MFS Web Enablement Version 9.3.0 User’s Guide and Reference

Contents

Chapter 1. Overview of MFS Web enablement version 9.3 1
How MFS Web Enablement works 4
Prerequisites for MFS Web Enablement 6
Limitations of MFS Web Enablement Version 9.3.0 . . 7

Chapter 2. Configuration 9
Configuring WebSphere Application Server 9
Configuring the WebSphere Application Server resource adapter 9
Configuring SSL in WebSphere Application Server 10
Setting timeout in WebSphere Application Server 11
Setting timeout for an enterprise application . . 12
Configuring logging and tracing for WebSphere Application Server Version 5.1.1 13
Configuring logging and tracing for WebSphere Application Server Version 6.0 15
Configuring the OMVS environment 16
Configuring form-based authentication for MFS Web Enablement in WebSphere Application Server (optional). 16
Configuring a Web browser for MFS Web Enablement 20
Configuring a Microsoft Internet Explorer browser 20
Configuring a Mozilla Firefox Web browser . . 22

Chapter 3. MFS Servlet. 25
Installing the instance servlet Web Application Archive (WAR) file 26
Invoking the deployed instance servlet 28

Chapter 4. MFS Adapter 31

Chapter 5. Sample MFS style sheets . . 33

Chapter 6. Security considerations for MFS Web Enablement 37
Secure Sockets Layer support 37
WebSphere Application Server user authentication 38

Chapter 7. Sample instructions to Web enable the IMS IVP Phonebook application 39
Step 1: Parsing the MFS source file with MFS XML Utility 39
Step 2: Generating an instance servlet 40
Step 3: Generating an Application WAR file . . . 42
Step 4: Configure MFS Web Enablement support on WebSphere Application Server 42
Step 5: Deploy the application WAR file on WebSphere Application Server 43
Step 6: Invoke the instance servlet. 45
Step 7: Invoke PHONEBOOK 45
Step 8: Logout 46

Chapter 8. Troubleshooting 47
Messages and Codes for MFS Web Enablement . . 47
MFS Servlet messages 47
MFS Adapter messages 49
MFS Importer messages 50
Logging and tracing in MFS Web Enablement . . . 53

Chapter 9. Notices. 55

Chapter 10. Trademarks 57

Chapter 1. Overview of MFS Web enablement version 9.3

IMS™ MFS Web Enablement Version 9.3.0 provides the tooling utility and runtime support to Web-enable existing or new IMS MFS-based applications in IBM® WebSphere® Application Server, and interactively render them for display in standard browsers such as Microsoft® Internet Explorer and Mozilla Firefox.

MFS Web Enablement tooling utility support is comprised of the MFS XML Utility and the MFS Importer. The tooling utility support also requires WebSphere Application Server. The MFS XML Utility invokes the MFS Importer to parse MFS source files and generates XML Metadata Interchange (XMI) files for each MID and DIF (Message Input Descriptor or Device Input Format) pair, MOD and DOF (Message Output Descriptor or Device Output Descriptor) pair, and MFS table. For more information about the MFS XML Utility, see the *MFS XML Utility User's Guide*.

MFS Web Enablement runtime support is comprised of application instance servlets, sample cascading style sheets, the MFS Servlet, and the MFS Adapter. MFS Web Enablement runtime support requires IMS Connect, IMS Connector for Java™, and WebSphere Application Server.

MFS XML Utility

The MFS XML Utility is a command line development-time tool that runs on a Microsoft DOS command prompt. This utility generates all of the necessary files needed to web enable your MFS-based IMS applications. It takes MFS source files as input and produces metadata XMI files and Web application archive (WAR) files as output. In addition, the utility provides FTP client support to transport the generated output to WebSphere Application Server.

For more information, see the *MFS XML Utility User's Guide and Reference*.

IMS MFS Importer

The IMS MFS Importer reads and parses MFS source files for an application and generates XMI instance files that describe the MFS-based application interface. The XMI file represents all the application interface information encapsulated by the MFS source, including the input and output device descriptors, message descriptors, MID-MOD relationship, device characteristics, and operation semantics. To ensure non-proprietary access, the MFS Importer is built using the MFS metamodel, which is part of the Object Management Group (OMG) Enterprise Application Integration or Common Application Metamodel (CAM) standard. The MFS Importer takes MFS source files as input. Users can recreate valid MFS source from MFS format library by using the IBM IMS Message Format Services Reversal Utility.

For more information, see the *MFS XML Utility Guide and Reference*.

IBM WebSphere Application Server (for Windows® or AIX®, and z/OS®)

IBM WebSphere Application Server is a Java-based application platform, Enterprise Edition (J2EE), and Web services technology-based application server that integrates enterprise data and transactions.

For more information, see "Configuring WebSphere Application Server" on page 9 and the WebSphere Application Server Information Center at www.ibm.com/software/webservers/appserv/was/library/index.html.

MFS instance servlet

Instance servlets are generated by the MFS XML Utility tool. An instance servlet supplies certain limited IMS Connector for Java connection properties, an MFS style sheet filepath, and an MFS XMI repository filepath to the MFS Servlet.

For more information, see Chapter 3, "MFS Servlet," on page 25.

MFS Servlet

The MFS Servlet is the super class of all MFS instance servlets. The MFS Servlet runs on

WebSphere Application Server and handles HTTP requests and responses to and from client browsers. The MFS Servlet is responsible for the connection state management, interaction with the MFS Adapter, and the rendering of MFS XMI objects using the style sheet to dynamically produce MFS Web pages.

For more information, see Chapter 3, “MFS Servlet,” on page 25.

IMS MFS Adapter

The MFS Adapter loads metadata XMI files for data transformation to and from byte stream that IMS applications understand. It runs inside WebSphere Application Server to work in conjunction with the MFS Servlet and IMS Connector for Java to supply input and to handle output for MFS-based IMS transactions.

For more information, see Chapter 4, “MFS Adapter,” on page 31.

IMS Connect

IMS Connect provides high-performance communications for IMS between one or more TCP/IP or local OS/390® clients and one or more IMS systems. IMS Connect enables TCP/IP or local OS/390 clients to exchange messages through the IMS Open Transaction Manager Access (OTMA) facility for accessing IMS transactions. IMS Connect provides communication links between TCP/IP clients, such as IMS Connector for Java and IMS.

For more information, see the *IMS Connect for z/OS* documentation at http://www.ibm.com/software/data/db2imstools/imstools-library.html.

IMS Connector for Java

IMS Connector for Java provides a way to create Java applications that can access IMS transactions through IMS Connect. IMS Connector for Java builds an OTMA message and sends it to IMS Connect, which in turn sends it to OTMA using XCF. OTMA sends the IMS transaction input message to IMS and receives the IMS transaction output message from IMS. The IMS transaction output message is sent back to IMS Connect by OTMA. IMS Connect then sends the Java application to the client using IMS Connector for Java.

