JD Edwards EnterpriseOne Tools

Business Services Server Reference Guide

9.2

JD Edwards EnterpriseOne Tools
Business Services Server Reference Guide

9.2

Part Number: E53621-10

Contents

Preface i

1 Understanding Business Services Server 1

Business Services Server Overview 1
Server Manager Overview 1

2 Configuring Business Services Server 5

Setting Up the Business Service Server Logging 5
Configuring the Business Services Server to Consume Web Services 6
Allowing for a PS_Token to be Received by the EnterpriseOne Login Module 12
Business Services Server Fault Tolerance 13
Configuring the Business Services Server for Media Object Operations 14

3 Configuring Business Services Server Security 15

Understanding Business Services Server Security 15
Securing WAS profiles (WebSphere application servers only) 16
Setting Up Anonymous Login 16
Enabling SSL for Provider and Consumer Business Services 21
Enabling SSL for HTTP Request/Reply 26
Enabling Secure File Transfer Protocol (SFTP) for Media Objects 33

4 Configuring Business Services Server Security for JAX-WS Based Business Services 35

Understanding Business Services Server Security for JAX-WS Based Business Services on WAS 35
Securing JAX-WS Based Business Services on WAS 35
Configuring WebSphere to Use Anonymous Login 36

5 Appendix A - Troubleshooting 39

Exceptions 39
JD Edwards EnterpriseOne Application Warnings 40

Business Services Server Logs 40

Preface

Welcome to the JD Edwards EnterpriseOne documentation.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc` .

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Information

For additional information about JD Edwards EnterpriseOne applications, features, content, and training, visit the JD Edwards EnterpriseOne pages on the JD Edwards Resource Library located at:

`http://learnjde.com`

Conventions

The following text conventions are used in this document:

Convention	Meaning
Bold	Boldface type indicates graphical user interface elements associated with an action or terms defined in text or the glossary.
Italics	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`Monospace`	Monospace type indicates commands within a paragraph, URLs, code examples, text that appears on a screen, or text that you enter.
> Oracle by Example	Indicates a link to an Oracle by Example (OBE). OBEs provide hands-on, step- by-step instructions, including screen captures that guide you through a process using your own environment. Access to OBEs requires a valid Oracle account.

1 Understanding Business Services Server

Business Services Server Overview

The Business Services Server enables JD Edwards EnterpriseOne to natively produce and consume web services. The Business Services Server is built on top of a Java 2 Platform, Enterprise Edition (J2EE) server, which can be an Oracle WebLogic Server or a WebSphere Application Server. Applications that are developed or run on the Business Services Server are written in the Java programming language.

This guide does not provide instructions on how to install the Business Services Server; it provides other Business Services Server configuration that you should consider, such as security options and logging.

For information about installing and setting up a Business Services Server, see:

- "Create a Business Services Server as a New Managed Instance" in the *JD Edwards EnterpriseOne Tools Server Manager Guide* .

  This section contains information on how to use Server Manager to install a J2EE container on the machine that you want to use as your Business Services Server.

- *"Working with Packages for Business Services" in the JD Edwards EnterpriseOne Tools Package Management Guide* .

  This section contains information on how to use the JD Edwards EnterpriseOne Package Deployment and Tools applications to build a package that contains business services and deploy that package to the J2EE container installed by Server Manager.

- Customers must conform to the supported platforms for the release as detailed in the JD Edwards EnterpriseOne Minimum Technical Requirements. In addition, JD Edwards EnterpriseOne may integrate, interface, or work in conjunction with other Oracle products. Refer to the following link for cross-reference material in the Program Documentation for Program prerequisites and version cross-reference documents to assure compatibility of various Oracle products.

  http://www.oracle.com/corporate/contracts/index.html

  See document 745831.1 (JD Edwards EnterpriseOne Minimum Technical Requirements Reference) on My Oracle Support. ***https://support.oracle.com/epmos/faces/DocumentDisplay?id=745831.1***

Server Manager Overview

You use Server Manager to manage the JD Edwards Enterprise Tools Releases. Please refer to the *JD Edwards EnterpriseOne Tools Server Manager Guide* for details on installing and configuring all JD Edwards EnterpriseOne server products. This guide contains reference information for functionality outside the Server Manager tool.

All configuration changes to configuration files (such as jde.ini, jas.ini, jdbj.ini, jdelog.properties, and so on) for any JD Edwards EnterpriseOne server managed by Server Manager should be accomplished through the Management Console interface of Server Manager. In addition to usability improvements, using Server Manager reduces the risk of introducing configuration errors by providing drop-down lists with the valid values where applicable.

Further, the tool provides a useful Audit History for any modifications made to configurations using Server Manager.

The Server Manager tool provides:

- Configuration Management

  Server Manager provides a web-based interface for managing the configuration of each managed server. Each configuration item is accessible along with integrated help describing the configuration setting.

- Audit History

  Server Manager maintains a history of changes made to the managed servers. This includes a history of each configuration change, each server start and stop, and each tools release update, including the user that performed the change or operation. The information is logged to a history file that you can query from within Server Manager.

- Configuration Comparison

  Use Server Manager to compare the configuration of two or more servers to identify configuration differences. This can be done directly through the Management Console application regardless of the platform or location of the actual JD Edwards EnterpriseOne server. You can also compare individual servers with the default configuration of the corresponding server group to which the server belongs.

- Web Based System Management

  You can securely access and manage your JD Edwards EnterpriseOne installation from anywhere using a standard web browser.

- Remote Deployment ad Upgrades

  You can install, uninstall, and update your JD Edwards EnterpriseOne servers regardless of their physical location or platform.

- Remote Operational Control

  You can start and stop any of your JD Edwards EnterpriseOne servers, Oracle J2EE application servers, or third-party J2EE application servers directly through the Management Console.

- Secure Administration Tasks

  Server Manager permits you to specify which existing JD Edwards EnterpriseOne users that may access the Management Console, including which servers users are allowed to view within the Management Console, and which administrative tasks users may perform on those servers for which the user is allowed to view.

- Integrated EnterpriseOne Software Management

  Use Server Manager to centrally maintain all your JD Edwards EnterpriseOne server tools releases, including the ability to copy the software to the remote server machines.

- Logical Server Grouping

  Server Manager allows grouping similar purposed servers into a logical grouping. These groups can include any of the server types such as Enterprise Server, Web Server, and so on. A default, or template, configuration is maintained for each server group.

- Application Release Independence

  Server Manager is compatible with any supported JD Edwards EnterpriseOne application release beginning with Application Release 8.10 through the currently supported release. There are no electronic software updates (ESUs) required to support Server Manager.

- Self-Contained Installation

  The Server Manager installation delivers all components that are necessary for the tool to run. There are no third-party requirements regardless of your existing or intended middleware topology (for example, Oracle WebLogic Server, WebSphere Application Server, or no application server).

2 Configuring Business Services Server

Setting Up the Business Service Server Logging

This section provides an overview of business service logging and discusses how to set up web service logging.

The business service framework package provides an interface that provides logging utilities for activities related to published business services and business services. You use log files to troubleshoot system behavior. The location of the business service and published business service log files is defined in the jdelog.properties file.

At runtime, the system updates the log file with these kinds of errors:

- Severe
- Warning
- Application
- Information

Setting Up Web Service Logging Using SM Console

To set up web service logging when using SM Console:

1. Login to the EnterpriseOne SeverManager Console.
2. Navigate to the specific EnterpriseOne Business Services Server Instance for which logging needs to be enabled.
3. In the Configuration section - Click on jdelog.properties Logging.

4. In the Log File Configuration screen; the following can be configured:

   Log File Name

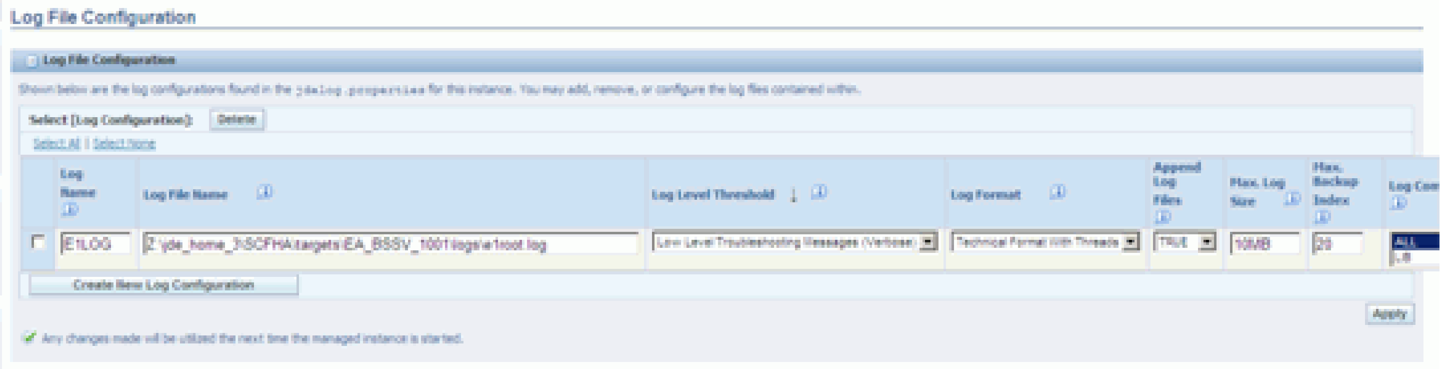
   Log Level Threshold - For detailed logging you can set Log Level to Low Level Troubleshooting Messages (Verbose).

   Log Format - For detailed logging you can set Log Format to Technical Format with Threads.

   Log Components - For detailed logging you can set Log Components to ALL so as to get information on all the Tools Components.

   Click on Apply.

   Restart the EnterpriseOne Business Services Server Instance for changes to take effect.



Log File Configuration

Log File Configuration

Shown below are the log configurations found in the jdelog.properties for this instance. You may add, remove, or configure the log files contained within.

Select [Log Configuration]: Delete

Select All | Select None

	Log Name	Log File Name	Log Level Threshold	Log Format	Append Log Files	Max. Log Size	Max. Backup Index	Log Com
□	E1LOG	Z:\jde_home_1\SCFHA\targets\EA_BSSV_1001\logs\e1root.log	Low Level Troubleshooting Messages (Verbose)	Technical Format With Threads	TRUE	10MB	20	ALL

Create New Log Configuration

Apply

✓ Any changes made will be utilized the next time the managed instance is started.

Configuring the Business Services Server to Consume Web Services

This section contains the following topics:

- *Understanding Business Services Server Configuration for Consuming Web Services*
- *Configuring JDENet*
- *Configuring an HTTP Proxy Server*
- *Enabling Business Services Running on WAS 7.0 to Consume Third-Party Web Services*
- *Enabling Business Services Running on WebLogic Server to Consume Third-Party Web Services*

Understanding Business Services Server Configuration for Consuming Web Services

You must start the JDENet kernel on the Business Services Server so that the Business Services Server can communicate with JD Edwards EnterpriseOne. You must also configure the Business Services Server with an HTTP proxy server so that outgoing connections can be made.

Configuring JDENet

The Business Services Server uses JDENet to communicate with a JD Edwards EnterpriseOne client (Microsoft Windows or HTML) or JD Edwards EnterpriseOne enterprise server. You can configure your system to use either one port or multiple ports to listen for and receive JDENet messages from the Enterprise Server.

Configuring a Business Services Server to Use a Single Port

If your Business Services Server is a standalone machine, or if you want to use only one port to listen for and receive JDENet messages from the JD Edwards enterprise server, you must configure the Object Configuration Manager (OCM). The client or enterprise server uses OCM to look up the machine name and port of the Business Services Server.

See "Setting Up OCM for Business Functions Calling Business Services" in the *JD Edwards EnterpriseOne Tools Business Services Development Guide* .

A JDENet kernel, SBFServerJavaKernel, is started on the Business Services Server. The business service server listens for messages for this kernel. This kernel listens for a Ping message and for messages that call a business service. The Business Services Server response to the Ping message indicates that the kernel is up and running on the Business Services Server.

