes the entries made in java environment.

   UNIX

   ```
   servlet.oracle.apps.icx.common.InitSystemProperties.initArgs=initFile=
   <physical path of COMMON_TOP>/util/apache/1.3.9/Apache/Jserv/etc/ssp_
   init.txt
   ```

   Windows NT

   ```
   servlet.oracle.apps.icx.common.InitSystemProperties.initArgs=initFile=
   <physical path of COMMON_TOP>\util\apache\1.3.9\Apache\Jserv\etc\ssp_
   init.txt
   ```

6. Under the Servlet Init Parameters section, add the following (Windows NT only):

   ```
   servlet.oracle.jsp.JspServlet.initArgs=page_provider_root=<physical path of
   OA_HTML>
   ```

7. Under the Servlet aliases section, add the following:

   ```
   servlet.CatalogTemplates.code=oracle.apps.icx.catalog.ui.DownloadServlet
   ```

8. Save and close the file.

◗◗ To create ssp_init.txt file:

Create the *ssp_init.txt* file in the directory specified in the *zone.properties* file (COMMON_TOP/util/apache/1.3.9/Apache/Jserv/etc directory on UNIX and

COMMON_TOP\util\apache\1.3.9\Apache\Jserv\etc directory on Windows NT). The contents of the *ssp_init.txt* file should be as follows:

> **Note:** If you're running Apache on Windows NT platform, you must use forward-slashes (/) or double-backslashes (\\) when defining directory paths for the DBCLocation and TemplatePath parameters, against the normal Windows NT convention. For example,
>
> c:\\apps\\fnd\\11.5.0\\secure\\appsnt1_appsdb.dbc or
>
> c:/apps/fnd/11.5.0/secure/appsnt1_appsdb.dbc

```
FND_MAX_JDBC_CONNECTIONS=100
FND_IN_USE_CONNECTION_TIMEOUT=600
FND_UNUSED_CONNECTION_TIMEOUT=600
PLSQLAgent=/<DAD>
DBCLocation=/<location_of_dbc_file>/<name_of_dbc_file>
APPS_JDBC_DRIVER_TYPE=THIN
DB_HOST=<dbhostname.domain>
DB_PORT=<database_listener_port_number>
TemplatePath=/<oa_html_path>/
XMLParser=ORACLE_V2
icxCatalogTemplateRoot=/<oa_html_path>/<language_code>/
```

where:

- PLSQLAgent is the Oracle Applications’ WebDB 2.5 DAD. Rapid Install will default the WebDB 2.5 DAD to the Oracle SID.
- DBCLocation is the physical path of $FND_TOP/secure/<dbcfilename>.dbc,
- DB_HOST is the database hostname.domain, not the Web Server hostname,
- DB_PORT is the Net8 listener port number for your database, and
- TemplatePath is the path to OA_HTML (with a trailing slash).

A sample ssp_init.txt file is presented in the Appendix.

To unlimit file descriptors:

Unlimiting file descriptors from the *apachectl* file helps you control the number of instances spawned from it. The *apachectl* file is available in the $APACHE_TOP/bin directory on UNIX platforms.

1. Using a text editor, open the *apachectl* file.
2. Add the following line to the beginning of the file:

   ```
   ulimit -n 1024
   ```

3. Save and close the file.

> **Note:** The above option is only available on UNIX platforms.

◗◗ To verify correct version of OJSP:

Rapid Install installs two versions of OJSP named ojsp.jar and ojsp.jar.new. These libraries are located under COMMON_TOP/apache/1.3.9/Apache/Ojsp/lib. In order for Internet Procurement 11*i* to work properly, please verify that the current ojsp.jar file is identical to osjp.jar.new.

◗◗ To restart Apache:

You must restart Apache after modifying configuration files to clear the cache and to ensure that the new settings take effect.

1. Stop Apache Server by executing the following command:

   `COMMON_TOP/admin/scripts/adapcctl.sh stop` (UNIX)
   `adapcctl.cmd stop` (Windows NT console)

2. Start Apache Server by executing the following command:

   `COMMON_TOP/admin/scripts/adapcctl.sh start` (UNIX)
   `adapcctl.cmd start` (Windows NT console)

Apache Configuration for Multiple JVMs

Verify that you have finished Apache configuration for Oracle Internet Procurement 11*i* before attempting to configure multiple JVMs for Apache. For details on configuring Apache for Oracle Internet Procurement 11*i*, see: Apache Configuration for Oracle Internet Procurement on page 1-9. For more information regarding the following configuration, please refer to the document titled “How to: Scalability - Load-Balancing - Fault tolerance with Apache JServ 1.1". This document can be found at the Apache web site (http://www.apache.org).

◗◗ To modify jserv.conf file:

You will need to modify the JServ configuration file *jserv.conf* available in the COMMON_TOP/util/apache/1.3.9/Apache/Jserv/etc/ directory. The procedures presented here assume that you are configuring two JVMs in UNIX environment.