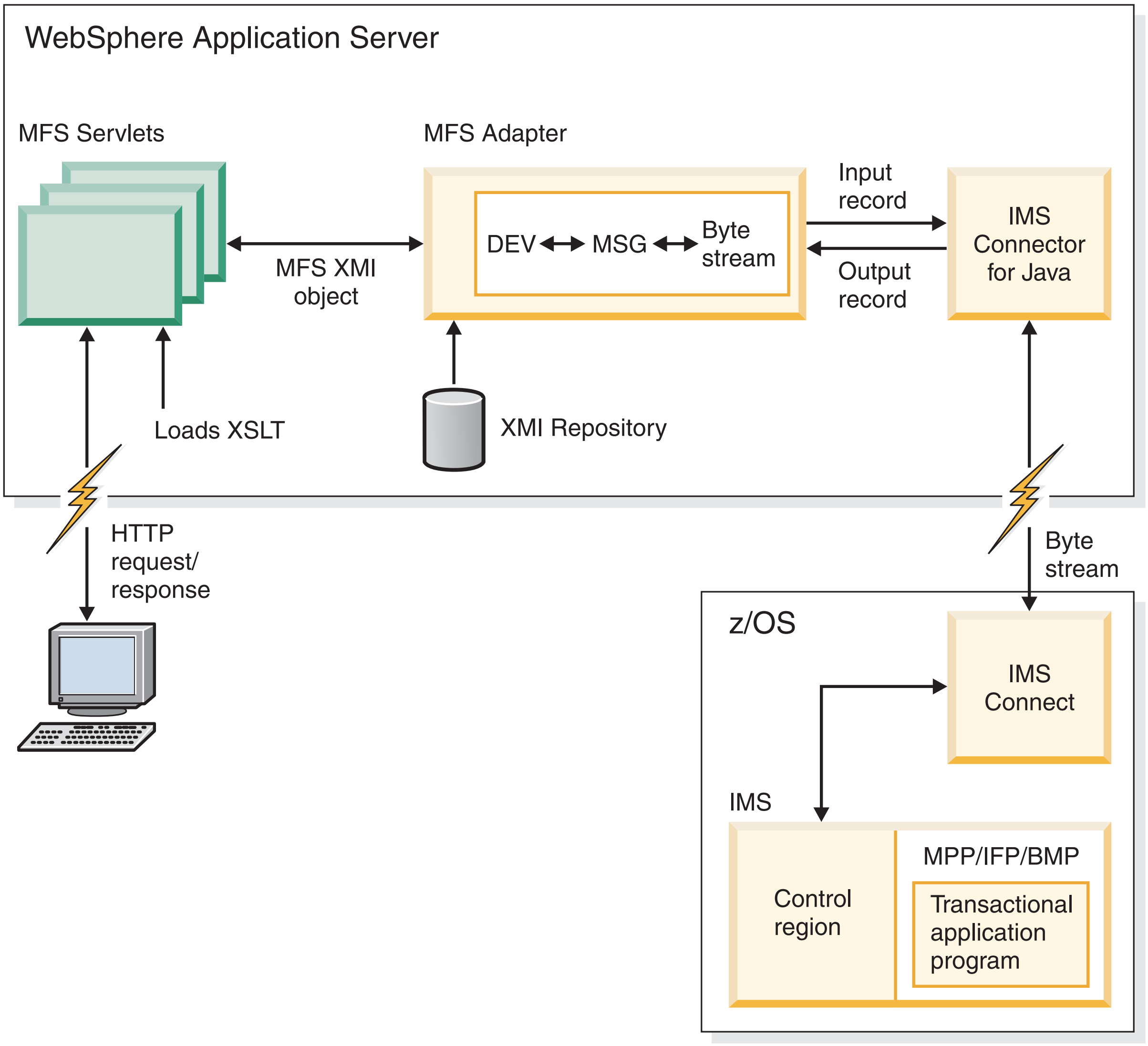
For more information, see the *IMS Connector for Java* documentation at http://www.ibm.com/software/data/db2imstools/imstools-library.html.

XSL Transformation in WebSphere Application Server

WebSphere Application Server provides a Xalan Extensible Stylesheet Language Transformation (XSLT) processor for converting XML data into HTML. The transformation is done by applying an XSL cascading style sheet, which is a well-formed XML file that contains template information.

For more information about XSLT, see http://java.sun.com/j2ee/1.4/docs/tutorial/doc/JAXPXSLT.html.

The figure below shows how MFS Web Enablement works.



The following features of core MFS are supported:

- 3270 type devices
- Attribute bytes
- Cursor positioning
- Extended attributes bytes (blinking only supported in Mozilla)
- Multiple physical pages input
- Multiple logical and physical pages output
- Message options 1 and 2 only for input and output
- PA1 key equivalent to advance to the next physical page
- PF keys with literal data (transaction code and two commands: /FOR and /EXIT) and two control functions: NEXTPP (next physical page) and ENDMPPI (end multiple physical pages input)
- System literals only for date, time, and LPAGENO
- System default MIDs and MODs, including DFSMI1, DFSMI2, DFSMO1, DFSMO2, DFSMO3, DFSMO5, and the blank screen

Other functional characteristics of MFS Web Enablement include:

Conversation Support

The host connection is created and managed by the MFS Servlet for the duration of the user session. The connection object is reused in conversations from the same user session. MFS Web Enablement runtime support handles /EXIT command requests to properly terminate the conversation on the host.

Instance servlet Web Application Archive (WAR)

The generated WAR file is deployable to WebSphere Application Server. Each WAR file contains a deployment descriptor file and one or more instance servlets, which contain specific host connectivity, specific IMS Connector for Java connection properties, XMI repository location, style sheet location, and IMS Connector for Java interactionSpec parameters such as execution timeout and socket timeout.

Style Sheet

The two sample MFS style sheets provided render the XML data stream into HTML for display in a browser. One style sheet displays a 3270 type terminal and the other style sheet displays a stylized 3270 type terminal.

For more information, see Chapter 5, “Sample MFS style sheets,” on page 33.

SSL/HTTPS

Secure Sockets Layer is provided by WebSphere Application Server configurations to encrypt the data transmitted between user browsers and the Web server.

More information about SSL/HTTPS is available in “Configuring SSL in WebSphere Application Server” on page 10.