Configure these settings in the [JDENET] section of the jdeinterop.ini file on the Business Services Server:

Parameter	Description
serviceNameListen=	Enter the port on which the business service server listens for JDENet messages. The setting for the Business Services Server and OCM must be the same. The port specified must be different than the serviceNameConnect port setting.
maxKernelRanges=	The maxKernelRanges setting defines the maximum allowed kernels on the Business Services Server. SBFServerJavaKernel is the only kernel supported on the Business Services Server, and this kernel is responsible for processing web service call messages from the client or enterprise server. Currently, the only allowed value for this property is 1.
enqueueTimeout=	The enqueueTimeout setting controls the amount of time the business service server waits while adding the JDENet messages in the queue, if the queue is full.

Configure these settings in the [JDENET_KERNEL_DEF1] section of the jdeinterop.ini file on the Business Services Server:

> **Note:** Values for all of the properties except maxNumberOfThread are static and do not change.

Parameter	Value or Description
krnlName=	SBFServerJavaKernel
processClassName=	oracle.e1.bssvfoundation.impl.jdenet.SBFServerJavaKernel

Parameter	Value or Description
startMessageRange=	16201
endMessageRange=	16450
maxNumberOfThread=	Defines the number of threads that will be started for the SBFServerJavaKernel, thereby defining the number of simultaneous web service call requests the kernel can process. This maxNumberOfThread value must be set appropriately, based on metrics such as call volume and server machine size.

Configuring a Clustered Business Services Server Instance for Consumer Business Services

You can use Server Manager to create a clustered business services server instance to support clustering for consumer business services. This clustering feature enables scalability, load balancing, and high availability for consumer business services by providing multiple incoming ports so that multiple servers in the cluster can open socket connection to different ports to receive incoming JDENet messages from the Enterprise Server, and the Enterprise Server can send JDENet messages to multiple clustered business services server ports.

See "Configuring a Clustered Business Services Server Instance for Consumer Business Services" in the *JD Edwards EnterpriseOne Tools Server Manager Guide* .

When you use Server Manager to create the business services server instance, you enter host/port information for all of the servers in the cluster in the Cluster Listen Host/Port field in the Miscellaneous area of the Configuration section. This information is stored with the property clusterHostPort under the CLUSTEROUTBOUND section of the JDEINTEROP.ini of the Busienss Services server instance. When the Business Services Server instance is started, the configuration information is read, and the host/port information is written to the Cluster Server Info table (F986102). Table F986102 includes a status for each business services service in the cluster. A status of AV indicates that the server is active and available. A status of NA indicates the server is stopped. When a business services server has a status of NA, the Enterprise Server does not attempt to send the JDENet message to that particular server.

When a consumer business service runs, the business service calls a business function that is running on the Enterprise Server. The business function reads the clustered business services server host/port information from the F986102 table and identifies only those business services servers in the cluster that have a status of AV and stores the host/port information for the available business services servers in a list. To minimize database calls, the business function does not read from the table if data already exists in the list. To ensure there is update-to-date information in the list, the business function reads and stores the information from the table and refreshes the list.

The business function tries to send the JDENet message to the first host/port item in the list. For subsequent requests, the business function tries to send the JDENet message to the next host/port item in the list in a round-robin fashion. Before sending the JDENet message, the business function pings the host/port. If the ping is successful, the message is sent; if the ping is not successful, the business function tries to send the message to the next active host/port in a round-robin fashion.

When a stopped business services server is started, this business services server is added to the list within 15 minutes, due to the database refresh for Table F986102. This enables load balancing and failover to be achieved effectively.

Configuring an HTTP Proxy Server

The Business Services Server also supports web services consumer functionality. The Business Services Server is deployed and configured using Server Manager. This section provides information on the additional configuration for consuming third-party web services.

An HTTP proxy server is commonly used for outgoing HTTP connections. When an HTTP proxy server is present, all Internet connections are made through the HTTP proxy server. The Business Services Server might need to make an Internet connection when calling an external web service. These HTTP proxy parameters are available for the Business Services Server:

Parameter	Description
http.proxyHost	The host name of the proxy server.
http.proxyPort	The port number of the proxy server.
http.nonProxyHosts	Optional. The machines and domains that do not need to be routed through the proxy server. Typically, these are all machines on the intranet. Multiple entries must be separated by a vertical bar (|).
http.proxyUser	Optional. The user name to be used for authentication on the proxy server.
http.proxyPassword	Optional. The password for the user name that is being used for authentication.

Configuring a Secure HTTP Connection

Both the Oracle WebLogic Server and the WebSphere application server support secure HTTP (HTTPS) connections. If you are using an HTTPS connection, you must manually configure HTTPS proxy parameters. The HTTPS proxy configurations are in addition to the HTTP proxy server configurations set by Server Manager. This table identifies the HTTPS proxy server parameters:

Parameter	Description
https.proxyHost	The host name of the proxy server.
https.proxyPort	The port number of the proxy server.
https.nonProxyHosts	Optional. The machines and domains that do not need to be routed through the proxy server. Typically, these are all machines on the intranet. Multiple entries must be separated by a vertical bar (|).
https.proxyUser	Optional. The user name to be used for authentication on the proxy server.
https.proxyPassword	Optional. The password for the user name that is being used for authentication.

Configuring an HTTP Proxy Server for WAS

Use these steps to set up an HTTP proxy server for WAS.

To set up an HTTP proxy on WAS:

1. On the IBM WebSphere Application Server, open the WebSphere Application Server console.
2. In the left navigation pane, click the Application Servers link under Servers.
3. Click the application server that is hosting the Business Services Server.
4. Expand Process Definition on the right, and then click the Java Virtual Machine (JVM) link.
5. On the JVM properties page, click Custom Properties.
6. Add the required http proxy configuration properties.
    - http.nonProxyHosts
    - http.proxyHost
    - http.proxyPassword
    - http.proxyPort
    - http.proxyUser
    - javax.xml.rpc.ServiceFactory - oracle.j2ee.ws.client.ServiceFactoryImpl
7. Add the required https proxy configuration properties.
    - https.nonProxyHosts
    - https.proxyHost
    - https.proxyPassword
    - https.proxyPort
    - https.proxyUser

Enabling Business Services Running on WAS 7.0 to Consume Third-Party Web Services

To consume third-party web services from Business Services Server proxy objects running on WAS, you must add specific JVM arguments. This section provides information on setting the JVM arguments for Business Services Server on WAS for consuming third-party web services.

To configure JVM arguments for WAS:

1. On the IBM WebSphere Application Server, open the WebSphere Application Server console.
2. In the left navigation pane, click to expand the **Servers** link.
3. Click to expand the **Server Types** link.
4. Click the **WebSphere application servers** link.
5. In the **Application servers** page, click the application server that is hosting the Business Services Server.
6. Under the **Server Infrastructure** section, expand **Java and Process Management** link.
7. Click the **Process Definition** link.
8. In **Additional Properties** on right, click the **Java Virtual Machine (JVM)** link.
9. On the **JVM properties** page, add these separate arguments in the **Generic JVM arguments** text box after the **default_path** argument:

```
-Djavax.xml.rpc.ServiceFactory=oracle.j2ee.ws.client.ServiceFactoryImpl
-Djavax.xml.soap.SOAPConnectionFactory=oracle.j2ee.ws.saaj.client.p2p.
 HttpSOAPConnectionFactory -Dweblogic.http.headers.enableHSTS=true
```

> **Note:** Enter each argument as a contiguous string with no spaces; however, each argument string must be separated with a space.

> **Note:** If you plan to implement HTTP Strict Transport Security, refer to this Oracle document: *Command Reference for Oracle WebLogic Server 14c* in the section entitled: *HTTP Strict Transport Security*.

10. Click **Apply**.
11. In **Messages**, click the **Save** link to save the JVM arguments directly to the master configuration.
12. In order for the changes to take effect, you must restart the Business Services Server Instance on WAS.

Enabling Business Services Running on WebLogic Server to Consume Third-Party Web Services

> **Note:** If you are using JD Edwards EnterpriseOne Tools Release 9.2 with JD Edwards EnterpriseOne Applications Release 9.2, JDeveloper 12c is installed on your system. With JDeveloper 12c, you can create only JAX-WS business services. If you are using JD Edwards EnterpriseOne Tools Release 9.2 with JD Edwards EnterpriseOne Applications Release 9.0 or 9.1, JDeveloper 11g is installed on your system. With JDeveloper 11g, you can create JAX-WS and JAX-RPC business services.

A business services .ear file, which was built for WebLogic Server (WLS) using JDeveloper, that contains business services proxy (consumer) objects created out of a secure web service running on WLS, requires that you perform the following tasks so that the business services proxy object can successfully run on WLS and consume the secure web service:

- Copy the policy file *Wssp1.2-2007-Https-UsernameToken-Plain.xml* to the domain where WLS is installed. For example, copy the policy file *Wssp1.2-2007-Https-UsernameToken-Plain.xml* to the Weblogic_Install_Path\\user_projects\domains\domain_name path.
- Copy the certificate file *DemoTrust.jks* to the domain where WLS is installed. For example, copy the *DemoTrust.jks* file needs to the Weblogic_ Install_Path\\user_projects\domains\domain_name path.

> **Note:** The policy file Wssp1.2-2007-Https-UsernameToken-Plain.xml can be found in the weblogic.jar or wseeclient.jar in the Weblogic_Install_Path\wlserver\ server\lib path. The certificate file DemoTrust.jks can be found in the Weblogic_Install_Path\wlserver\server\lib path.

Verifying the JAVA Argument for AIS Server (Oracle WebLogic Server Only)

If the AIS Server is deployed on Oracle WebLogic Server, you must make sure that the server configuration includes a JAVA argument for starting the server. To do so:

1. In the WebLogic Admin Console, locate the AIS Server instance.
2. Click the **Server Start** tab.

3. Verify that the following argument is in the Arguments field:

   ```
   -DUseSunHttpHandler=true
   ```

   ```
   -Dweblogic.http.headers.enableHSTS=true
   ```

> **Note:** If you plan to implement HTTP Strict Transport Security, refer to this Oracle document: *Command Reference for Oracle WebLogic Server 14c* in the section entitled: *HTTP Strict Transport Security*.

Adding a Custom Property for Improving Business Service Server Startup in IBM WebSphere 9.0

You can increase the performance of Business Service Server while starting it in WebSphere 9.0 by adding a custom JVM property to the server to disable implicit bean scanning. This setting skips the scanning of archives that do not contain a bean descriptor (beans.xml) file.

1. On the IBM WebSphere Application Server, open the WebSphere Application Server console.
2. In the left navigation pane, click to expand the **Servers** link.
3. Click to expand the **Server Types** link.
4. Click the **WebSphere application servers** link.
5. In the **Application servers** page, click the application server that is hosting the Business Services Server.
6. Under the **Server Infrastructure** section, expand the **Java and Process Management** link.
7. Click the **Process Definition** link.
8. In **Additional Properties** on the right, click the **Java Virtual Machine (JVM)** link.
9. In **Additional Properties** on the right, click the **Custom properties** link.
10. Click **New** to add the following property:

    ```
    Name: com.ibm.ws.cdi.enableImplicitBeanArchives
    Value: false. The default value is true.
    Description: Enter any optional description for the property.
    ```
11. Click **Apply**.
12. Click the **Save** link to save the JVM custom property directly to the master configuration.
13. Restart the Business Services Server instance on WebSphere Application Server in order for the changes to take effect.

Allowing for a PS_Token to be Received by the EnterpriseOne Login Module

In order to support all the mechanisms used by JD Edwards EnterpriseOne for authentication, the E1LoginModule allows a PS_TOKEN to be received. To use a PS_TOKEN for authentication, the web service call needs to provide additional information in the username field. At minimum both "DN=<E1 user id>" and "PS_TOKEN=true" must be specified. For example:

```
username - DN=KB5236952,PS_TOKEN=true
password - PS_TOKEN in string form
```

In addition, environment and role can be specified in combination with the PS_TOKEN indicator.

```
username - DN=KB5236952,ENV=STGAWSC1,ROLE=*ALL,PS_TOKEN=true
password - PS_TOKEN in string form
```

The sequence and case of the user name values are not significant.

Business Services Server Fault Tolerance

When a machine in the system goes down or is brought down, other machines in the system should gracefully degrade while it is down and reconnect once it is back up. For the Business Services Server the relevant machines are the Security Server and the Enterprise Server. The connection to the Enterprise Server is fault tolerant. If the Enterprise Server is down, the SOAP faults thrown from a called web service are descriptive and indicate the problem. If the Enterprise Server comes back up, subsequent web service calls connect correctly without restarting or any further administration of the Business Services Server. If connections to the Enterprise Server time out the connections are reestablished.