1. Using a text editor, open *jserv.conf* file.
2. Set the ApJServManual value to *on*. By default, this value is set to *off*.
3. Towards the end of the file, before the *</IfModule>* add the following in the Mount Point section:

   ```
   ApJServMount /servlets balance://set1/root

   ApJServBalance set1 PC1
   ApJServBalance set1 PC2

   ApJServHost PC1 ajpv12://<hostmachine>:<jserv_port1>
   ApJServHost PC2 ajpv12://<hostmachine>:<jserv_port2>

   ApJServRoute JS1 PC1
   ApJServRoute JS2 PC2
   ```

   where:

   - set1 is a set of equivalent JServs, which contains PC1 and PC2, the two Java virtual machines,
   - hostmachine is the name of the machine where Apache resides, and
   - jserv_port1 and jserv_port2 are the ports used by JServ. Both these ports must be free of any conflict.
4. Save and close the file.

◗◗ To prepare the jserv.properties files:

1. Create *jserv.properties1* and *jserv.properties2* files. You can do this by copying the *jserv.properties* file configured earlier. Note that these files must be created in the same directory as the *jserv.properties* file.
2. Modify the port specified in *jserv.properties1* and set it to the `<jserv_port1>` port configured in *jserv.conf* file.
3. Modify the port specified in *jserv.properties2* and set it to the `<jserv_port2>` port configured in *jserv.conf* file.

◗◗ To create scripts to start and stop the JVMs:

1. Create the following script to start JVMs. Name the script *startjvm*. Please note that this script is for UNIX environments.

```
#!/bin/csh
setenv IAS_TOP <location where Apache is installed>
setenv JSERV_CLASSPATH
"$IAS_TOP/Apache/jdk/lib/classes.zip:$IAS_
TOP/Apache/xdk/lib/classes111.zip:$IAS_TOP/jdbc/lib/classes111.zip:$IAS_
TOP/Apache/Jserv/libexec/ApacheJServ.jar:$IAS_
TOP/Apache/Jsdk/lib/jsdk.jar:$IAS_TOP/Apache/xdk/lib/xmlparserv2.jar:$IAS_
TOP/Apache/Ojsp/lib/ojsp.jar:$IAS_TOP/Apache/Ojsp/lib/servlet.jar:<location
of Java files>

unlimit descriptors

$IAS_TOP/Apache/jdk/bin/java -mx400m -classpath $JSERV_CLASSPATH
org.apache.jserv.JServ $IAS_TOP/Apache/Jserv/etc/jserv.properties1 >&
$IAS_TOP/Apache/Apache/logs/jvm1.log &

$IAS_TOP/Apache/jdk/bin/java -mx400m -classpath $JSERV_CLASSPATH
org.apache.jserv.JServ $IAS_TOP/Apache/Jserv/etc/jserv.properties2 >&
$IAS_TOP/Apache/Apache/logs/jvm2.log &
```

2. Create the following script to stop the JVMs. Name the script *stopjvm*. Please note that this script is for UNIX environment.

```
#!/bin/csh
setenv IAS_TOP <location where iAS is installed>
setenv JSERV_CLASSPATH
"$IAS_TOP/Apache/jdk/lib/classes.zip:$IAS_
TOP/Apache/xdk/lib/classes111.zip:$IAS_TOP/jdbc/lib/classes111.zip:$IAS_
TOP/Apache/Jserv/libexec/ApacheJServ.jar:$IAS_
TOP/Apache/Jsdk/lib/jsdk.jar:$IAS_TOP/Apache/xdk/lib/xmlparserv2.jar:$IAS_
TOP/Apache/Ojsp/lib/ojsp.jar:$IAS_TOP/Apache/Ojsp/lib/servlet.jar:<location
of Java files>

$IAS_TOP/Apache/jdk/bin/java -mx400m -classpath $JSERV_CLASSPATH
org.apache.jserv.JServ $APACHE_TOP/Apache/Jserv/etc/jserv.properties1 -s

$IAS_TOP/Apache/jdk/bin/java -mx400m -classpath $JSERV_CLASSPATH
org.apache.jserv.JServ $IAS_TOP/Apache/Jserv/etc/jserv.properties2 -s
```

3. Use the following command to start Apache.

```
COMMON_TOP/admin/scripts/adapcctl.sh start
```

4. Use `startjvm` to start the JVM and the `stopjvm` to stop it.

   where startjvm is the script you created in step 1 and stopjvm is the script you created in step2.

Administering the Apache Web Server

You can perform all the basic Apache server maintenance routines by executing the *apachectl* script (UNIX) or the *adapcctl.cmd* program (Windows NT). For more information on Administering the Apache Web Server, see Maintaining Oracle Applications, Release 11*i*.

Configuring the Logon Page (ICXINDEX.htm)

ICXINDEX.htm is the file users will access to log on to Oracle Internet Procurement 11*i*. It is already configured using WebDB listener if you are in the Release 11*i* application environment.

The contents of the *ICXINDEX.htm* file should be as follows.

◗◗ To configure the logon page:

- Using a text editor open the *ICXINDEX.htm* file. This file is available under the <OA_HTML>/<language> directory, for example, /d01/sspcomn/html/US on UNIX or d:\SSPCOMN\HTML\US on Windows NT.