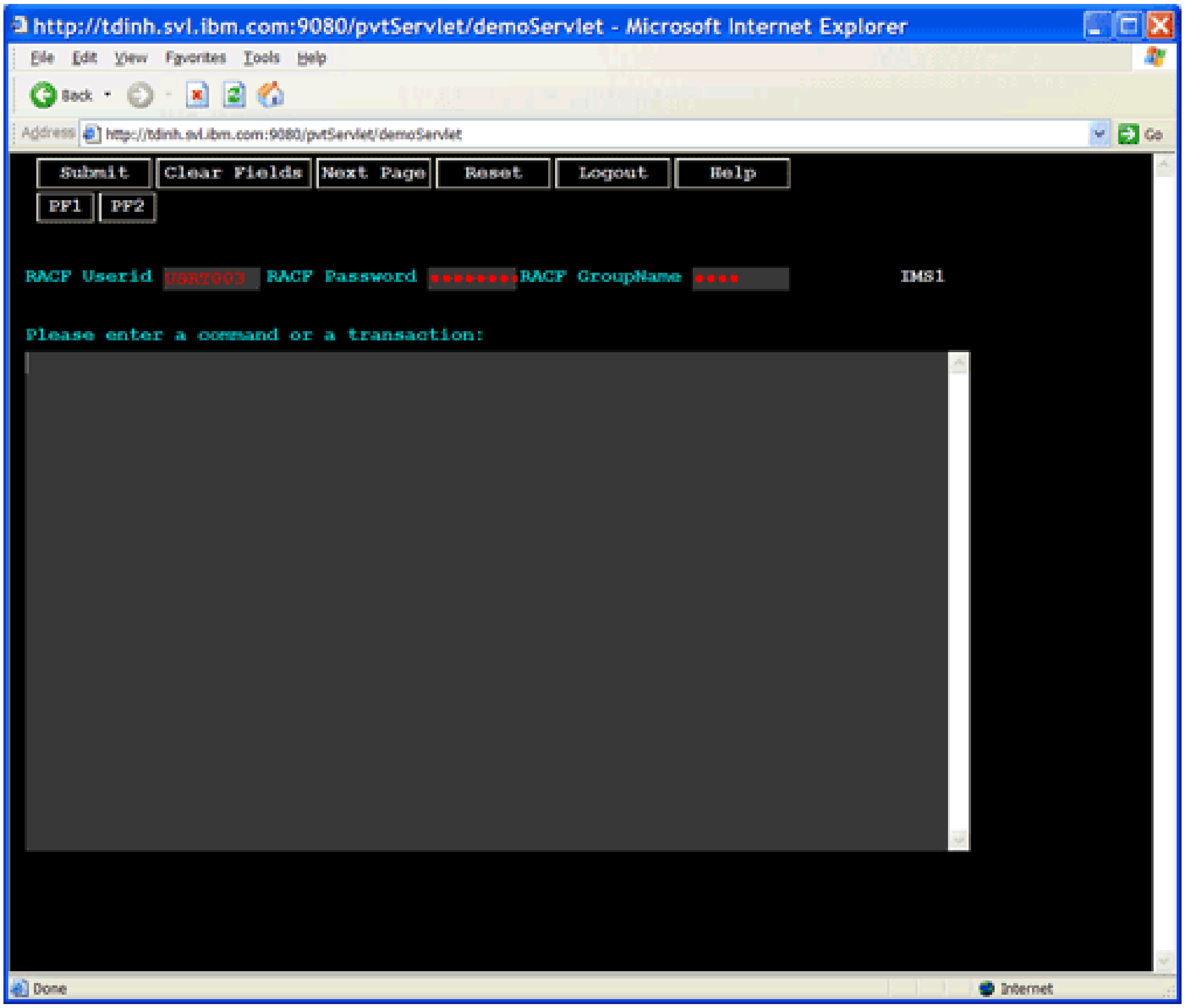
WebSphere Application Server user authentication

WebSphere Application Server user authentication can be optionally configured so that clients accessing a particular servlet for the first time must log in. See “WebSphere Application Server user authentication” on page 38 for more information about the sample login mechanism.

How MFS Web Enablement works

The MFS servlet receives an HTTP request, for example http://servername/context root/servlet name, and loads the session objects into memory. If this request is the initial request, the servlet creates a new session and sends out the initial blank page, which is a representation of the 3270 type terminal blank screen, for display. Similar to using the Clear key on a 3270 terminal, you can also request the initial blank screen at any time by pressing the **Reset** button.

Here is what the screen looks like:


http://tdinh.svl.ibm.com:9080/pvtServlet/demoServlet - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://tdinh.svl.ibm.com:9080/pvtServlet/demoServlet
Go
Submit
Clear Fields
Next Page
Reset
Logout
Help
PF1
PF2
Done
Internet

RACF Userid USRT003 RACF Password ●●●●●●●● RACF GroupName ●●●● IMS1

Please enter a command or a transaction:

From the above screen, you can enter the following inputs:

- RACF® user ID, password, and group name
- /FOR or /FORMAT *modname* command
- Transaction code followed by data on the blank page
- /EXIT command

RACF userid, password, and group name

The RACF information that is specified per instance servlet in the MFS XML Utility is displayed on the blank screen as the default. You can choose to supply a different set of credentials. The supplied credentials are valid for the entire session. The session is terminated after you log out, close the browser, or the session times out. New credentials are used to create a new IMS Connector for Java connection object and the previously active connection and pending conversation are automatically terminated.

The RACF userid, password, and group name, will be converted to uppercase text by the MFS Adapter.

/FOR or /FORMAT *modname* command

The format (/FOR or /FORMAT) command is processed by the MFS Adapter, which attempts to load the MOD /DOF XMI file, based on the *modname*, from the XMI repository. The MFS Servlet then renders the DOF metadata with the MFS style sheet and returns the formatted HTML page to the client browser. If the specified *modname* cannot be found, the system returns the message `IXFT003E: REQUESTED XMI NOT FOUND: MODNAME` using system default DFSMO3.xmi.

Transaction code followed by data on the blank page

The transaction code and optional data are written to the input byte array and are then sent to IMS. Trancode is converted to uppercase text by the MFS Adapter. The data remains unchanged (mixed cased allowed). Output execution follows the same flow as in processing execution in "Transaction data on a formatted page."

/EXIT command

The exit (/EXIT) command is processed by the MFS Servlet and the MFS Adapter to end the current pending conversation. The MFS Servlet determines if the client is in the middle of a conversation. To end a conversational message, the MFS Adapter sets the SYNC_END_CONVERSATION in the IMSInteractionSpec, and then sends an empty request through IMS Connector for Java to terminate the host conversation. The system returns one of the following messages:

- If in a conversation: `DFS058I HH:MM:SS EXIT COMMAND COMPLETED`
- If not in a conversation: `DFS180 HH:MM:SS NO ACTIVE CONVERSATION IN PROCESS, CANNOT PROCESS COMMAND`

From a formatted page, you can enter transaction data.

Transaction data on a formatted page

Data input on a formatted page is processed by the MFS Servlet and the MFS Adapter using the corresponding MID /DIF XMI files. The MFS Adapter transforms the device data into a byte array and uses IMS Connector for Java to send it to the specified host. In the connection interaction specification, the MFS Adapter sets the imsRequestType to 3 (the MFS type) and sets the mapName to the next message of the current MID or to the `DFSMO2` XMI file when the next message is unspecified.

For a successful execution, the output byte array is transformed by the MFS Adapter using the corresponding MOD /DOF XMI file associated with the mapName specified in the IMSInteractionSpec. The MFS Servlet then renders the XML object with a style sheet and returns the generated HTML page to the client browser. If the specified XMI file cannot be found, the system returns the message `IXFT003E: REQUESTED XMI NOT FOUND: MODNAME` using the system default `DFSMO3.xmi`.