> **Note:** When the Security and Enterprise Servers are bounced, or kernel recycling occurs, the Business Services Server does not need to be bounced.

Enterprise Server is Unavailable

The connection from the Business Services Server to the Enterprise Server is based on a token. If the Enterprise Server is down or cannot be contacted, the exception thrown to the web service caller indicates that server login has failed. When the Security Server comes back up, the token is revalidated as necessary without any administrator interaction. If the Enterprise Server remains unavailable, the caller receives a descriptive message.

The following sample messages in the Business Services Server log indicate that the Enterprise Server is unavailable:

```
17 Sep 2007 16:27:13,140 [Line ?] [main] [SEVERE]  - [INTEROP]         Fail to execute
 BSFNMethod com.jdedwards.system.kernel.CallObjectSystemException: COSE#1002 Connection
 failed: LOCALHOST:6081 com.jdedwards.system.connector.dynamic.ServerFailureException:
 Fail to execute BSFNMethod com.jdedwards.system.kernel.CallObjectSystemException:
 COSE#1002 Connection failed: LOCALHOST:6081
com.jdedwards.system.connector.dynamic.ServerFailureException: Fail to execute BSFNMethod
 com.jdedwards.system.kernel.CallObjectSystemException: COSE#1002 Connection failed:
 LOCALHOST:6081
```

Security Server is Unavailable

The connection from the Business Services Server to the Security Server is based on a token. If the security server is down or cannot be contacted, the exception thrown to the web service caller indicates that server login has failed. When the Security Server comes back up, the token is revalidated as necessary without any administrator interaction.

When the first published business service call makes a new connection after the security token expires, the cached token is revalidated.

The following sample messages in the Business Services Server log indicate that the Security Server is unavailable:

```
17 Sep 2007 16:37:20,281 [Line ?] [main] [SEVERE]  - [INTEROP]         Cannot connect to any OneWorld Security
 Server.FAILURE: null    com.jdedwards.system.connector.dynamic.ServerFailureException: Cannot connect to any
 OneWorld Security Server.FAILURE: null
```

```
com.jdedwards.system.connector.dynamic.ServerFailureException: Cannot connect to any OneWorld Security
 Server.FAILURE: null
at com.jdedwards.system.connector.dynamic.Connector.loginBase(Unknown Source)
 at com.jdedwards.system.connector.dynamic.Connector.login(Unknown Source)
 at oracle.e1.bssvfoundation.impl.security.E1Principal.login(Unknown Source)
 at oracle.e1.bssvfoundation.impl.security.PrincipalCache.getIniPrincipal(Unknown Source)
 at oracle.e1.bssvfoundation.impl.jdenet.CallSBFHandler.callSBF(Unknown Source)
 at oracle.e1.bssvfoundation.base.TestBusinessService.callBSSVWithXMLFile(Unknown Source)
 at oracle.e1.bssv.JTRH90I10.RI_HTTP_ParseTransformSendMessage.main(RI_HTTP_ParseTransformSendMessage.java:22)
17 Sep 2007 16:37:20,281 [Line ?] [main] [DEBUG ]  - [BSSVFRAMEWORK]    [Context ID: ]    Login failed
Cannot connect to any OneWorld Security Server.FAILURE: null
17 Sep 2007 16:37:20,500 [Line ?] [main] [DEBUG ]  - [INTEROP]         Connector shut down completely
```

Business Services Server is Unavailable

When the Business Services Server is down, the Business Services Server log displays these messages:

```
4756/5824 WRK:Starting jdeCallObject                    Mon Nov 12 11:55:39.171089
    XMLRequest.cpp1260
            ICU0000017 - ICU CodePage for 1252 is ibm-1252.
4756/5824 WRK:Starting jdeCallObject                    Mon Nov 12 13:03:06.734001
    callsbfmsg.c311
            Error when sending JDENET message. JDENET Error = eConnectionFailed
4756/664 WRK:Starting jdeCallObject          Mon Nov 12 13:07:15.390053
 XMLRequest.cpp1260
            ICU0000017 - ICU CodePage for 1252 is ibm-1252.
4756/664 WRK:Starting jdeCallObject          Mon Nov 12 13:08:24.453002
 callsbfmsg.c311
            Error when sending JDENET message. JDENET Error = eConnectionFailed
4756/664 WRK:Starting jdeCallObject          Mon Nov 12 13:12:37.093001
 callsbfmsg.c311
            Error when sending JDENET message. JDENET Error = eConnectionFailed
4756/664 WRK:Starting jdeCallObject          Mon Nov 12 13:15:09.546001
 B953002.c367
            Internal Server Error during execution of Business Service.
```

Configuring the Business Services Server for Media Object Operations

The Business Services Server is deployed and configured using Server Manager. To support Media Object operations on the Business Services Server, you must identify the Media Object Server type, user, and password for accessing the Media Object Server shared location. In Server Manager, configure the settings in the Media Object Settings section of the Business Services Server to establish access to the Media Object Server. The settings enable you to use secure file transfer protocol (SFTP) for accessing media objects. For more information about SFTP, see *Enabling Secure File Transfer Protocol (SFTP) for Media Objects*.

The Media Object Settings also enable you to identify the media object file types (extensions) that you do not want allowed in EnterpriseOne. For a description of the settings, including valid values, refer to the Server Manager internal help for each setting.

3 Configuring Business Services Server Security

Understanding Business Services Server Security

> **Note:** This chapter covers the authentication of users of business services. For information about authorizing users to access published business service objects, see *"Managing Published Business Services Security" in the JD Edwards EnterpriseOne Tools Security Administration Guide* .

JD Edwards EnterpriseOne provides authentication security to ensure that published business service users are authenticated in JD Edwards EnterpriseOne. The Business Services Server uses the JD Edwards EnterpriseOne Login Module as the authentication mechanism for authenticating users against the security server.

The module is automatically installed during the deployment of a JD Edwards EnterpriseOne business services package to the Business Services Server and configured for all published services. The module uses Java authentication and authorization service (JAAS) to validate the JD Edwards EnterpriseOne users against the JD Edwards EnterpriseOne Security Server.

To allow access to JD Edwards EnterpriseOne published business services without providing user credentials, you must set up anonymous login. Anonymous login directs the application server to use the anonymous login credentials stored in the jdbj.ini file for user authentication, instead of the EnterpriseOne Login Module.

> **Note:**
>
> - *"Create a Business Services Server as a New Managed Instance" in the JD Edwards EnterpriseOne Tools Server Manager Guide* for more information on how to install a Business Services Server instance.
> - *"Working with Packages for Business Services" in the JD Edwards EnterpriseOne Tools Package Management Guide* for information on how to use JD Edwards EnterpriseOne to deploy published business services to the Business Services Server.
> - *"Applying TLS Configuration Setting for Server Manager Console and Agent" in the JD Edwards EnterpriseOne Tools Server Manager Guide* for information on how to apply TLS configuration setting for Server Manager Console and Agent.
> - *"Additional JVM Arguments" in the JD Edwards EnterpriseOne Tools Server Manager Guide* for information on additional JVM arguments.

Securing WAS profiles (WebSphere application servers only)

As specifically directed by the IBM WebSphere administration documentation, you should ensure that your WebSphere installation meets these general recommendations:

- Standalone configuration - Secure the default profile
- Network Deployment configuration - Create a new profile and secure that profile
- Recommended method of securing profile via WAS Integrated Solutions Console:
  - Administrative security: Enable Administrative security.
  - Application security: Enable application security.
  - Java 2 security: Disabled.
  - User account repositories: Federated repositories with the admin_user and admin_password (same as defined in soap.client.props file for profile).

Setting Up Anonymous Login

This section contains the following topics:

- *Understanding Anonymous Login*
- *Configuring WebSphere to Use Anonymous Login*
- *Configuring WebLogic to Use Anonymous Login*
- *Removing the Security Policy from an EnterpriseOne Web Service*

Understanding Anonymous Login

Anonymous login provides a mechanism to access published business services without providing JD Edwards EnterpriseOne user credentials. To enable anonymous login, you must disable the authentication mechanism (E1 Login Module) for a published business service in the application server. When the authentication mechanism is disabled, instead of using the user credentials of the consumer of the published business service for authentication, the application server uses the anonymous login credentials stored in the jdbj.ini file on the application server. These credentials are authenticated by the JD Edwards EnterpriseOne Security Server. The anonymous login password in the jdbj.ini file is encrypted.

You must configure anonymous login for each individual published business service by disabling the default authentication mechanism for that service. If the authentication mechanism is not disabled for a published business service, the user request will be rejected even if the anonymous login credentials have been entered in the jdbj.ini file.

You use Server Manager to enter these anonymous login credentials in the jdbj.ini configuration file:

- Bootstrap User
- Bootstrap User Password

- Bootstrap Role
- Bootstrap Environment

> **Note:** The anonymous login must be configured every time the Business Services Server is deployed.

See "JDBJ Bootstrap Session" in the *JD Edwards EnterpriseOne Tools Server Manager Guide* for information on how to configure these settings.

Configuring WebSphere to Use Anonymous Login

In WebSphere, you can disable security for a published business service, which directs the system to use anonymous login credentials.

> **Note:** If you want to configure Anonymous login for a JAX-WS web service package on IBM WebSphere, see *Configuring WebSphere to Use Anonymous Login*.

This section provides an example of turning off the security for the CustomerManager reference implementation, which is a fully functional example of a published business service. Use it as an example to help you disable security for a particular published business service so that the system uses anonymous login instead.

To set up anonymous login on IBM WebSphere:

1. Locate ibm-webservices-bnd.xmi and ibm-webservices-ext.xmi, which are in the following two locations:
   - *WebSphere Home*\AppServer\profiles\profile name\config\cells\Cell Name\applications\Application Name\deployments\Server\Web Module\WEB-INF
   - *WebSphere Home*\AppServer\profiles\profile name\installedApps\Cell Name\Application Name\Web Module Name\WEB-INF
2. Make a backup of these two files in both locations.
3. Using the following example of the RI_CustomerManager web service, delete the bold text from both the ibm-webservices-bnd.xmi and ibm-webservices-ext.xmi files. You must delete the code from these files in both locations where the files reside:
   - In the ibm-webservices-bnd.xmi file, delete the text shown in bold in this code sample:

```
 <wsdescBindings xmi:id="WSDescBinding_1185554582312" wsDescNameLink="RI_CustomerManager">
   <pcBindings xmi:id="PCBinding_1185554582312" pcNameLink="RI_CustomerManagerHttpPort"
 scope="Application">

     <securityRequestConsumerBindingConfig
 xmi:id="SecurityRequestConsumerBindingConfig_1185554610375910436757521891737910436757521891737">
                                                     <tokenConsumer
 xmi:id="TokenConsumer_1185554610375910436757521891737"
 classname="com.ibm.wsspi.wssecurity.token.UsernameTokenConsumer" name="UserTokenConsumer">
                                                      <valueType
 xmi:id="ValueType_1185554610375910436757521891737" localName="http://docs.oasis-open.org/
wss/2004/01/oasis-200401-wss-username-token-profile-1.0#UsernameToken" name="Username Token"/>
                                                        <jAASConfig
 xmi:id="JAASConfig_1186013028227" configName="e1BssvLogin"/>
                                                       <partReference
 xmi:id="PartReference_1185554610375910436757521891737" part="UserToken"/>
                                                    </tokenConsumer>
                                                    </securityRequestConsumerBindingConfig>

   </pcBindings>
 </wsdescBindings>
```

- In the ibm-webservices-ext.xmi file, delete the text shown in bold in this code sample:

```
 <wsDescExt wsDescNameLink="RI_CustomerManager" xmi:id="WsDescExt_1185554582328">
   <pcBinding pcNameLink="RI_CustomerManagerHttpPort" xmi:id="PcBinding_1185554582328">

     <serverServiceConfig
 xmi:id="ServerServiceConfig_118555460310966390350416797703696639035041679770369">
                                                       <securityRequestConsumerServiceConfig
 xmi:id="SecurityRequestConsumerBindingConfig_118555460310966390350416797703696639035041679770369">
                                                       <caller name="basicAuth"
 part="" uri="" localName="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#UsernameToken"/><requiredSecurityToken
 xmi:id="RequiredSecurityToken_11855546031096639035041679770369" name="UserToken" uri=""
 localName="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#UsernameToken" usage="Required"/>
                                                </securityRequestConsumerServiceConfig>
                                                    </serverServiceConfig>


   </pcBinding>
 </wsDescExt>
```

4. Restart the application server.

Configuring WebLogic to Use Anonymous Login

You use the WebLogic Server Admin Console to remove the security policy that is attached to a web service so that you can use Anonymous Login.