  Find the following lines that contain:

  ```
  <FORM NAME="Logon0" ACTION="/<DAD>/oraclemypage.home" METHOD="POST"
  TARGET="_top">
  <INPUT TYPE="hidden" NAME="i_1" VALUE="">
  <INPUT TYPE="hidden" NAME="i_2" VALUE="">
  <INPUT TYPE="hidden" NAME="rmode" VALUE="2">
  ```

  where <DAD> is the WebDB 2.5 DAD.

Setup the Oracle Internet Procurement 11*i* Home Page

You must define the following profile options through Oracle Applications Release 11*i* to be able to bring up the Oracle Internet Procurement 11*i* home page:

- ICX:Requisition Server

You must also create an appropriate dbc file to enable Oracle Internet Procurement 11*i* to connect to the database.

The .dbc file is created by Rapid Install and is available in the FND_TOP/secure directory. Verify that the GUEST_USER_PWD in the dbc file is a valid Oracle Application user.

◗◗ To complete the setup to display the home page:

1. Log on to Oracle Applications Release 11*i* using the System Administrator responsibility.
2. Set the profile option **ICX:Requisition Server** to the following:

   ```
   <hostname.domain>:<apache_listener_port_number>
   ```

   For example, sspbox.us.oracle.com:8000

   where:

   - sspbox is the host machine,
   - us.oracle.com is the domain, and
   - 8000 is the listener port number for Apache.
3. Set the Applications Web Agent profile to the following:

   ```
   http://<hostname.domain>:<WebDB_listener_port>/<DAD>
   ```

   For example: http://sspbox.yourserver.com:9000/ssp530

   where:

   - sspbox is the host machine,
   - yourserver.com is the domain,
   - 9000 is the WebDB 2.5 listener port number, and
   - ssp530 is the name of the WebDB 2.5 DAD.
4. Verify that the **POR: Servlet Virtual Path** profile option is set to the following:

   ```
   oa_servlets
   ```

   where:

   - oa_servlets refers to the SERVLET zone specified in the *jserv.conf* file.

5. Navigate to the Form Functions window through **Application>Function**, then set up the following menu functions:

   POR_SSP_HOME

   POR_SSP_ECMANAGER

   Query up these functions, then select the Web HTML tab. Complete the URLs in the HTML call field for each function by replacing the host, domain, WebDB listener port and WebDF 2.5 DAD name values with your own values.

6. Login to SQL*Plus as the APPS user and set the session cookie domain in SQL*Plus using the following commands:

   ```
   SQL> update icx_parameters set session_cookie_domain='<domain>';
   SQL> commit;
   ```

   where:

   - <domain> is the name of your domain, for example, ‘yourserver.com’

7. Restart your server.

8. Verify that the setup steps you have completed so far are correct.

   a. Create a new user. Assign “Internet Procurement” responsibility to the new user.

   b. From your browser, go to the URL that brings up the Log On page (*ICXINDEX.htm* file) you configured earlier. For more information, see Configuring the Logon Page (ICXINDEX.htm) on page 1-16.

   c. Sign on as the new user you created in Step (a) above. This should bring up the Main Menu for Oracle Internet Procurement 11*i*.

   d. From the main menu, choose Internet Procurement. This should bring up the home page.

Note: Before you can use the full functionality of Oracle Internet Procurement 11*i*, you must perform the steps listed in the Oracle Internet Procurement Implementation Manual. Please refer to that guide for further information.

Procedures for Customization

Oracle Internet Procurement 11*i* allows you to customize some of the online help pages with information specific to your organization. You can customize the following pages:

- Purchasing News
- Purchasing Policies

Customizing Purchasing News Page

Oracle Internet Procurement 11*i* comes with a purchasing news page which is just a place holder. You can modify this page with purchasing news specific to your organization. The Purchasing News page provides an easy medium for informing your purchasing personnel about the important purchasing-related events.

◗◗ To customize purchasing news:

1. Using any HTML or text editor, open the *PORPNEWS.htm* file. This file is available in the language subdirectory of OA_HTML. OA_HTML is the virtual path for documents in the web server's setup.
2. Change the contents of the *PORPNEWS.htm* file as required.
3. Save the file.

Customizing Purchasing Policies Page

Oracle Internet Procurement 11*i* comes with a place holder page where you can insert the purchasing policies specific to your organization. You can modify this page to inform your purchasing personnel about purchasing policies, for example, about new policies or changes to existing ones.

◗◗ To customize purchasing policies:

1. Using any HTML or text editor open the *PORPOLCY.jsp* file. This file is available in the language subdirectory (for example, US) of the OA_HTML

directory. OA_HTML is the virtual path in the web server's setup for documents and templates.

2. Change the contents of the *PORPOLCY.jsp* file. Though it has a .jsp extension, *PORPOLCY.jsp* is an html file. Be sure to format the file using the appropriate html tags.
3. Save and close the file. Ensure that the file is saved with .jsp extension.

Note: You may need to restart the Apache server to display the updated file.