If the specified mapName is MFS Bypass, for example DFS™.EDTN or DFS.EDT, the system returns the message `IXFT003E: Unsupported MFS bypass: mapName` using the system default `DFSMO2.xmi`.

For an unsuccessful execution, if the data returned starts with the prefix "DFS," the system returns the output data using the system default (single segment data) DFSMO1.xmi or (multiple segments data) DFSMO5.xmi. Otherwise, for all other errors that occur during runtime processing, the system returns an error message using system default DFSMO2.xmi. The error message should contain a valid code for further explanation.

The MFS Servlet then renders the DOF metadata with the MFS style sheet and returns the formatted HTML page to the client browser.

Prerequisites for MFS Web Enablement

Platforms that support IMS MFS Web Enablement Version 9.3 include WebSphere Application Server for Microsoft Windows, AIX, and z/OS. The target runtime server is WebSphere Application Server (for Windows, AIX, and z/OS) Version 5.1.1 and Version 6.0 with the latest fix packs installed.

You must have the following products and tools installed to use MFS Web Enablement Version 9.3.0:

- IMS Version 9.1 or later
- IMS Connect Version 9.1.0.*x*

- IBM WebSphere Application Server distributed platforms (Windows and AIX only) Version 5.1 or 6.0, or IBM WebSphere Application Server z/OS Version 6.0 with the latest fix packs applied. Version 5.1.1 Websphere Application Server requires both the Cumulative fix pack and the Cumulative fix pack for SDKs.
- IMS Connector for Java Version 9.1.0.*x*
- One of the following Web browsers:
  - Microsoft Internet Explorer Version 6 or later
  - Mozilla Firefox Version 1.0.1 or later

Limitations of MFS Web Enablement Version 9.3.0

MFS Web Enablement does not support the following features of MFS:

- The **Back** button and **Refresh** button on the browser are not supported.
- Device field literals other than system literals
- Double Byte Character Set (DBCS)
- Extended Graphic Character Set (EGCS)
- IMS system generated MFS parameters, such as PAGDEL
- Magnetic Strip reading device
- Message option 3
- MFS Bypass
- MFS Buffer Pool
- MFS Field Exit routine
- MFS Pool Manager
- MFS Segment Exit routine
- Operator Control Table
- Operator Logical Paging
- PA2 key to advance to the next message
- PA3 key (Copy to the local printer)
- PF keys, except literals, next page command, and end multiple physical pages input command
- Commands, except /FOR and /EXIT
- Password
- Printer devices
- Programmed Symbols, such as scientific or technical symbols
- Selector Pen
- System Control Area (SCA)
- System literal defined for output sequence number, logical terminal name, and the queue number of the message waiting
- $$IMSDIR (Resident directory)
- Asynchronous send-only message requests
- Commit Mode 0
- SyncLevel Confirm
- Confirm with purgable
- Confirm with purgable not deliverable
- Resume Tpipe
- LTerm messages
- Transactional Level Security

- SSL between IC4J and IMS Connect

Chapter 2. Configuration

You must configure WebSphere Application Server and your Web browser so that they are optimized for MFS Web Enablement.

The following topics provide additional information:

- "Configuring WebSphere Application Server"
- "Configuring a Web browser for MFS Web Enablement" on page 20

Configuring WebSphere Application Server

You must configure WebSphere Application Server to optimize it for use with MFS Web Enablement.

The following topics provide additional information:

- "Configuring the WebSphere Application Server resource adapter"
- "Configuring SSL in WebSphere Application Server" on page 10
- "Setting timeout in WebSphere Application Server" on page 11
- "Setting timeout for an enterprise application" on page 12
- "Configuring logging and tracing for WebSphere Application Server Version 5.1.1" on page 13
- "Configuring form-based authentication for MFS Web Enablement in WebSphere Application Server (optional)" on page 16

Configuring the WebSphere Application Server resource adapter

This topic describes how to configure WebSphere Application Server resource adapter.

Prerequisite: To begin configuring the WebSphere Application Server resource adapter, you must complete the following tasks:

- Download and install IMS Connector for Java Version from http://www.ibm.com/software/data/db2imstools/imstools/imsjavcon.html.
- Download the MFS Web Enablement ZIP files from http://www.ibm.com/software/data/ims/toolkit/mfswebsupport/index.html. Unzip and place the JAR files in an accessible location on WebSphere Application Server.

  Note: If you FTP the JAR files to the WebSphere Application Server system, make sure it is first set to binary mode.

To configure the WebSphere Application Server resource adapter:

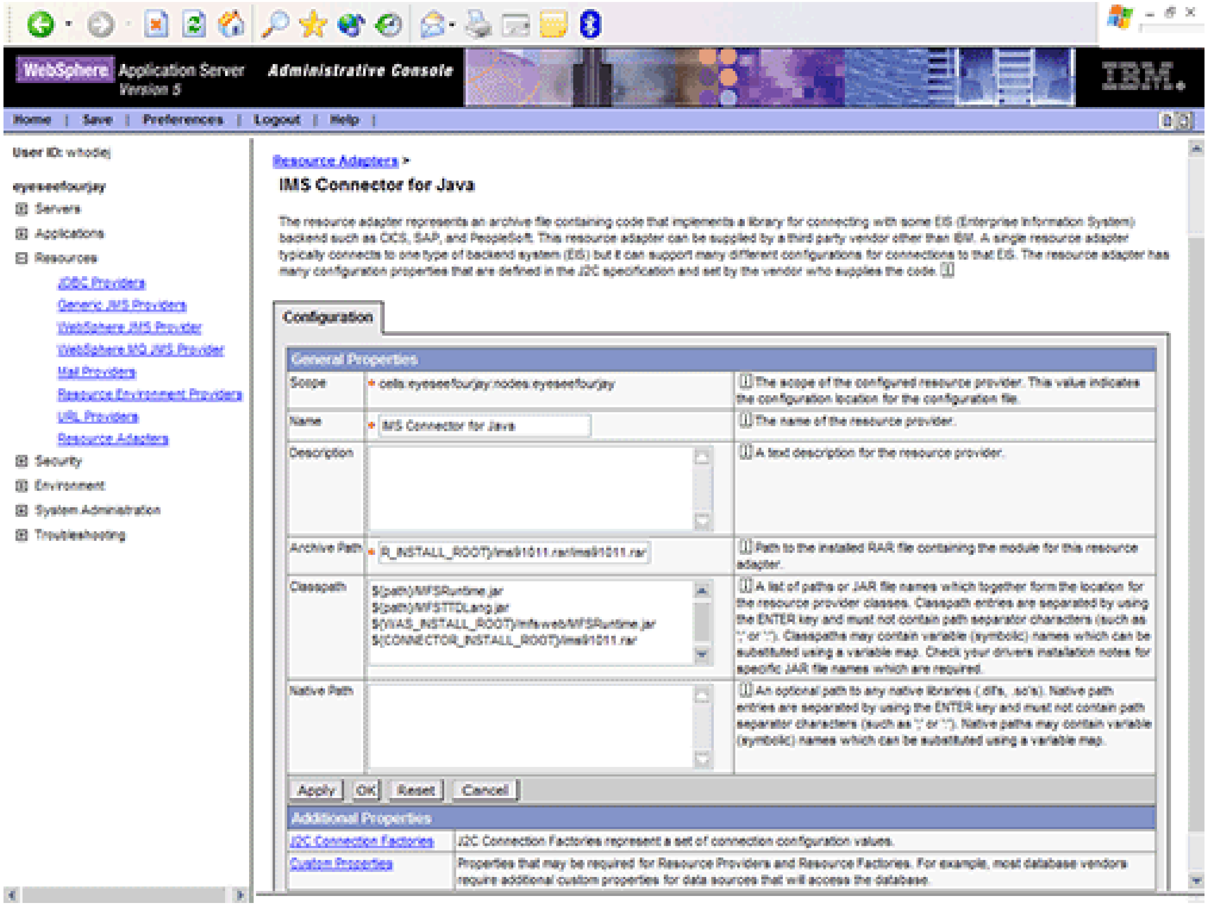
1. Start WebSphere Application Server and then open the Administrative console.
2. From the contents pane, expand **Resources** and then click **Resource Adapters**.
3. Create a directory named "mfsweb" under your WAS_INSTALL_ROOT directory. Copy and paste the MFSRuntime.jar and MFSTDTDLang.jar into this new folder.