For WLS, there is a WebLogic service patch. For detail information about installing the service patch, see Bug 12761125.

> **Note:** If you have applied any other type of SSL certificate to WLS, this patch may not work. The workaround is to remove the certificate, then remove the WS-policy, and then apply the certificate again.

If you are running a WebLogic Server on a Windows platform, you must remove the class path from the server before you remove the security policy, and then you must add the class path back to the server after you remove the security policy.

Removing the Class Path from a WebLogic Server Running on a Windows Platform

> **Note:** The steps in this task are required only if you are running a WebLogic Server on a Windows platform. This task is not applicable if your WebLogic Server is running on a non-Windows platform.

Use these steps to remove and save the WebLogic server class path:

1. Log into the WebLogic Admin Console.
2. On the left side Domain Structure section, expand Environment, and then click the Server option.
3. From the list of servers, click on the server where your business service package is deployed.
   The server does not need to be running at this time.
4. On the right side Settings page, click the Configuration tab, and then click the Server Start tab
5. From the left side Change Center section, click the Lock & Edit button.
6. On the right side Settings page, copy the entire entry from the Class Path section and paste the entry in Notepad and save it.
7. On the Admin Console on the right side Settings page, delete the entry in the Class Path section.
8. Click Save.
   A message stating that the settings were successfully updates appears on the Setting page.

9. On the left side Change Center section, click Activate Changes.
10. Find the server that you selected above and do one of the following:
    - If the managed server is in a stopped state, start it
    - If the managed server is in a running state, then stop it and start it.

Now you can remove the security policy for a web service.

See *Removing the Security Policy from an EnterpriseOne Web Service*

Adding the Class Path Back to the WebLogic Server (Windows Platform Only)

> **Note:** The steps in this task are required only if you are running a WebLogic Server on a Windows platform. This task is not applicable if your WebLogic server is running on a non-Windows platform.

After you remove one or more security policies from your web services, you must add the class path back to your WebLogic Server so that web services that still have a security policy work.

Use the following steps to add the server class path back to the WebLogic Server:

1. From the Change Center section on left side of the admin console, expand Environment, click the Server option, and select the same server you selected in the previous set of steps.
2. On the right side Settings page, click the Configuration tab, and then click the Server Start tab
3. On left side Change Center section, click the Lock & Edit button.
4. On the right side Settings page, copy and paste the class path entry that you saved in Notepad from the previous task to the Class Path section.
5. Click Save.
6. On the left side Change Center section, click the Activate Changes button
7. Restart the server.

Now, web services with security policy attached and web services without a security policy can be run on the Windows platform.

Removing the Security Policy from an EnterpriseOne Web Service

You can use the WebLogic Server Admin Console to remove the security policy that is attached to an EnterpriseOne web service so that you can use Anonymous Login.

> **Note:** If you are running WLS 11g on a Windows platform, you must remove the class path from the WebLogic Server before you remove the security policy. See *Removing the Security Policy from an EnterpriseOne Web Service* If you are running WLS 12c on a Windows platform, you are not required to remove the server start class path.

When you remove the HTTP security policy, the authentication uses the BootStrap User ID and Password in jdbj.ini for BSSV. You are not able to use WS-Security in the Soap Header section because Anonymous login is mandatory.

Every time you redeploy the business service package, you must manually detach the security policy file from the web services using the WebLogic Admin Console or manually edit the weblogic-webservices-policy.xml file.

Note:

- The policy file name for JAX-WS is bssvpolicy.xml.
- The policy file name for JAX-RPC is Wssp1.2-20077-Https-Usernametoken-Plain.xml

WebLogic Server 12c with Tools Release 9.2.x and above

To Remove the weblogic-webservices-policy.xml policy file:

1. On your BSSV server, access the path D:\jde_home_wls\SCFHA\targets\DV_BSSV directory

   and create a backup for "owl_deployment" folder. Copy this backup folder to a different location in your system.
2. Stop the BSSV instance.
3. From the location `D:\jde_home_wls\SCFHA\targets\DV_BSSV\owl_deployment\E1Services-DV920-wls.ear\app\E1Services-DV920-web.war\WEB-INF`, open the file weblogic-webservices-policy.xmlin a notepad.
4. In the file weblogic-webservices-policy.xml, search your web service. For example, search for RI_AddressBookManagerPort.

   ```
   <port-policy> <port-name>RI_AddressBookManagerPort</port-name> <ws-policy> <uri>policy:bssvpolicy.xml</
   uri> <direction>both</direction> <status>enabled</status> </ws-policy> </port-policy>
   ```
5. Change the status to `deleted`.

   ```
   <port-policy> <port-name>RI_AddressBookManagerPort</port-name> <ws-policy> <uri>policy:bssvpolicy.xml</
   uri> <direction>both</direction> <status>deleted</status> </ws-policy> </port-policy>
   ```
6. Save the file.
7. In the location, `D:\Oracle\Middleware\user_projects\domains\BSSV_domain\servers\BSSV_DV\stage`, rename the BSSV instance folder (xxxx_backup) or delete it.
8. Start the BSSV instance.

   Open the WSDL, for example, RI_AddressBookManager, to verify that the policy is removed.

WebLogic server 11g and 12c with Tools Release 9.1.5.x

Use these steps to remove the security policy from an EnterpriseOne web service:

Note: The deployment must be active.

1. On the WebLogic Admin Console, go to Deployments and click the deployment that you want to modify.

   The list of services appears under the deployment that you selected.
2. Click the service from which you want to remove the security policy.
3. On the right side Settings page, click the Configuration tab, and then click the WS-Policy tab.

   The service endpoint and the policy attached to it appear on the Settings page.
4. Click the service end point.
5. From the left side Change Center section, click Lock & Edit.

   A list of policies appears in the Settings page.
6. Under Chosen Endpoint Policies on the Settings page, select the security policy to be removed and use the arrow button to move the policy to the Available Endpoint Policies on the left side.
7. Click OK.

   A message appears on the Settings page indicting that the deployment plan was successfully created and identifies the location of the deployment plan.

The first time that you remove a policy, the Plan.xml file is created under these two locations:

- -C:\jde_agent\SCFHA\targets\BSSV_WLS_7020\owl_deployment\E1Services-DV900-wls.ear\ app \Plan.xml
- -C:\Oracle\Middleware\user_projects\domains\BSSVHTTP\servers\BSSV_HTTP_7080\stage \BSSV_HTTP_7080\plan\Plan.xml

8. Repeat steps 2 through 7 to remove security policies from additional web services.

   After completing all of the policy removals, perform the next steps to update the deployment.
9. On the WebLogic Admin Console, go to Deployments and select the deployment that you modified.
10. Click Update.
11. In the next screen, select the option, *Update this application in place with new deployment plan changes.*

    A deployment plan must be specified for this option.
12. Click Next.
13. Click Finish.

Now the web services from which you removed the policy can be invoked without passing the user name and password (anonymous access) in the SOAP header.

> **Note:** If you are running WLS on a Windows platform, you must add the class path back to the WebLogic Server before you run your web service. See *Adding the Class Path Back to the WebLogic Server (Windows Platform Only)*

> **Note:** To reattach a policy to a web service, use the above steps, except in Step 6, move the security policy from the left side (Available Endpoint Polices) to the right side (Chosen Endpoint Policies).

Enabling SSL for Provider and Consumer Business Services

This section describes how to enable SSL for business services running on an IBM WebSphere Application Server (WAS) for both Provider and Consumer Scenarios. SSL is enabled for business services running on Oracle WebLogic Server by default.

You can disable SSL for business services running on the Oracle WebLogic Server. See *Removing the Security Policy from an EnterpriseOne Web Service*

Configuring SSL with IBM WebSphere Application Server (WAS)

The section describes how to configure Secure Socket Layer (SSL) with the IBM WebSphere Application Server and includes these tasks:

- Configuring SSL on the IBM HTTP Server
- Configuring SSL on IBM WebSphere
- Configuring BSSV for SSL and WebSphere and Server Manager

Configuring SSL on the IBM HTTP Server

For production environments, we recommend you request a Self-Signed Certificate from a Certificate Authority. For instructions to request a CA-Signed Personal Certificate, refer to the IBM Info Center. The procedure in this section assume you have already obtained the CA-Signed Personal Certificate.

To configure SSL on the IBM HTTP Server:

1. Start the Key Management Utility by navigating the following path:

   Start > Programs > IBM HTTP Server V 6.1 > Start Key Management Utility
2. Create a folder named keys in the HTTP Server directory.
3. Start the ikeyMan utility which is located in your HTTP Server's bin directory (for Windows platforms, this path is typically:

   ```
   C:/WebSphere/IBM HTTP Server/bin
   ```
4. In the ikeyMan utility, create a Key Database File by selecting:

   Key Database File > New
5. At the prompt, complete these fields:

Field	Values
Key Database Type	CMS Note that only CMS is supported with the IBM HTTP Server.
File Name	serverkey.kdb
Location	C:\WebSphere\IBM HTTP Server\keys

6. Enter the password (for example, serverkey) and select this option:

   stash the password file
7. Click the **OK** button.
8. From the drop down box, select this option:

   Personal Certificates
9. Click the **New Self-Signed** button.
10. On the next screen complete these fields:

Field	Values
Key Label	Enter any label. For example: ○ server_cert
Version	X509V3

Field	Values
Key Size	1024
Common Name	Enter a fully qualified server name. For example: ○ denicdep5.mlab.jdedwards.com
Organization	Enter your organization name. For example: ○ Oracle
Country or region	Enter your country or region. For example: ○ US
Validity Period	Enter the validity for your certificate. For example: ○ 365 days

11. Click the **OK** button.

    The program displays your certificate in the list.
12. Delete all other certificates that might exist.
13. Open the httpd.conf file in a text editor, and add the following virtual host definition.

> **Note:** The text in the httpd.conf file is case sensitive; type the host definition exactly as shown. If you have already configured a port on the HTTP Server (for example, port 85), the file will include an Alias. You should use the same alias under your Virtual Host definition as shown by the **bold** segment in the section file sample below.

For Websphere 8.5.x make the changes below:

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
<IfModule mod_ibm_ssl.c>
Listen 443
<VirtualHost denicdep5.mlab.jdedwards.com:443>
Alias /jde "C:/WebSphere/AppServer/installedApps
                                       /denicdep5Node01/EA_JS_85.ear/webclient.war"
SSLEnable
</VirtualHost>
</IfModule>
SSLDisable
KeyFile "C:/WebSphere/IBMIHS/keys/serverkey.kdb"
```

For Websphere 9.0 make the changes below (Release 9.2.1):

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
<IfModule mod_ibm_ssl.c>
# IPv6 support:
Listen 13445
<VirtualHost *:13445>
```

```
Alias /jde "C:\Program Files\IBM\WebSphere\AppServer\profiles
\AppSrv05\installedApps\den01eatNode06Cell\HTML_7987.ear\webclient.war"
SSLEnable
</VirtualHost>
<Directory "C:\Program Files\IBM\WebSphere\AppServer\profiles
\AppSrv05\installedApps\den01eatNode06Cell\HTML_7987.ear\webclient.war\WEB_INF">
Require all denied
</Directory>
<Directory "C:\Program Files\IBM\WebSphere\AppServer\profiles
\AppSrv05\installedApps\den01eatNode06Cell\HTML_7987.ear\webclient.war">
Require all granted
</Directory>
SSLDisable
KeyFile C:\HTTPServer3\keys\serverkey.kdb
```

For BSSV, the following Virtual Host definition should be used.