A

Sample Apache Configuration Files

Sample jserv.conf file

```
###############################################################################
#                     Apache JServ Configuration File                         #
###############################################################################

# Note: this file should be appended or included into your httpd.conf

# Tell Apache on win32 to load the Apache JServ communication module
#LoadModule jserv_module modules/ApacheModuleJServ.dll

# Tell Apache on Unix to load the Apache JServ communication module
# For shared object builds only!!!
#LoadModule jserv_module libexec/mod_jserv.so

<IfModule mod_jserv.c>

# Whether Apache must start Apache JServ or not (On=Manual Off=Autostart)
# Syntax: ApJServManual [on/off]
# Default: "Off"
ApJServManual off
```

```
# Properties filename for Apache JServ in Automatic Mode.
# In manual mode this directive is ignored
# Syntax: ApJServProperties [filename]
# Default: "./conf/jserv.properties"
ApJServProperties /d1/dbf/iasbase2/Apache/Jserv/etc/jserv.properties

# Log file for this module operation relative to Apache root directory.
# Set the name of the trace/log file.  To avoid possible confusion about
# the location of this file, an absolute pathname is recommended.
#
# This log file is different than the log file that is in the
# jserv.properties file. This is the log file for the C portion of Apache
# JServ.
#
# On Unix, this file must have write permissions by the owner of the JVM
# process. In other words, if you are running Apache JServ in manual mode
# and Apache is running as user nobody, then the file must have its
# permissions set so that that user can write to it.
# Syntax: ApJServLogFile [filename]
# Default: "./logs/jserv.log"
# Note: when set to "DISABLED", the log will be redirected to Apache error log
ApJServLogFile /d1/dbf/iasbase2/Apache/Jserv/logs/jserv.log

# Log Level for this module
# Syntax: ApJServLogLevel [debug|info|notice|warn|error|crit|alert|emerg]
# Default: info    (unless compiled w/ JSERV_DEBUG, in which case it's debug)
ApJServLogLevel notice

# Protocol used by this host to connect to Apache JServ
# (see documentation for more details on available protocols)
# Syntax: ApJServDefaultProtocol [name]
# Default: "ajpv12"
ApJServDefaultProtocol ajpv12

# Default host on which Apache JServ is running
# Syntax: ApJServDefaultHost [hostname]
# Default: "localhost"
#ApJServDefaultHost java.apache.org

# Default port that Apache JServ is listening to
# Syntax: ApJServDefaultPort [number]
# Default: protocol-dependant (for ajpv12 protocol this is "8007")
ApJServDefaultPort 8007
```

```
# The amount of time to give to the JVM to start up as well
# as the amount of time to wait to ping the JVM to see if it
# is alive. Slow or heavily loaded machines might want to
# increase this value.
# Default: 10 seconds
# ApJServVMTimeout 10

# Passes parameter and value to specified protocol.
# Syntax: ApJServProtocolParameter [name] [parameter] [value]
# Default: NONE
# Note: Currently no protocols handle this. Introduced for future protocols.

# Apache JServ secret key file relative to Apache root directory.
# Syntax: ApJServSecretKey [filename]
# Default: "./conf/jserv.secret.key"
# Warning: if authentication is DISABLED, everyone on this machine (not just
# this module) may connect to your servlet engine and execute servlet
# bypassing web server restrictions. See the documentation for more information
#ApJServSecretKey /d1/dbf/iasbase2/Apache/Jserv/etc/jserv.secret.key
ApJServSecretKey DISABLED

# Mount point for Servlet zones
# (see documentation for more information on servlet zones)
# Syntax: ApJServMount [name] [jserv-url]
# Default: NONE
# Note: [name] is the name of the Apache URI path to mount jserv-url on
#       [jserv-url] is something like "protocol://host:port/zone"
#  If protocol, host or port are not specified, the values from
#  "ApJServDefaultProtocol", "ApJServDefaultHost" or "ApJServDefaultPort"
#  will be used.
#  If zone is not specified, the zone name will be the first subdirectory of
#  the called servlet.
# Example: "ApJServMount /servlets /myServlets"
#  if user requests "http://host/servlets/TestServlet"
#  the servlet "TestServlet" in zone "myServlets" on default host
#  thru default protocol on defaul port will be requested
# Example: "ApJServMount /servlets ajpv12://localhost:8007"
#  if user requests "http://host/servlets/myServlets/TestServlet"
#  the servlet "TestServlet" in zone "myServlets" will be requested
# Example: "ApJServMount /servlets ajpv12://jserv.mydomain.com:15643/myServlets"
#  if user requests "http://host/servlets/TestServlet" the servlet
#  "TestServlet" in zone "myServlets" on host "jserv.mydomain.com" using
#  "ajpv12" protocol on port "15643" will be executed
ApJServMount /servlets /root
ApJServMount /servlet /root
ApJServMount /OA_SERVLET /root

# Whether <VirtualHost> inherits base host mount points or not
# Syntax: ApJServMountCopy [on/off]
# Default: "On"
# Note: This directive is meaninful only when virtual hosts are being used
ApJServMountCopy on

# Executes a servlet passing filename with proper extension in PATH_TRANSLATED
# property of servlet request.
# Syntax: ApJServAction [extension] [servlet-uri]
# Defaults: NONE
# Notes: This is used for external tools.
#ApJServAction .jsp /servlets/org.gjt.jsp.JSPServlet
#ApJServAction .gsp /servlets/com.bitmechanic.gsp.GspServlet
#ApJServAction .jhtml /servlets/org.apache.servlet.ssi.SSI
#ApJServAction .xml /servlets/org.apache.cocoon.Cocoon


# Enable the Apache JServ status handler with the URL of
# "http://servername/jserv/" (note the trailing slash!)
# Change the "deny" directive to restrict access to this status page.
<Location /jserv/>
  SetHandler jserv-status

  order deny,allow
  deny from all
  allow from localhost
</Location>

############################## W A R N I N G ##################################
# Remember to disable or otherwise protect the execution of the Apache JServ  #
# Status Handler (see right above) on a production environment since this may #
# give untrusted users the ability to obtain restricted information on your   #
# servlets and their initialization arguments such as JDBC passwords and      #
# other important information. The Apache JServ Status Handler should be      #
# accessible only by system administrators.                                   #
###############################################################################

</IfModule>
```