   Note: You can look up WAS_INSTALL_ROOT" directory from **Environment**->**WebSphere Variables** on WebSphere Application Server Administrative Console.
4. Select the IMS resource adapter (IMS Connector for Java). Note that the current class path shows the following path: `$(CONNECTOR_INSTALL_ROOT)/ims91011.rar`.

Important: Add the MFS JAR files **before** the IMS resource adapter entry. The order of the class path, if incorrect wil cause the servlet to fail during initialization. Each class path entry must be on a new line.

```
${WAS_INSTALL_ROOT}/mfsweb/MFSRuntime.jar
${WAS_INSTALL_ROOT}/mfsweb/MFSTDTDLang.jar
```

The following figure shows the MFS JARS files located in the `WAS_INSTALL_ROOT/mfsweb` directory, where `WAS_INSTALL_ROOT` is the root install directory of WebSphere Application Server.



IMS Connector for Java

The resource adapter represents an archive file containing code that implements a library for connecting with some EIS (Enterprise Information System) backend such as CICS, SAP, and PeopleSoft. This resource adapter can be supplied by a third party vendor other than IBM. A single resource adapter typically connects to one type of backend system (EIS) but it can support many different configurations for connections to that EIS. The resource adapter has many configuration properties that are defined in the J2C specification and set by the vendor who supplies the code.

General Properties		
Scope	* cells:eyeseefourjay:nodes:eyeseefourjay	The scope of the configured resource provider. This value indicates the configuration location for the configuration file.
Name	* IMS Connector for Java	The name of the resource provider.
Description		A text description for the resource provider.
Archive Path	* R_INSTALL_ROOT}/ims91011.rar/ims91011.rar	Path to the installed RAR file containing the module for this resource adapter.
Classpath	${path}/MFSRuntime.jar ${path}/MFSTTDLang.jar ${WAS_INSTALL_ROOT}/mfsweb/MFSRuntime.jar ${CONNECTOR_INSTALL_ROOT}/ims91011.rar	A list of paths or JAR file names which together form the location for the resource provider classes. Classpath entries are separated by using the ENTER key and must not contain path separator characters (such as ';' or ':'). Classpaths may contain variable (symbolic) names which can be substituted using a variable map. Check your drivers installation notes for specific JAR file names which are required.
Native Path		An optional path to any native libraries (.dll's, .so's). Native path entries are separated by using the ENTER key and must not contain path separator characters (such as ';' or ':'). Native paths may contain variable (symbolic) names which can be substituted using a variable map.

Additional Properties	
J2C Connection Factories	J2C Connection Factories represent a set of connection configuration values.
Custom Properties	Properties that may be required for Resource Providers and Resource Factories. For example, most database vendors require additional custom properties for data sources that will access the database.

Figure 1. MFS JAR files in the WAS_INSTALL_ROOT/mfsweb directory

5. Click **Apply**.
6. Click **Save** inside the Messages box.
7. Click **Save**.
8. Restart WebSphere Application Server.

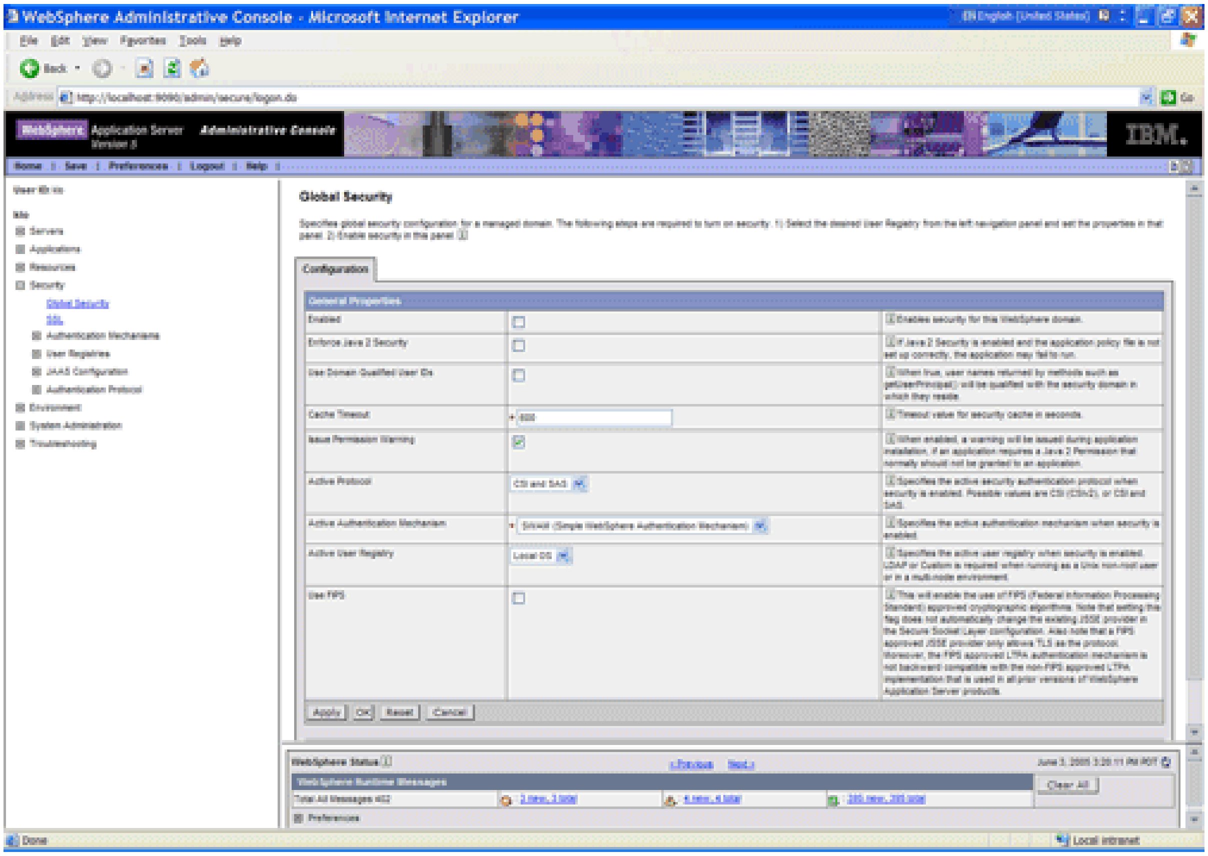
Configuring SSL in WebSphere Application Server