For Websphere 8.5.x make the changes below:

```
Listen 0.0.0.0:443
## IPv6 support:
<VirtualHost *:443>
Alias /PD812_WEB "C:\WebSphere61\AppServer\profiles\BSSV/installedApps/[node_name]/
BSSV_PD_93.ear\PD812_WEB.war"
SSLEnable
</VirtualHost>
<Directory "C:\WebSphere61\AppServer\profiles\BSSV/installedApps/[node_name]/
BSSV_PD_93.ear\PD812_WEB.war\WEB_INF">
Order Deny,Allow
Deny from All
</Directory>
<Directory "C:\WebSphere61\AppServer\profiles\BSSV/installedApps/[node_name]/
BSSV_PD_93.ear\PD812_WEB.war">
Order Deny,Allow
Allow from All
</Directory>
</IfModule>
KeyFile C:\WebSphere61\IHS2\keys\serverkey.kdb
SSLDisable
# End of example SSL configuration
```

For Websphere 9.0 make the changes below (Release 9.2.1):

```
Listen 0.0.0.0:443
## IPv6 support:
<VirtualHost *:13445>
Alias /PD812_WEB "C:\Program Files\IBM\WebSphere61\AppServer\profiles\BSSV/
installedApps/[node_name]/BSSV_PD_93.ear\PD812_WEB.war"
SSLEnable
</VirtualHost>
<Directory "C:\Program Files\IBM\WebSphere61\AppServer\profiles\BSSV/installedApps/
[node_name]/BSSV_PD_93.ear\PD812_WEB.war\WEB_INF">
Require all denied
</Directory>
<Directory "C:\Program Files\IBM\WebSphere61\AppServer\profiles\BSSV/installedApps/
[node_name]/BSSV_PD_93.ear\PD812_WEB.war">
Require all granted
</Directory>
</IfModule>
KeyFile C:\HTTPServer3\keys\serverkey.kdb
```

```
SSLDisable
# End of example SSL configuration
```

Configuring SSL on IBM WebSphere

To configure SSL on the IBM HTTP Server:

1. Log on to your WebSphere Admin Console.
2. Navigate to Environment > Virtual Hosts.
3. Select your virtual host.
   For example, if you initially installed your application on port 85, then the virtual host should have a name similar to VH_EA_JS_85.
4. Under the virtual host, select Additional Properties > Host Aliases.
5. Under Host Aliases, click the **New** button.
6. Create a new host alias by completing these fields:

Field	Values
Host	Enter a fully qualified server name. For example: ○ denicdep5.mlab.jdedwards.com
Port	Enter the default SSL port number, which is: ○ 443

7. Regenerate the plugin and restart your Application Server.
8. Select your particular webserver.
9. Select Plug-in properties.
10. Click on copy to web server key store directories.

11. Restart application server and HTML instance.
    You should be able to login to the following URL:
    https://*fully_qualified_server_name*/jde/E1Menu.maf

Configuring BSSV for SSL and WebSphere and Server Manager

This section describes the additional steps that are required in order for BSSV to work with SSL and WebSphere and the Server Manager for JD Edwards EnterpriseOne.

1. From the WAS Admin Console, extract the signer and personal certificates from the node default trust store and node default key store.
2. Using the ikeyman tool, open the dummy client key file (DummyClientKeyFile.jks) and import the personal signer certificate that you extracted in Step 1 of this procedure.
3. Open the dummy client trust file (DummyClientTrustFile.jks) and import the signer certificate that you extracted in Step 1 of this procedure.
4. Using the ikeyman tool, open the plugin-key.kdb of the HTTP Server that is configured with the WAS Profile hosting the BSSV Server.
5. Import the personal signer and signer certificates extracted in Step 1 of this procedure into the plugin-key.kdb file.
6. Modify the httpd.conf file and change the key store value as shown below:

```
keystore=plugin-key.kdb
```

7. If you want to disable HTTP access and only have HTTPS (SSL) access, you must comment the include file for BSSV that was automatically added by Server Manager. The name of the Server Manage configuration file is scf_xxxx.conf.
8. After completing these steps and restarting HTTP server, the secure https url should be working properly.

Enabling SSL for HTTP Request/Reply

In JD Edwards EnterpriseOne, you can configure a business service to communicate with a third-party system using HTTP POST. You can secure communication between the Business Services Server and third-party sites by using the Secure Sockets Layer (SSL) protocol.

This section describes:

- *INI Configuration Changes for Communication Over SSL*
- *Configuring Production Application Server to Work with Certificates*

INI Configuration Changes for Communication Over SSL

The KEYSTORE and TRUST_STORE sections of the jdeinterop.ini file contain parameters that you must complete to enable SSL for HTTP Request/Reply. When you create a certificate for communication over SSL, the values that you enter for the certificate should match the values set for these parameters.

The following parameters in the KEYSTORE section of the jdeinterop.ini are used for the SSL configuration for HTTP Request/Reply:

Parameter	Description
keystorefile=	The path to the keystore file.
keystorepasswd=	The keystore password.
keyalias=	The keystore alias name.
certificatepasswd=	The keystore certificate password.

Note: The default settings for these parameters are blank.

The following parameters in the TRUST_STORE section of the jdeinterop.ini are used for the SSL configuration for HTTP Request/Reply:

Parameter	Description
truststorefile=	The path to the truststore file.

Parameter	Description
truststorepasswd=	The truststore password.

Note: The default settings for these parameters are blank.

Configuring Production Application Server to Work with Certificates

This section describes how to:

- *Configuring the HTTP Adapter Service*
- *Configuring the Listener Service*

Configuring the HTTP Adapter Service

Perform these tasks to configure the HTTP Adapter Service:

- Configure client authentication.
- Check the trustability of the server during handshake.

To configure client authentication:

Create a certificate request (CSR) using keytool.

1. Go to the HTTP Adapter deployed location.

   ```
   ../WEB-INF/classes/.
   ```
2. From a command prompt navigate to:

   ```
   <Business Services deployed location>/WEB-INF/classes/.
   ```
3. Use the following commands to create a certificate request:

   ```
   <JAVA_HOME>\bin\keytool -genkey -keyalg RSA -alias httpclientcer -keystore HTTPAdapterKS.keystore
   -keypass httpadapter -storepass httpadapter -dname "CN=Oracle,OU=Oracle,O=Oracle USA L=Redwood
   Shores,S=CA,C=US"
   ```

   Provide all the details for generating the key.

   ```
   <JAVA_HOME>\bin\keytool -certreq -alias httpclientcer -file clientkeyCSR -keystore HTTAdapterKS.kestore
   -keypass httpadapter -storepass httpadapter
   ```

   The preceding command generates the certificate request and writes to a file clientkeyCSR.
4. You obtain the user certificate from a certification authority by submitting the generated CSR and saving it to a file HTTPAdapter.cer.
5. Obtain the certification authority root certificate (rootCA.cer) and intermediate CA certificate (rootInterCA.cer).
6. Import the signer certificates rootCA.cer and rootInterCA.cer in to HTTP Adapter's keystore using this command:

```
<JAVA_HOME>\bin\keytool -import -alias rootCAcer -file rootCA.cer -keystore HTTAdapterKS.keystore -keypass httpadapter -storepass httpadapter
```

```
<JAVA_HOME>\bin\keytool -import -alias rootInterCAcer -file rootInterCA.cer -keystore HTTAdapterKS.keystore -keypass httpadapter -storepass httpadapter
```

7. Import the certificate HTTPAdapter.cer in to the HTTP Adapter's key store using the following command:

   ```
   <JAVA_HOME>\bin\keytool -import -v -alias AliasName -file HTTPAdapter.cer -keystore HTTAdapterKS.keystore -keypass KeyPassword -storepass httpadapter
   ```

   Where AliasName is the alias of the certificate. This value must be updated in the jdeinterop.ini file for keyalias parameter after the certificate is imported.

   Where KeyPassword is the password for the certificate stored in the keystore. This value must be updated in the jdeinterop.ini file for property certficatepasswd after the certificate is imported

To check the trustability of the server during handshake:

Obtain the SSL certificate (ServerRoot.cer) of server's certificate root CA.

1. Go to the HTTP Adapter deployed location.

   ../WEB-INF/classes/.
2. From a command prompt navigate to:

   ```
   <Business Services deployed location>/WEB-INF/classes/
   ```
3. Import the certificate ServerRoot.cer in to the HTTP Adapter's trust store using the following command:

   ```
   <JAVA_HOME>\bin\keytool -import -v -trustcacerts -alias AliasName -file ServerRoot.cer -keystore cacerts -keypass KeyPassword -storepass passward
   ```

   where AliasName is the name for alias of the certificate.

   where KeyPassword is the password for the certificate stored in the keystore.

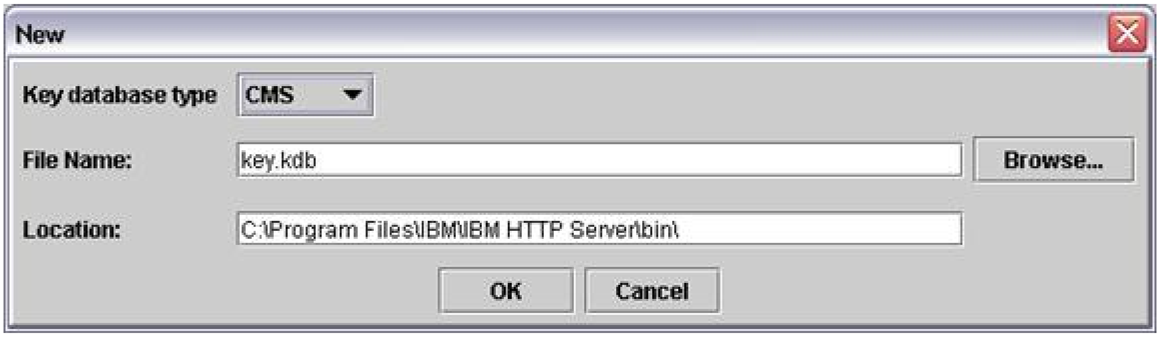
Configuring the Listener Service

Perform these tasks to configure the listener service:

- Configure SSL for the Listener Service on a IBM HTTP Server.

To configure SSL for the listener service on an IBM HTTP Server:

1. Open IKeyman tool (ikeyman.bat under:<IBM_HTTP_SERVER_INSTALL_ROOT>\bin).



Key database type: CMS

File Name: key.kdb

Location: C:\Program Files\IBM\IBM HTTP Server\bin\

2. On the New screen, select CMS from the Key database type list and complete these fields to create a new key database file:

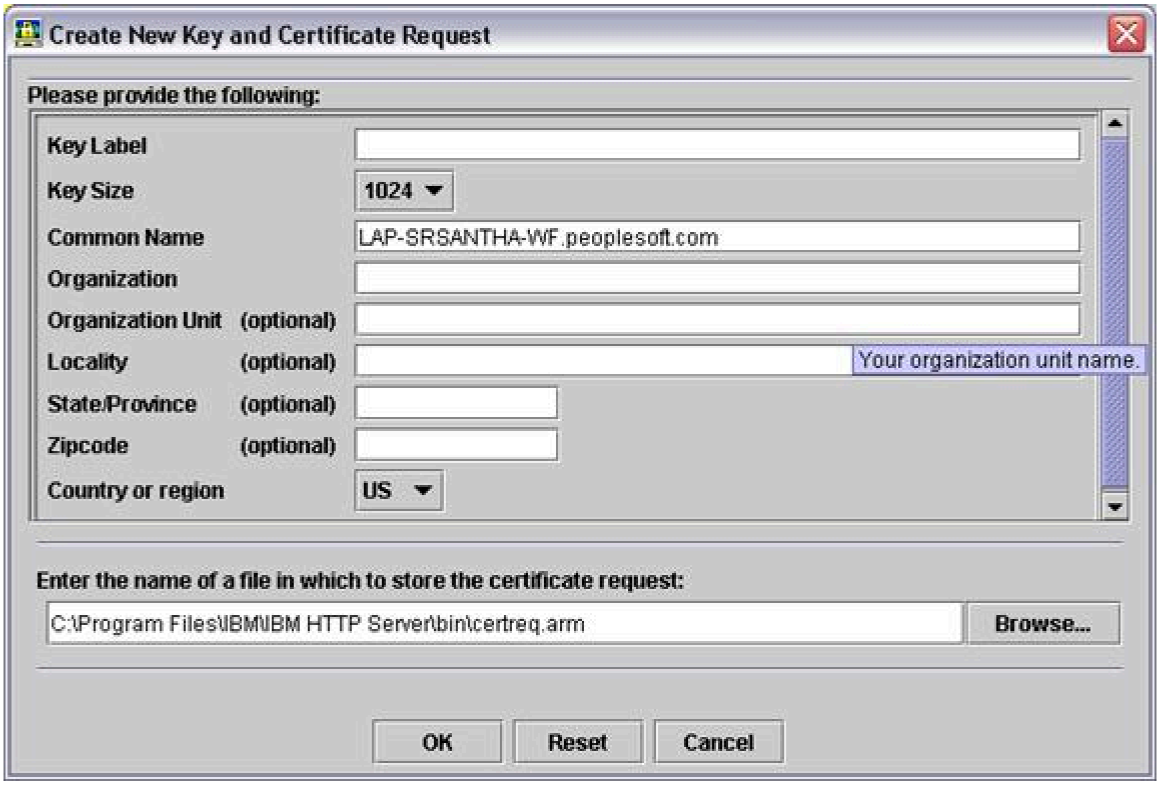
- File name

  Enter a name or click the Browse button to select a key database file.
- Location

  Enter the path to the key database file.