Sample jserv.properties File

```
###############################################################################
#                        Apache JServ Configuration File                      #
###############################################################################

################################ W A R N I N G ################################
# Unlike normal Java properties, JServ configurations have some important
# extentions:
#
#    1) commas are used as token separators
#    2) multiple definitions of the same key are concatenated in a
#       comma-separated list.
###############################################################################

#
# Execution parameters
#######################

# The Java Virtual Machine interpreter.
# Syntax: wrapper.bin=[filename] (String)
# Note: specify a full path if the interpreter is not visible in your path.
wrapper.bin=/d1/dbf/iasbase2/Apache/jdk/bin/java

# Arguments passed to Java interpreter (optional)
# Syntax: wrapper.bin.parameters=[parameters] (String)
# Default: NONE
wrapper.bin.parameters=-mx400m

# Apache JServ entry point class (should not be changed)
# Syntax: wrapper.class=[classname] (String)
# Default: "org.apache.jserv.JServ"

# Arguments passed to main class after the properties filename (not used)
# Syntax: wrapper.class.parameters=[parameters] (String)
# Default: NONE
# Note: currently not used

# PATH environment value passed to the JVM
# Syntax: wrapper.path=[path] (String)
# Default: "/bin:/usr/bin:/usr/local/bin" for Unix systems
#          "c:\(windows-dir);c:\(windows-system-dir)" for Win32 systems
# Notes: if more than one line is supplied these will be concatenated using
#        ":" or ";" (depending wether Unix or Win32) characters
#        Under Win32 (windows-dir) and (windows-system-dir) will be
#        automatically evaluated to match your system requirements

# CLASSPATH environment value passed to the JVM
# Syntax: wrapper.classpath=[path] (String)
# Default: NONE (Sun's JDK/JRE already have a default classpath)
# Note: if more than one line is supplied these will be concatenated using
#       ":" or ";" (depending wether Unix or Win32) characters. JVM must be
#       able to find JSDK and JServ classes and any utility classes used by
#       your servlets.
# Note: the classes you want to be automatically reloaded upon modification
#       MUST NOT be in this classpath or the classpath of the shell
#       you start the Apache from.
#
wrapper.classpath=/d1/dbf/iasbase2/Apache/Jserv/libexec/ApacheJServ.jar
wrapper.classpath=/d1/dbf/iasbase2/Apache/Jsdk/lib/jsdk.jar

# An environment name with value passed to the JVM
# Syntax: wrapper.env=[name]=[value] (String)
# Default: NONE on Unix Systems
#          SystemDrive and SystemRoot with appropriate values on Win32 systems

# An environment name with value copied from caller to Java Virtual Machine
# Syntax: wrapper.env.copy=[name] (String)
# Default: NONE

# Copies all environment from caller to Java Virtual Machine
# Syntax: wrapper.env.copyall=[true|false] (boolean)
# Default: false

# Protocol used for signal handling
# Syntax: wrapper.protocol=[name] (String)
# Default: ajpv12

#
# General parameters
######################

# Set the default IP address or hostname Apache JServ binds (or listens) to.
#
# If you have a machine with multiple IP addresses, this address
# will be the one used. If you set the value to localhost, it
# will be resolved to the IP address configured for the locahost
# on your system (generally this is 127.0.0.1). This feature is so
# that one can have multiple instances of Apache JServ listening on
# the same port number, but different IP addresses on the same machine.
# Use bindaddress=* only if you know exactly what you are doing here,
# as it could let JServ wide open to the internet.
# You must understand that JServ has to answer only to Apache, and should not
# be reachable by nobody but mod_jserv. So localhost is usually a
# good option. The second best choice would be an internal network address
# (protected by a firewall) if JServ is running on another machine than Apache.
# Ask your network admin.
# "*" _may_ be used on boxes where some of the clients get connected using
# "localhost"and others using another IP addr.
#
# Syntax: bindaddress=[ipaddress] or [localhost] or [*]
# Default: localhost
bindaddress=localhost

# Set the port Apache JServ listens to.
# Syntax: port=[1024,65535] (int)
# Default: 8007
port=8007

#
# Servlet Zones parameters
###########################

# List of servlet zones Apache JServ manages
# Syntax: zones=[servlet zone],[servlet zone]... (Comma separated list of
String)
# Default: NONE
zones=root

# Configuration file for each servlet zone (one per servlet zone)
# Syntax: [servlet zone name as on the zones list].properties=[full path to
configFile] (String)