SSL enablement is optional. If you choose to enable SSL, you will need to have SSL enabled in both your Web browser and in WebSphere Application Server. For more information about SSL enablement in your Web browser, see “Enabling SSL for Microsoft Internet Explorer (optional)” on page 21 or “Enabling SSL for Mozilla Firefox (optional)” on page 23.

To configure SSL for WebSphere Application Server, you must configure HTTPS, which securely handles the HTTP transport or a web container.

To enable WebSphere Application Server SSL:

1. Start WebSphere Application Server, open the Administrative Console, expand **Security** from the left-hand side, and then click **Global Security**.
2. Under General Properties, select the **Enable** check box and, de-select the **Enforce Java 2 Security** check box and accept all other default values:



Global Security

Specifies global security configuration for a managed domain. The following steps are required to turn on security: 1) Select the desired User Registry from the left navigation panel and set the properties in that panel. 2) Enable security in this panel.

Configuration

General Properties		
Enabled	□	Enables security for this WebSphere domain.
Enforce Java 2 Security	□	If Java 2 Security is enabled and the application policy file is not set up correctly, the application may fail to run.
Use Domain Qualified User IDs	□	When true, user names returned by methods such as getUserPrincipal() will be qualified with the security domain in which they reside.
Cache Timeout	* 600	Timeout value for security cache in seconds.
Issue Permission Warning	☑	When enabled, a warning will be issued during application installation, if an application requires a Java 2 Permission that normally should not be granted to an application.
Active Protocol	CSI and SAS	Specifies the active security authentication protocol when security is enabled. Possible values are CSI (CSIv2), or CSI and SAS.
Active Authentication Mechanism	* SWAM (Simple WebSphere Authentication Mechanism)	Specifies the active authentication mechanism when security is enabled.
Active User Registry	Local OS	Specifies the active user registry when security is enabled. LDAP or Custom is required when running as a Unix non-root user or in a multi-node environment.
Use FIPS	□	This will enable the use of FIPS (Federal Information Processing Standard) approved cryptographic algorithms. Note that setting the flag does not automatically change the existing JSSE provider in the Secure Socket Layer configuration. Also note that a FIPS approved JSSE provider only allows TLS as the protocol. Moreover, the FIPS approved LTPA authentication mechanism is not backward compatible with the non-FIPS approved LTPA implementation that is used in all prior versions of WebSphere Application Server products.

Figure 2. Configuring SSL in the WebSphere Application Server Administrative Console

Note:
The screen above has "https" is in the Address bar of the browser indicating SSL is enabled.

3. Click **Apply**.
4. Click **Save** in the Messages box.
5. Click **Save**.
6. Restart WebSphere Application Server.

For more information about how to configure SSL in WebSphere Application Server, see the following Web pages:

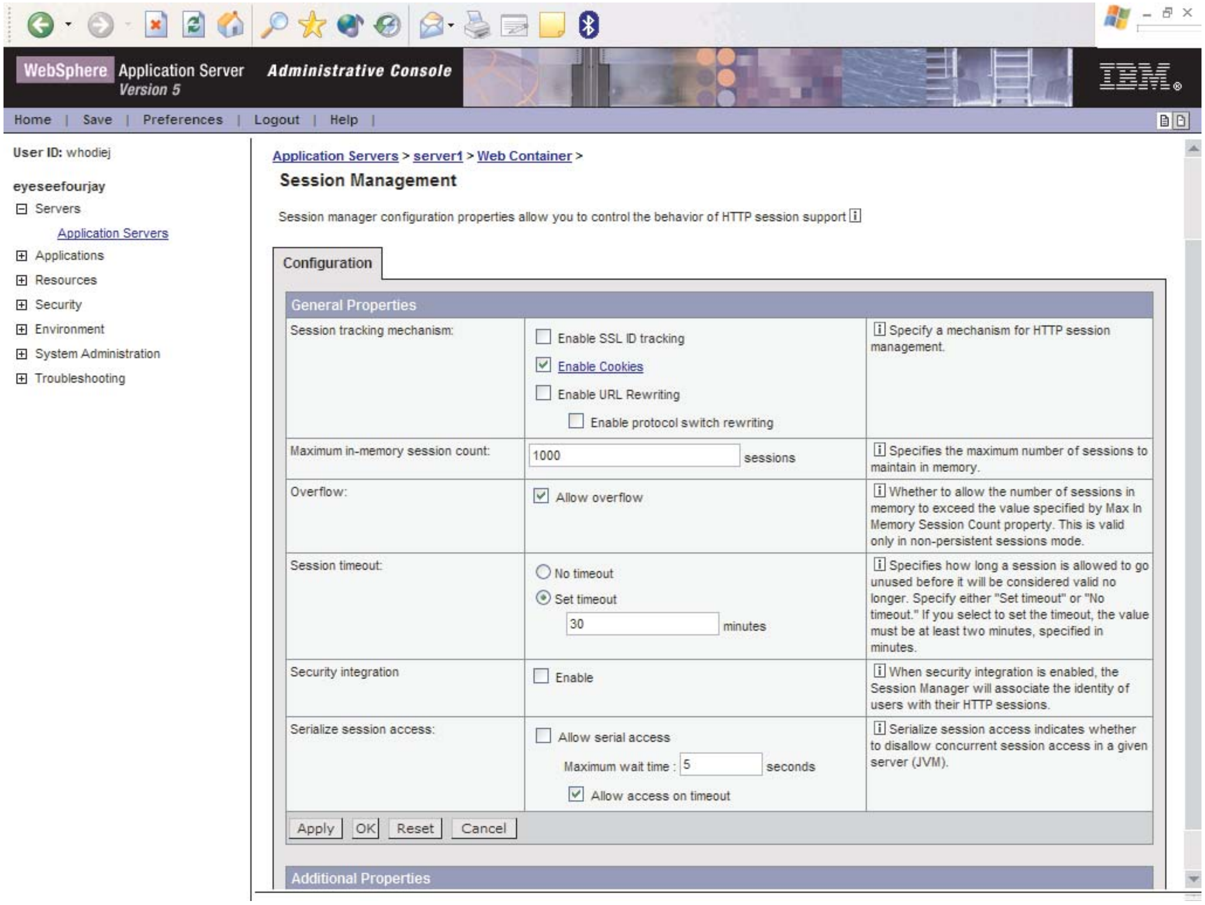
- http://www.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter/was/06061801a07.html
- http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/zseries/ae/uwbs_transssl.html