3. Click OK.

   The Password Prompt window opens.
4. On Password Prompt, enter a password in the Password and Confirm Password fields, and then select the "Stash password to a file?" option.



Create New Key and Certificate Request

Please provide the following:

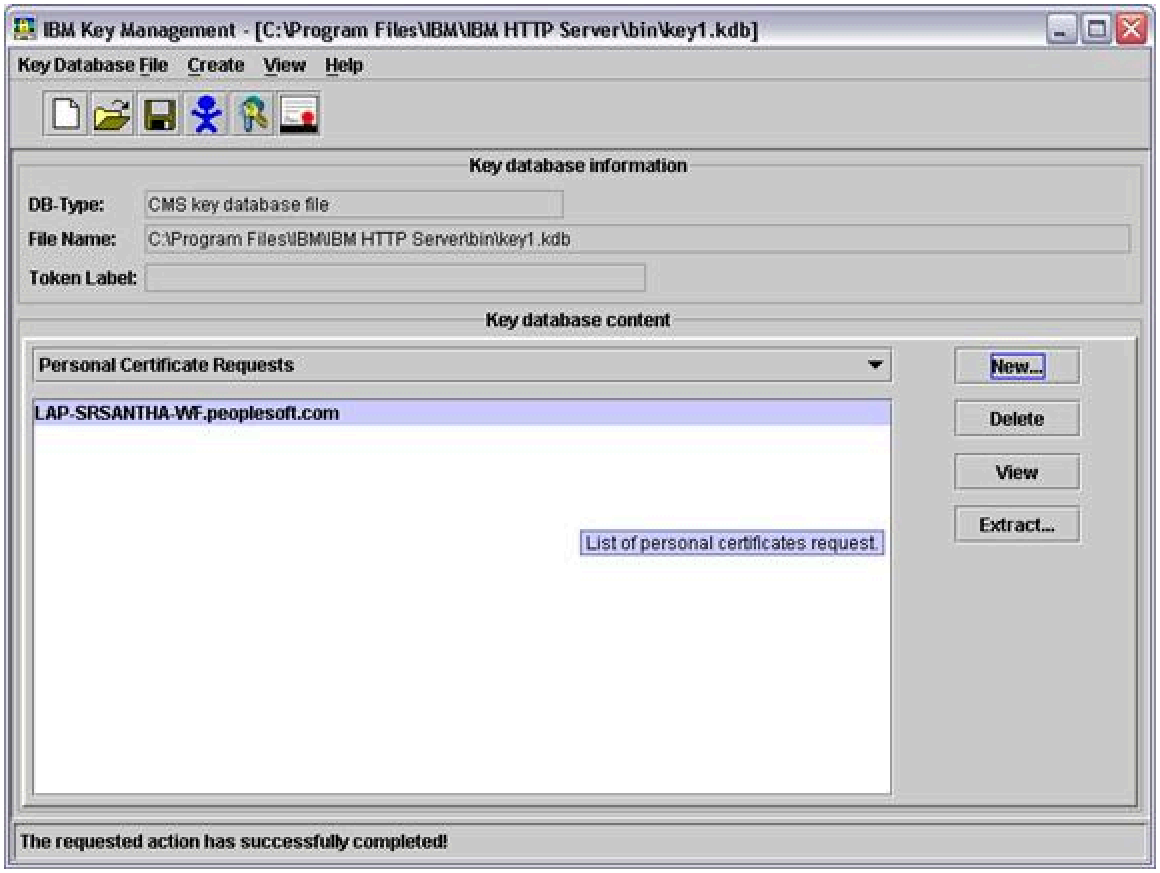
Key Label		
Key Size		1024
Common Name		LAP-SRSANTHA-WF.peoplesoft.com
Organization		
Organization Unit	(optional)	
Locality	(optional)	
State/Province	(optional)	
Zipcode	(optional)	
Country or region		US

Enter the name of a file in which to store the certificate request:

C:\Program Files\IBM\IBM HTTP Server\bin\certreq.arm

5. On Create New Key and Certificate Request, enter the name of the new certificate in the Common Name field. Enter the name of the file where the certificate request is stored, and click OK.

6. Select Personal certificate requests from Key database context menu and click New.


IBM Key Management - [C:\Program Files\IBM\IBM HTTP Server\bin\key1.kdb]
Key Database File
Create
View
Help
Key database information
Key database content
New...
Delete
View
Extract...
List of personal certificates request.

DB-Type: CMS key database file

File Name: C:\Program Files\IBM\IBM HTTP Server\bin\key1.kdb

Token Label:

Personal Certificate Requests

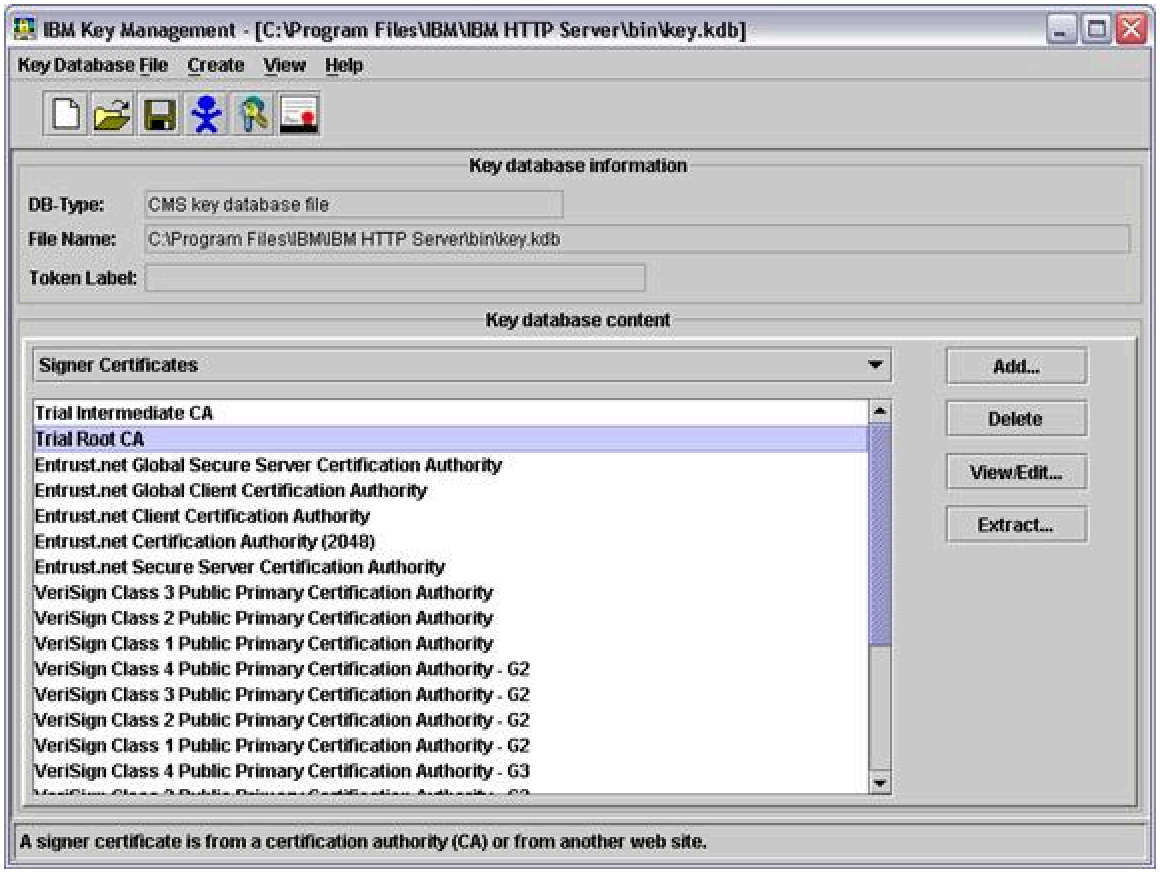
LAP-SRSANTHA-WF.peoplesoft.com

The requested action has successfully completed!

7. Provide the required information. The Certificate Request File is created at <IBM_HTTP_SERVER_INSTALL_ROOT>\bin. By default it is certreq.arm.
8. Create a CSR at any Certificate Authority with the Certificate Request information contained in the Certificate Request File.

   Also, obtain Root CA and Intermediate CA certificate from the Certificate Authority vendor.

9. Select the Signer Certificates option from Key database content.



DB-Type: CMS key database file

File Name: C:\Program Files\IBM\IBM HTTP Server\bin\key.kdb

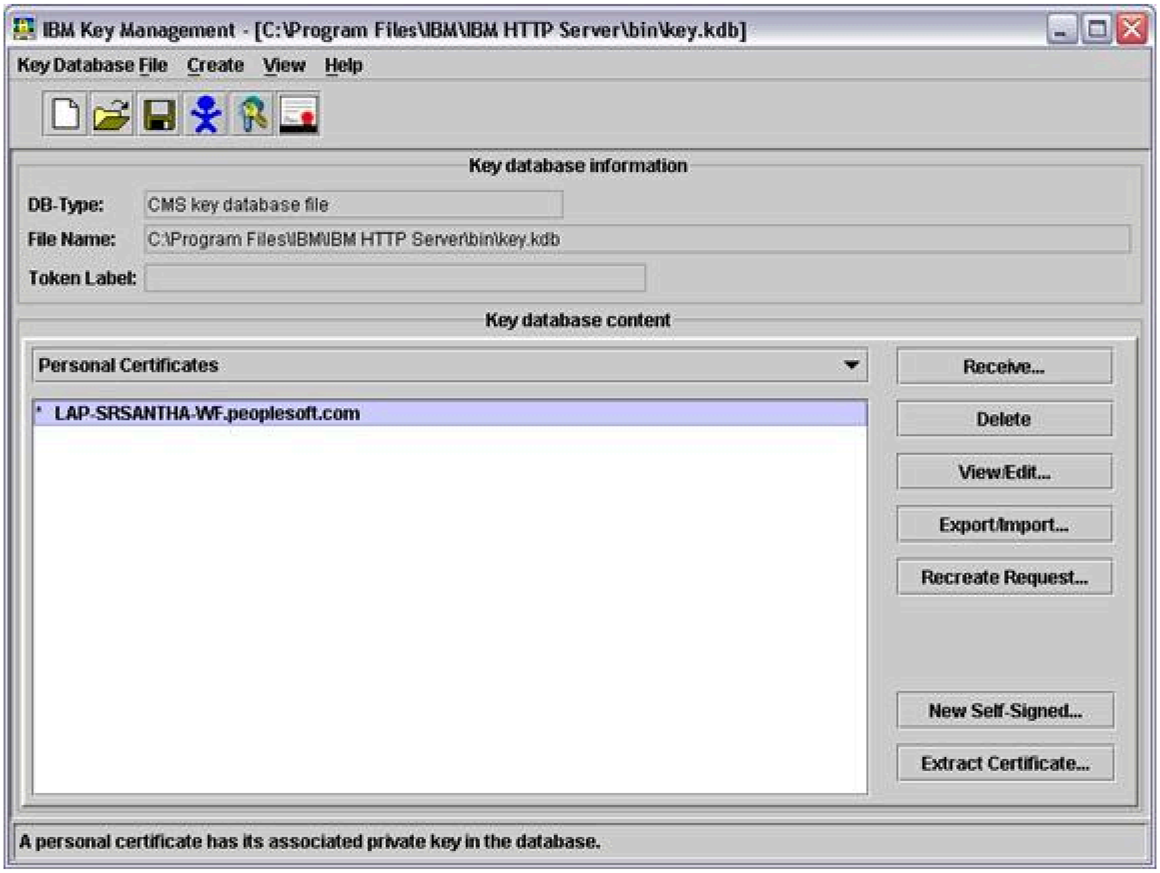
Token Label:

Signer Certificates

- Trial Intermediate CA
- Trial Root CA
- Entrust.net Global Secure Server Certification Authority
- Entrust.net Global Client Certification Authority
- Entrust.net Client Certification Authority
- Entrust.net Certification Authority (2048)
- Entrust.net Secure Server Certification Authority
- VeriSign Class 3 Public Primary Certification Authority
- VeriSign Class 2 Public Primary Certification Authority
- VeriSign Class 1 Public Primary Certification Authority
- VeriSign Class 4 Public Primary Certification Authority - G2
- VeriSign Class 3 Public Primary Certification Authority - G2
- VeriSign Class 2 Public Primary Certification Authority - G2
- VeriSign Class 1 Public Primary Certification Authority - G2
- VeriSign Class 4 Public Primary Certification Authority - G3
- VeriSign Class 3 Public Primary Certification Authority - G3

A signer certificate is from a certification authority (CA) or from another web site.