# Default: NONE
# Note: if the file could not be opened, try using absolute paths.
root.properties=/d1/dbf/iasbase2/Apache/Jserv/etc/zone.properties

#
# Thread Pool parameters
#########################

# Enables or disables the use of the thread pool.
# Syntax: pool=[true|false] (boolean)
# Default: false
# WARNING: the pool has not been extensively tested and may generate deadlocks.
# For this reason, we advise against using this code in production environments.
```

```
pool=false

# Indicates the number of idle threads that the pool may contain.
# Syntax: pool.capacity=(int)>0
# Default: 10
# NOTE: depending on your system load, this number should be low for contantly
# loaded servers and should be increased depending on load bursts.
pool.capacity=10

# Indicates the pool controller that should be used to control the
# level of the recycled threads.
# Syntax: pool.controller=[full class of controller] (String)
# Default: org.apache.java.recycle.DefaultController
# NOTE: it is safe to leave this unchanged unless special recycle behavior
# is needed. Look at the "org.apache.java.recycle" package javadocs for more
# info on other pool controllers and their behavior.
pool.controller=org.apache.java.recycle.DefaultController

#
# Security parameters
#####################

# Enable/disable the execution of org.apache.jserv.JServ as a servlet.
# This is disabled by default because it may give informations that should
# be restricted.
# Note that the execution of Apache JServ as a servlet is filtered by the web
# server modules by default so that both sides should be enabled to let this
# service work.
# This service is useful for installation and configuration since it gives
# feedback about the exact configurations Apache JServ is using, but it should
# be disabled when both installation and configuration processes are done.
# Syntax: security.selfservlet=[true|false] (boolean)
# Default: false
# WARNING: disable this in a production environment since may give reserved
# information to untrusted users.
security.selfservlet=true

# Set the maximum number of socket connections Apache JServ may handle
# simultaneously. Make sure your operating environment has enough file
# descriptors to allow this number.
# Syntax: security.maxConnections=(int)>1
# Default: 50
security.maxConnections=50
```

```
# Backlog setting for very fine performance tunning of JServ.
# Unless you are familiar to sockets leave this value commented out.
# security.backlog=5

# List of IP addresses allowed to connect to Apache JServ. This is a first
# security filtering to reject possibly unsecure connections and avoid the
# overhead of connection authentication.
# <warning>
# (please don't use the following one unless you know what you are doing :
#    security.allowedAddresses=DISABLED
#   allows connections on JServ'port from entire internet.)
#   You do need only to allow YOUR  Apache to talk to JServ.
# </warning>
#
# Default: 127.0.0.1
# Syntax: security.allowedAddresses=[IP address],[IP Address]... (Comma
separated list of IP addresses)
#security.allowedAddresses=127.0.0.1

# Enable/disable connection authentication.
# NOTE: unauthenticated connections are a little faster since authentication
# handshake is not performed at connection creation.
# WARNING: authentication is disabled by default because we believe that
# connection restriction from all IP addresses but localhost reduces your
# time to get Apache JServ to run. If you allow other addresses to connect and
# you don't trust it, you should enable authentication to prevent untrusted
# execution of your servlets. Beware: if authentication is disabled and the
# IP address is allowed, everyone on that machine can execute your servlets!
# Syntax: security.authentication=[true,false] (boolean)
# Default: true
security.authentication=false

# Authentication secret key.
# The secret key is passed as a file that must be kept secure and must
# be exactly the same of those used by clients to authenticate themselves.
# Syntax: security.secretKey=[secret key path and filename] (String)
# Default: NONE
# Note: if the file could not be opened, try using absolute paths.
#security.secretKey=./etc/jserv.secret.key

# Length of the randomly generated challenge string (in bytes) used to
# authenticate connections. 5 is the lowest possible choice to force a safe
# level of security and reduce connection creation overhead.
# Syntax: security.challengeSize=(int)>5
# Default: 5
```

```
#security.challengeSize=5
#
# Logging parameters
####################