Setting timeout in WebSphere Application Server

To set session timeout in WebSphere Application Server:

1. Go to the Administrative Console, expand **Servers** from the left-hand side, and then click **Application servers**.

2. Select the server you want by clicking on the server name hyperlink (for example: `server1`). The application server configuration window appears.
3. From the **server1** window, click **Web Container**.
4. From the **Web Container** window, click **Session Management**.
5. In the **Session Management** window, under **General Properties** -> **Session timeout**, choose either **No timeout** or **Set timeout**. If you choose **Set timeout**, enter the number of minutes that your session can run before it times out:



Application Servers > server1 > Web Container >

Session Management

Session manager configuration properties allow you to control the behavior of HTTP session support

General Properties

Session tracking mechanism:	□ Enable SSL ID tracking ☑ Enable Cookies □ Enable URL Rewriting □ Enable protocol switch rewriting	Specify a mechanism for HTTP session management.
Maximum in-memory session count:	1000 sessions	Specifies the maximum number of sessions to maintain in memory.
Overflow:	☑ Allow overflow	Whether to allow the number of sessions in memory to exceed the value specified by Max In Memory Session Count property. This is valid only in non-persistent sessions mode.
Session timeout:	○ No timeout ● Set timeout 30 minutes	Specifies how long a session is allowed to go unused before it will be considered valid no longer. Specify either "Set timeout" or "No timeout." If you select to set the timeout, the value must be at least two minutes, specified in minutes.
Security integration	□ Enable	When security integration is enabled, the Session Manager will associate the identity of users with their HTTP sessions.
Serialize session access:	□ Allow serial access Maximum wait time : 5 seconds ☑ Allow access on timeout	Serialize session access indicates whether to disallow concurrent session access in a given server (JVM).

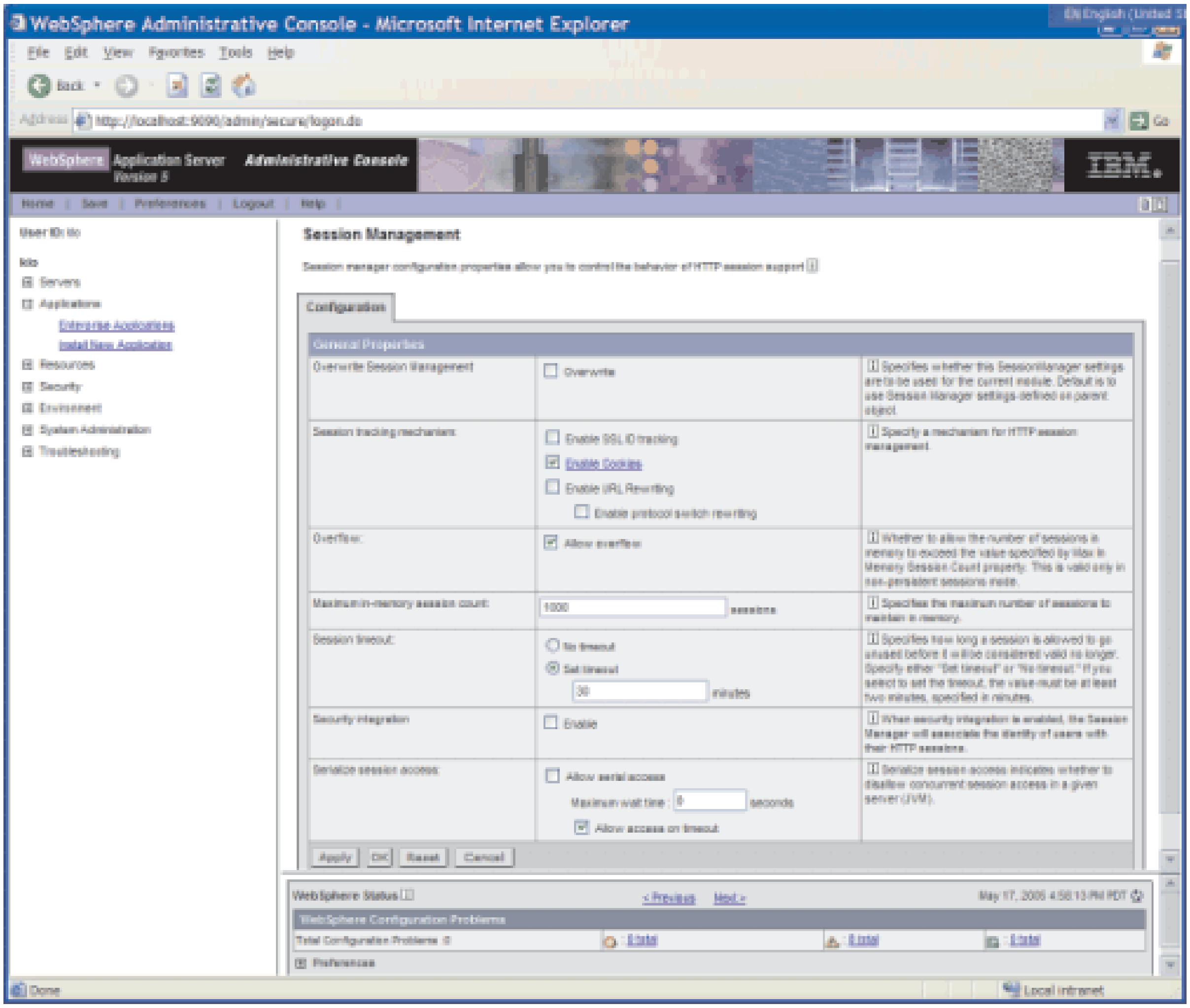
Figure 3. Setting timeout in the WebSphere Application Server

6. Leave all other defaults and click **Apply**.
7. Click **Save** in the Messages box.
8. Click **Save**.
9. Restart WebSphere Application Server.