10. Add the Root CA and then Intermediate CA by clicking Add.



IBM Key Management - [C:\Program Files\IBM\IBM HTTP Server\bin\key.kdb]

Key database information

DB-Type: CMS key database file

File Name: C:\Program Files\IBM\IBM HTTP Server\bin\key.kdb

Token Label:

Key database content

Personal Certificates

* LAP-SRSANTHA-WF.peoplesoft.com

A personal certificate has its associated private key in the database.

11. Select the Personal Certificates option from Key database content. Add the certificate provided by CA by clicking the Receive option.
12. Save the file. A key database file with extension.kdb is created.
13. Go to the file <IBM_HTTP_SERVER_INSTALL_ROOT>\conf\httpd.conf and add the following to the VirtualHost:

```
LoadModule  ibm_ssl_module   modules/mod_ibm_ssl.so
Listen 443
<VirtualHost  DENOSCL244.mlab.jdedwards.com:443>
SSLEnable
SSLClientAuth none
</VirtualHost>
SSLDisable
Keyfile "C:\Program Files\IBM\IBM HTTP Server\bin\key.kdb"
```

Customize it according to your environment.

```
DENOSCL244.mlab.jdedwards.com -  DNS Name
Keyfile "C:\Program Files\IBM\IBM HTTP Server\bin\key.kdb" -Key Database
```

14. Go to plugin-cfg.xml under <WAS_INSTALL_ROOT>/Plugin/config/webserver1, where webserver1 is the webserver name.

    Add <Uri Name="/ListenerService/ ListenerService"/> under the node UriGroup.

    Add <VirtualHost Name="*:443"/> under node VirtualHostGroup
15. Go to plugin-cfg.xml under<WAS_INSTALL_ROOT>/profiles/default/cells/DENOSCL244Node01Cell/node/webserver1_node/servers/webserver1.

Add <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/ListenerService/ListenerService"/> under the node UriGroup.

Add <VirtualHost Name="*:443"/> under node VirtualHostGroup

16. Restart WAS.
17. Restart IBM HTTP Server.
18. Deploy the Listener service.

> **Note:**
>
> - *"Testing a Business Service That Consumes an External Web Service" in the JD Edwards EnterpriseOne Tools Business Services Development Guide* .

Enabling Secure File Transfer Protocol (SFTP) for Media Objects

Oracle recommends using SSH file transfer protocol, otherwise referred to as Secure FTP (SFTP), for media object access as a more secure alternative to FTP. When the Business Services Server is configured to use SFTP for media objects, the Media Object published business service can securely upload, download, and delete media objects.

When using SFTP, make sure that the SFTP user home folder is the same as the FTP user home folder for media object operations.

To enable SFTP, configure the following Business Services Server settings in Server Manager:

- **Use Secure FTP for MediaObject Fetch**. Click this check box to use SFTP to access media objects.
- **Timeout for SFTP connection**. The amount of time, in milliseconds, the Business Services Server will wait to make a secure SFTP connection. If you receive a connection timeout error when trying to connect to SFTP server, increase the timeout value.

For details about the configuration settings mentioned in this section, refer to the Server Manager internal help for each setting. For information on how to access the configuration settings in Server Manager, see the *JD Edwards EnterpriseOne Tools Server Manager Guide* .

> **Note:** If you are using Cygwin SFTP make sure you add below setting to the /etc/sshd_config file: `KexAlgorithms diffie-hellman-group1-sha1`

4 Configuring Business Services Server Security for JAX-WS Based Business Services

Understanding Business Services Server Security for JAX-WS Based Business Services on WAS

> **Note:** This chapter is applicable only if you use a WebSphere Application Server as your business services server.

JAX-WS based EnterpriseOne business services deployed to the WebSphere Application Server (WAS) are secure by default. They are only invoked by supplying valid EnterpriseOne credentials in the WS-Security part of the SOAP header. The Business Services Server uses the JD Edwards EnterpriseOne Login Module as the authentication mechanism for authenticating the credentials in the SOAP Header against the EnterpriseOne Security Server.

Prerequisite

Before you deploy a JAX-WS based business service application to a business services server on WAS, ensure that the business services server on WAS conforms to the EnterpriseOne Minimum Technical Requirements.

See document 745831.1 (JD Edwards EnterpriseOne Minimum Technical Requirements Reference on My Oracle Support.

See `https://support.oracle.com/rs?type=doc&id=745831.1`

> **Note:** JAX-WS based business services deployed to Oracle WebLogic Server are secure by default, and they use the same security model as JAX-RPC business services, as discussed in Chapter 3.

Securing JAX-WS Based Business Services on WAS

When the business services application is deployed to the business services server on WAS, Server Manager automatically installs and configures the following modules for all published services to ensure they are secure:

- The *wss.generate.issuedToken*, *wss.consume.issuedToken*, System Java Authentication and Authorization Service (JAAS) login configurations.
- The custom *E1JAXWSBSSV_UNT JAX-WS* policy set with WS-Security as the main policy to handle the UsernameToken element with user name and password elements in the SOAP Header.
- The custom *E1JAXWSBSSVBinding* JAX-WS binding to configure the generic issued token consumer for the inbound UsernameToken and to configure the caller.
- The custom Java Authentication and Authorization Service (JAAS) Application Login Module, *application.e1JAXWSBssvLogin*, to validate the JD Edwards EnterpriseOne users against the JD Edwards EnterpriseOne Security Server.

The system JAAS login module, the custom JAAS application login module, and the custom JAX-WS policy set are all installed once for a particular WAS profile. After a business service application is deployed to a business service instance, the custom JAX-WS policy set and binding are attached to the entire business service application making them applicable to all of the published services.

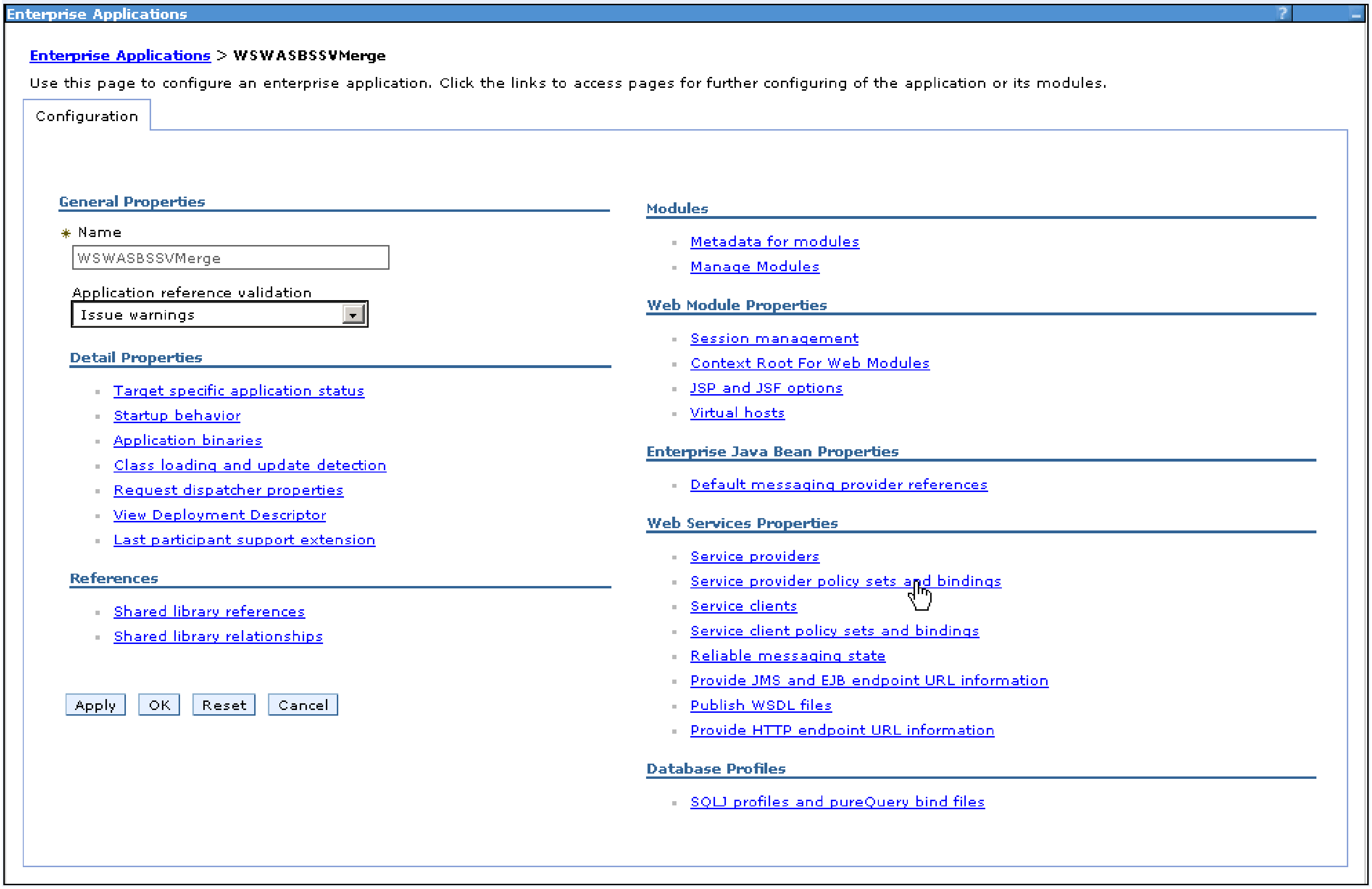
Configuring WebSphere to Use Anonymous Login

In WebSphere, you can disable security for the entire business services server application by detaching the custom JAX-WS policy set and binding. When you disable security, the system uses anonymous login credentials for authentication for all of the published services instead of the user credentials supplied in the WS-Security part of the SOAP Header. The anonymous login credentials are stored in the jdbj.ini file on the business services server.

To set up anonymous login for JAX-WS business services on WAS:

1. Login to the WAS Admin Console.
2. From the left-hand menu, click Applications > Application Types > WebSphere enterprise applications.
3. On the right-hand Enterprise Applications page, select the business services server application/instance for which you want to set up anonymous login.

4. On the Business Services Server Applications page, with the Configuration tab selected, click the Service provider policy sets and bindings link under the Web Services Properties heading.