# Enable/disable Apache JServ logging.
# WARNING: logging is a very expensive operation in terms of performance. You
# should reduced the generated log to a minumum or even disable it if fast
# execution is an issue.  Note that if all log channels (see below) are
# enabled, the log may become really big since each servlet request may
# generate many Kb of log. Some log channels are mainly for debugging
# purposes and should be disabled in a production environment.
# Syntax: log=[true,false] (boolean)
# Default: true
log=true

# Set the name of the trace/log file.  To avoid possible confusion about
# the location of this file, an absolute pathname is recommended.
#
# This log file is different than the log file that is in the
# jserv.conf file. This is the log file for the Java portion of Apache
# JServ.
#
# On Unix, this file must have write permissions by the owner of the JVM
# process. In other words, if you are running Apache JServ in manual mode
# and Apache is running as user nobody, then the file must have its
# permissions set so that that user can write to it.
# Syntax: log.file=[log path and filename] (String)
# Default: NONE
# Note: if the file could not be opened, try using absolute paths.
log.file=/d1/dbf/iasbase2/Apache/Jserv/logs/jserv.log

# Enable the timestamp before the log message
# Syntax: log.timestamp=[true,false] (boolean)
# Default: true
log.timestamp=true

# Use the given string as a data format
# (see java.text.SimpleDateFormat for the list of options)
# Syntax: log.dateFormat=(String)
# Default: [dd/MM/yyyy HH:mm:ss:SSS zz]
log.dateFormat=[dd/MM/yyyy HH:mm:ss:SSS zz]
```

```
# Since all the messages logged are processed by a thread running with
# minimum priority, it's of vital importance that this thread gets a chance
# to run once in a while. If it doesn't, the log queue overflow occurs,
# usually resulting in the OutOfMemoryError.
#
# To prevent this from happening, two parameters are used: log.queue.maxage
# and log.queue.maxsize. The former defines the maximum time for the logged
# message to stay in the queue, the latter defines maximum number of
# messages in the queue.
#
# If one of those conditions becomes true (age > maxage || size > maxsize),
# the log message stating that fact is generated and the log queue is
# flushed in the separate thread.
#
# If you ever see such a message, either your system doesn't live up to its
# expectations or you have a runaway loop (probably, but not necessarily,
# generating a lot of log messages).
#
# WARNING: Default values are lousy, you probably want to tweak them and
# report the results back to the development team.

# Syntax: log.queue.maxage = [milliseconds]
# Default: 5000
log.queue.maxage = 5000

# Syntax: log.queue.maxsize = [integer]
# Default: 1000
log.queue.maxsize = 1000

# Enable/disable logging the channel name
# Default: false
# log.channel=false

# Enable/disable channels, each logging different actions.
# Syntax: log.channel.[channel name]=[true,false] (boolean)
# Default: false

# Info channel - quite a lot of informational messages
# hopefully you don't need them under normal circumstances
# log.channel.info=true

# Servlets exception, i.e. exception caught during
# servlet.service() processing are monitored here
# you probably want to have this one switched on
log.channel.servletException=true
```

```
# JServ exception, caught internally in jserv
# we suggest to leave it on
log.channel.jservException=true

# Warning channel, it catches all the important
# messages that don't cause JServ to stop, leave it on
log.channel.warning=true

# Servlet log
# All messages logged by servlets. Probably you want
# this one to be switched on.
log.channel.servletLog=true

# Critical errors
# Messages produced by critical events causing jserv to stop
log.channel.critical=true

# Debug channel
# Only for internal debugging purposes
# log.channel.debug=true
# Oracle XSQL Servlet
wrapper.classpath=/d1/dbf/iasbase2/Apache/xdk/lib/oraclexsql.jar
# Oracle JDBC (8.1.6)
wrapper.classpath=/d1/dbf/iasbase2/Apache/xdk/lib/classes111.zip
# Oracle XML Parser V2 (with XSLT Engine)
wrapper.classpath=/d1/dbf/iasbase2/Apache/xdk/lib/xmlparserv2.jar
# Oracle XML SQL Components for Java
wrapper.classpath=/d1/dbf/iasbase2/Apache/xdk/lib/oraclexmlsql.jar
# XSQLConfig.xml File location
wrapper.classpath=/d1/dbf/iasbase2/Apache/xdk/lib
#
# Oracle Servlet
wrapper.classpath=/d1/dbf/iasbase2/Apache/Ojsp/lib/servlet.jar
# Oracle Java Server Pages
wrapper.classpath=/d1/dbf/iasbase2/Apache/Ojsp/lib/ojsp.jar
# Oracle Java SQL
wrapper.classpath=/d1/dbf/iasbase2/Apache/Ojsp/lib/translator.zip
# Oracle JDBC
wrapper.classpath=/d1/dbf/iasbase2/jdbc/lib/classes111.zip
wrapper.classpath=/d1/dbf/iasbase2/orahome/forms45/java
# OJSP environment settings
wrapper.env=ORACLE_HOME=/d1/dbf/iasbase2
wrapper.env=ORACLE_SID=8ISID
wrapper.env=LD_LIBRARY_PATH=/d1/dbf/iasbase2/lib
```