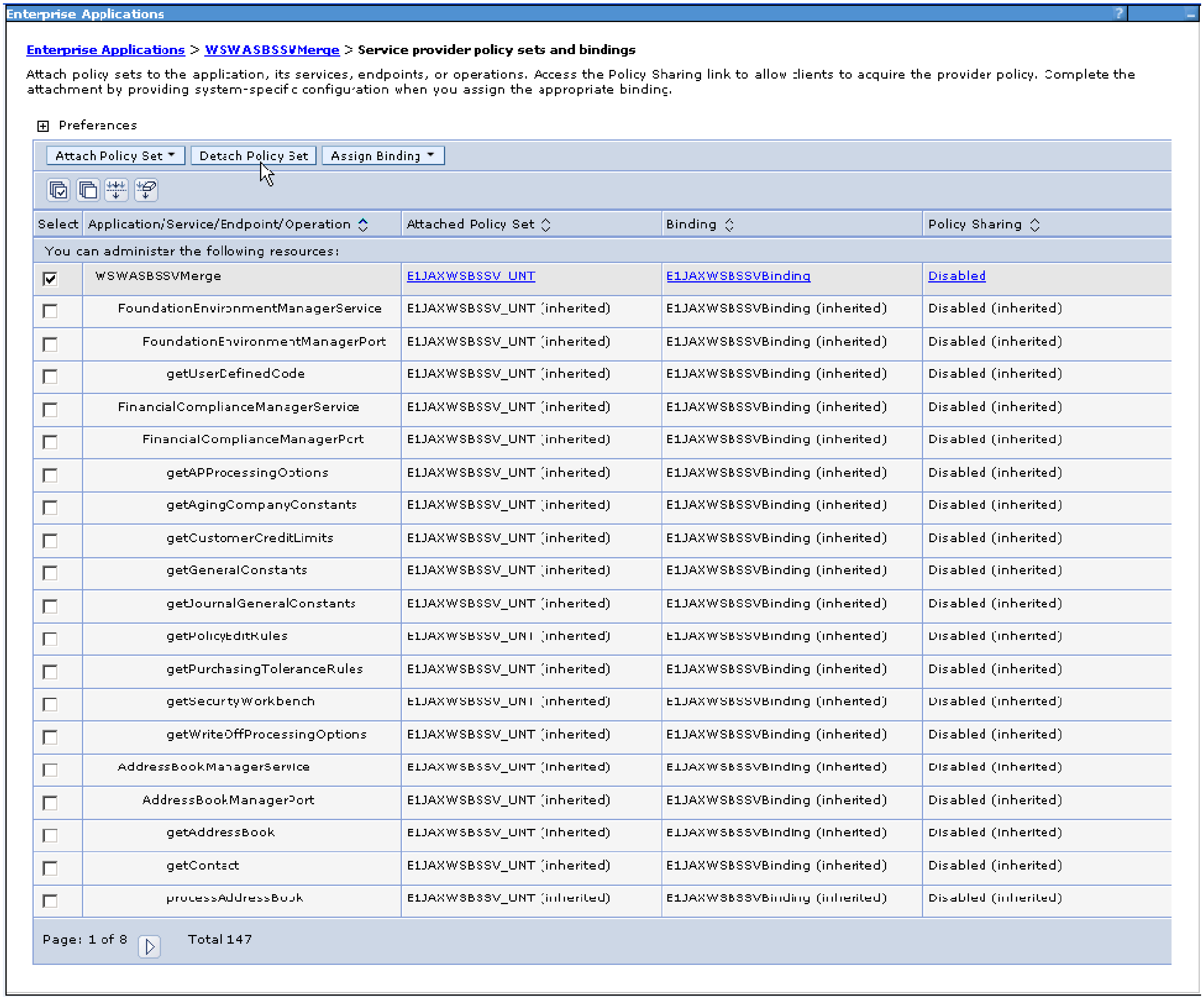

Enterprise Applications
Enterprise Applications > WSWASBSSVMerge

Use this page to configure an enterprise application. Click the links to access pages for further configuring of the application or its modules.

Configuration
General Properties
* Name
WSWASBSSVMerge
Application reference validation
Issue warnings
Detail Properties
Target specific application status
Startup behavior
Application binaries
Class loading and update detection
Request dispatcher properties
View Deployment Descriptor
Last participant support extension
References
Shared library references
Shared library relationships
Apply
OK
Reset
Cancel
Modules
Metadata for modules
Manage Modules
Web Module Properties
Session management
Context Root For Web Modules
JSP and JSF options
Virtual hosts
Enterprise Java Bean Properties
Default messaging provider references
Web Services Properties
Service providers
Service provider policy sets and bindings
Service clients
Service client policy sets and bindings
Reliable messaging state
Provide JMS and EJB endpoint URL information
Publish WSDL files
Provide HTTP endpoint URL information
Database Profiles
SQLJ profiles and pureQuery bind files

5. On the Service provider policy sets and bindings page, select the business services server application and click the Detach Policy Set button.

   This action detaches both the policy set and the binding for all of the published services in the business service application.



Enterprise Applications > WSWASBSSVMerge > **Service provider policy sets and bindings**

Attach policy sets to the application, its services, endpoints, or operations. Access the Policy Sharing link to allow clients to acquire the provider policy. Complete the attachment by providing system-specific configuration when you assign the appropriate binding.

Select	Application/Service/Endpoint/Operation	Attached Policy Set	Binding	Policy Sharing
You can administer the following resources:				
☑	WSWASBSSVMerge	E1JAXWSBSSV_UNT	E1JAXWSBSSVBinding	Disabled
□	FoundationEnvironmentManagerService	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	FoundationEnvironmentManagerPort	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	getUserDefinedCode	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	FinancialComplianceManagerService	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	FinancialComplianceManagerPort	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	getAPProcessingOptions	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	getAgingCompanyConstants	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	getCustomerCreditLimits	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	getGeneralConstants	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	getJournalGeneralConstants	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	getPolicyEditRules	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	getPurchasingToleranceRules	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	getSecurtyWorkbench	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	getWriteOffProcessingOptions	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	AddressBookManagerService	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	AddressBookManagerPort	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	getAddressBook	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	getContact	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)
□	processAddressBook	E1JAXWSBSSV_UNT (inherited)	E1JAXWSBSSVBinding (inherited)	Disabled (inherited)

Page: 1 of 8 ▷ Total 147

6. Save the changes.
7. Restart the business services server for changes to take effect.

After anonymous login is set up, you can invoke all of the published services in the business services application anonymously without passing user credentials in the WS-Security part of the SOAP Header.

5 Appendix A - Troubleshooting

Exceptions

When an error occurs an exception is thrown to the caller. These exceptions fall into different categories.

This section discusses:

- *Simple Object Access Protocol (SOAP) faults*
- *JD Edwards EnterpriseOne System Exceptions*
- *JD Edwards EnterpriseOne Application Exceptions/Errors*

Simple Object Access Protocol (SOAP) faults

This is most generic of the categories, all exceptions in this category are expressed as SOAP faults. For example, if something is incorrectly configured on the applications server or connectivity problems exist you will see these as SOAP faults.

JD Edwards EnterpriseOne System Exceptions

This category identifies errors inside the JD Edwards EnterpriseOne space. Most commonly these are server connectivity issues. For example, if a server is dropped, an exception is thrown to the caller to report this. The caller receives an exception if the authorization/authentication fails. Typically such exceptions are the result of a temporary system condition or are configuration issues.

Server Connectivity

It is important to be aware that inside the Business Services Server the feature used to contact the Enterprise Server is the Dynamic Java Connector. If the configuration settings for the Dynamic Java Connector are incorrect, it could cause connectivity problems with the Enterprise Server.

Authorization/Authentication

The system is secure by default, if you are getting authorization/authentication errors, most likely you are missing Security Workbench records.

JD Edwards EnterpriseOne Application Exceptions/Errors

Every error that happens within the application logic is reported as an exception to the caller. For example; an error in the business data setup indicates that incorrect data was passed to the business data or a necessary service property could be missing.

JD Edwards EnterpriseOne Application Warnings

Warnings from the JD Edwards EnterpriseOne application are not expressed as exceptions. Warnings are reported back to the caller in the response information.

Business Services Server Logs

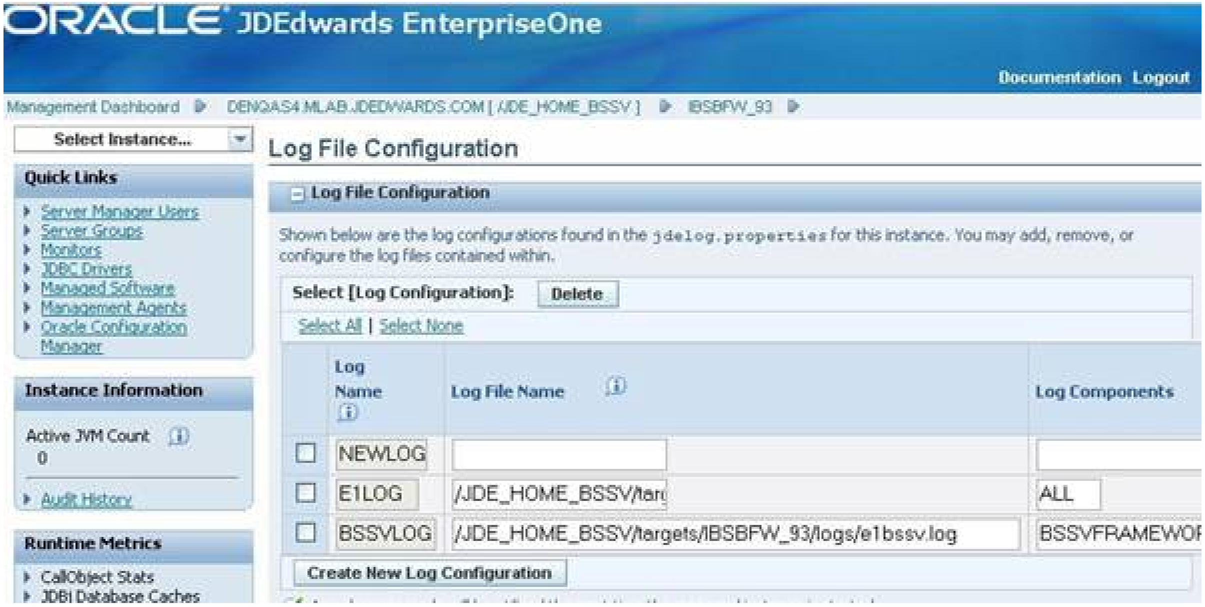
Business Services Server logs are configured and accessed through Server Manager.

> **Note:**
> - "View Log Files" in the *JD Edwards EnterpriseOne Tools Server Manager Guide*

A log component called BSSVFRAMEWORK is available for component level logging. When this is enabled, the log file displays system errors and enables you to gain insight to some application errors.

To enable component level logging:

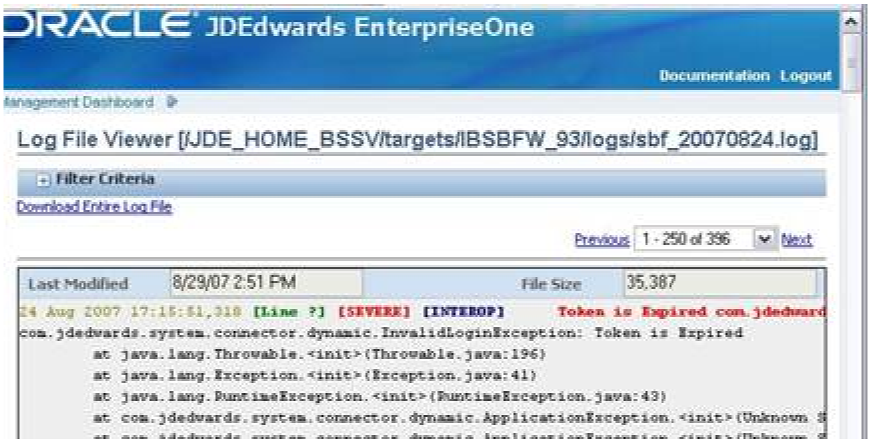
1. Access Server Manager.
2. Select a Business Services Server instance.
3. Select Log File Configuration.



Shown below are the log configurations found in the jdelog.properties for this instance. You may add, remove, or configure the log files contained within.

	Log Name	Log File Name	Log Components
□	NEWLOG		
□	E1LOG	/JDE_HOME_BSSV/tarç	ALL
□	BSSVLOG	/JDE_HOME_BSSV/targets/IBSBFW_93/logs/e1bssv.log	BSSVFRAMEWOI

4. On Log File Configuration, enter BSSVFRAMEWORK in thee Log Components column for the Business Services server log.
5. Click the Save button.



Log File Viewer [/JDE_HOME_BSSV/targets/IBSBFW_93/logs/sbf_20070824.log]

Last Modified 8/29/07 2:51 PM

File Size 35,387

```
24 Aug 2007 17:15:51,318 [Line ?] [SEVERE] [INTEROP]     Token is Expired com.jdedward
com.jdedwards.system.connector.dynamic.InvalidLoginException: Token is Expired
        at java.lang.Throwable.<init>(Throwable.java:196)
        at java.lang.Exception.<init>(Exception.java:41)
        at java.lang.RuntimeException.<init>(RuntimeException.java:43)
        at com.jdedwards.system.connector.dynamic.ApplicationException.<init>(Unknown S
        at com.jdedwards.system.connector.dynamic.ApplicationException.<init>(Unknown S
```