Sample zone.properties file

```
###############################################################################
#                        Servlet Zone Configuration File                      #
###############################################################################

################################ W A R N I N G ################################
#
# Unlike normal Java properties, JServ configurations have some important
# extentions:
#
#    1) commas are used as token separators
#    2) multiple definitions of the same key are concatenated in a
#       comma-separated list.
#
###############################################################################

# List of Repositories
#######################

# The list of servlet repositories controlled by this servlet zone
# Syntax: repositories=[repository],[repository]...
# Default: NONE
# Note: The classes you want to be reloaded upon modification should be put
#       here.

repositories=/d4/db/ay11i_
appltop/ay11icomn/util/apache/1.3.9/Apache/Jserv/servlets,/d4/db/ay11i_
appltop/ay11icomn/java/apps.zip

# Classloader parameters
#########################

# Enable servlet class autoreloading.
# Syntax: autoreload.classes=[true,false] (boolean)
# Default: true
autoreload.classes=false

# Enable servlet resourced autoreloading (properties and other loaded resources)
# Syntax: autoreload.file=[true,false] (boolean)
# Default: true
autoreload.file=false

# Set the number of millisecond to wait before giving up on initializing a
servlet.
```

```
# (a timeout of zero means no timeout)
# Syntax: init.timeout=(long)>0
# Default: 10000 (10 secs)
init.timeout=10000

# Set the number of millisecond to wait before giving up on destroying a
servlet.
# (a timeout of zero means no timeout)
# Syntax: destroy.timeout=(long)>0
# Default: 10000 (10 secs)
destroy.timeout=10000

# Set whether or not to use cookies to maintain session state.
# If false, then response.encodeUrl() will always be the method
# to maintain session state. If true, then the servlet engine will
# attempt to set a cookie when request.getSession(true) is called.
# Syntax: session.useCookies=[true,false] (boolean)
# Default: true
session.useCookies=true

# Set the number of millisecond to wait before invalidating an unused session.
# Syntax: session.timeout=(long)>0
# Default: 1800000 (30 mins)
session.timeout=1800000

# Set how frequently (milliseconds) to check for timed-out sessions.
# Syntax: session.checkFrequency=(long)>0
# Default: 30000 (30 secs)
session.checkFrequency=30000

# SingleThreadModel Servlets parameters
########################################

# Set the initial capacity of the STM servlets pool.
# Syntax: singleThreadModelServlet.initialCapacity=(int)>1
# Default: 5
singleThreadModelServlet.initialCapacity=5

# Set the number of servlet instances should be added to the pool if found
empty.
# Syntax: singleThreadModelServlet.incrementCapacity=(int)>1
# Default: 5
singleThreadModelServlet.incrementCapacity=5

# Set the maximum capacity of the STM pool
# Syntax: singleThreadModelServlet.maximumCapacity=(int)>1
# Default: 10
singleThreadModelServlet.maximumCapacity=10

################### S E R V L E T    P A R A M E T E R S ######################

################################## N O T E ####################################
# When "classname" is specified, it means a Java dot-formatter full class name
# without the ".class". For example, a class with source file named
# "Dummy.java" with a package name "org.fool" is defined as "org.fool.Dummy".
#
# Since each servlet may have lots of private initialization data, Apache JServ
# allows you to store those servlet initArgs in a separate file. To do this,
# simply do not set any initArgs in this file: Apache JServ will then look for
# a file named "[servlet classname].initargs" in the same directory of that
# class. Note that this may work with even class archives.
###############################################################################

# Startup Servlets
###################

# Comma or space delimited list of servlets to launch on startup.
# This can either be a class name or alias.
# Syntax: servlets.startup=[classname or alias],[classname or alias],...
# Default: NONE
# servlets.startup=hello,snoop,org.fool.Dummy
servlets.startup=oracle.apps.icx.common.InitSystemProperties

# Servlet Aliases
##################

# This defines aliases from which servlets can be invoked.
# Each alias give a new instance of the servlet. This means that if a servlet
# is invoked both by class name and by alias name, it will result in _TWO_
# instances of the servlet being created.
# Syntax: servlet.[alias].code=[classname] (String)
# Default: NONE
# servlet.snoop.code=SnoopServlet
# servlet.hello.code=org.fool.Dummy
servlet.CatalogTemplates.code=oracle.apps.icx.catalog.ui.DownloadServlet

# Global Init Parameters
#########################

# Parameters passed here are given to each of servlets. You should put
# configuration information that is common to all servlets.
#
# The value of the property is a comma delimited list of "name=value" pairs
# that are accessible to the servlet via the method getInitParameter()
# in ServletConfig.
# Syntax: servlets.default.initArgs=[name]=[value],[name]=[value],...
# Default: NONE
# servlets.default.initArgs=common.to.everybody=Hi everybody!

# Servlet Init Parameters
##########################

# These properties define init parameters for each servlet that is invoked
# by its classname.
# Syntax: servlet.[classname].initArgs=[name]=[value],[name]=[value],...
# Default: NONE
# servlet.org.fool.Dummy.initArgs=message=I’m a dummy servlet

# Aliased Servlet Init Parameters
##################################

# These properties define init parameters for each servlet that is invoked
# by its alias.
# Syntax: servlet.[alias].initArgs=[name]=[value],[name]=[value],...
# Default: NONE
# servlet.snoop.initArgs=message=I’m a snoop servlet
# servlet.hello.initArgs=message=I say hello world to everyone
servlet.oracle.jsp.JspServlet.initArgs=translate_params=true
servlet.oracle.apps.icx.common.InitSystemProperties.initArgs=initFile=/d4/db/ay1
1i_appltop/ay11icomn/util/apache/1.3.9/Apache/Jserv/etc/ssp_init.txt
```