Setting timeout for an enterprise application

To set session timeout for an enterprise application:

1. With WebSphere Application Server started, go to your WebSphere Application Server Administrative Console, expand **Applications** -> **Enterprise Applications**.
2. Select the application name that you want to set timeout for (for example, demoServlet_war).
3. Under Additional Properties, click **Session Management**.
4. From the Session timeout box, select either **No timeout** or **Set timeout**. If you choose **Set timeout**, enter the number of minutes that your session can run before it times out:



WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://localhost:9090/admin/secure/logon.do
Go
WebSphere Application Server Version 5
Administrative Console
IBM
Home | Save | Preferences | Logout | Help
User ID: ido
ido
Servers
Applications
Enterprise Applications
Install New Application
Resources
Security
Environment
System Administration
Troubleshooting
Done
Local intranet

Session Management

Session manager configuration properties allow you to control the behavior of HTTP session support

Configuration

General Properties		
Overwrite Session Management	□ Overwrite	Specifies whether this SessionManager settings are to be used for the current module. Default is to use Session Manager settings defined on parent object.
Session tracking mechanism:	□ Enable SSL ID tracking ☑ Enable Cookies □ Enable URL Rewriting □ Enable protocol switch rewriting	Specify a mechanism for HTTP session management.
Overflow:	☑ Allow overflow	Whether to allow the number of sessions in memory to exceed the value specified by Max In Memory Session Count property. This is valid only in non-persistent sessions mode.
Maximum in-memory session count:	1000 sessions	Specifies the maximum number of sessions to maintain in memory.
Session timeout:	○ No timeout ● Set timeout 30 minutes	Specifies how long a session is allowed to go unused before it will be considered valid no longer. Specify either "Set timeout" or "No timeout." If you select to set the timeout, the value must be at least two minutes, specified in minutes.
Security integration	□ Enable	When security integration is enabled, the Session Manager will associate the identity of users with their HTTP sessions.
Serialize session access:	□ Allow serial access Maximum wait time : 5 seconds ☑ Allow access on timeout	Serialize session access indicates whether to disallow concurrent session access in a given server (JVM).

WebSphere Status < Previous Next > May 17, 2005 4:58:13 PM PDT

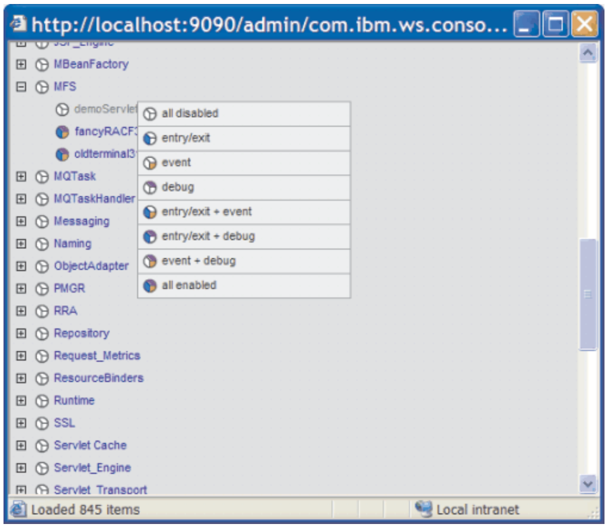
WebSphere Configuration Problems			
Total Configuration Problems: 0	0 total	0 total	0 total

Figure 4. Setting timeout for an enterprise application

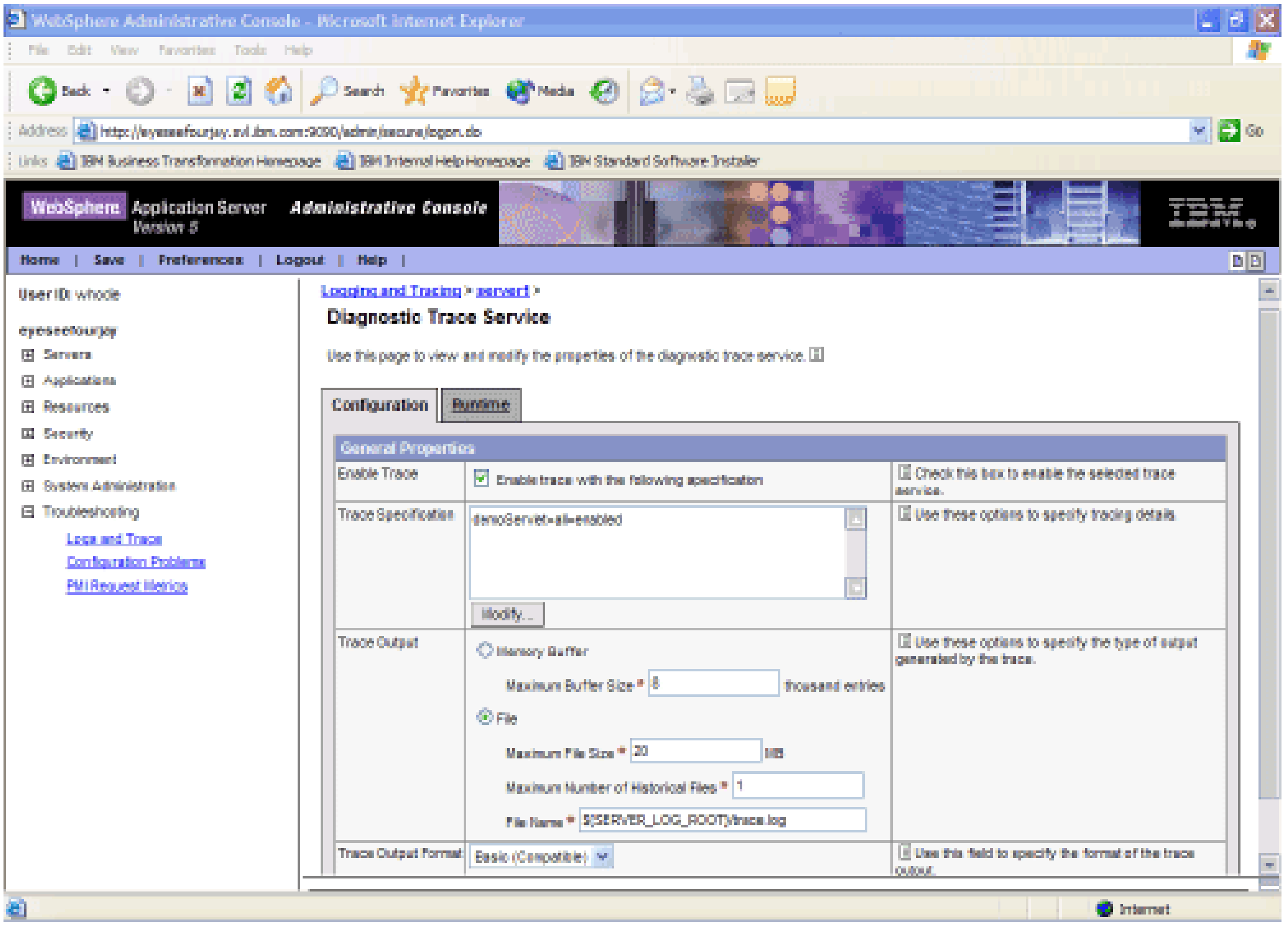
Configuring logging and tracing for WebSphere Application Server Version 5.1.1

To configure logging and tracing for WebSphere Application Server Version 5.1.1, complete the following steps:

1. With WebSphere Application Server started, go to the WebSphere Application Server Administrative Console, expand **Troubleshooting**, and then click **Logs and Trace**.
2. In the Logging and Tracing view, click the server name that you want to configure (for example, `server1`).
3. Under General Properties, select **Diagnostic Trace**.
4. In the **Diagnostic Trace Service** window, modify the Trace Specification by clicking **Modify...**. This opens a window that lists all of the groups.



5. Select the required trace level and then click **Apply**. For more information about the different trace levels and combinations, see Logging and tracing.
6. Verify that the selection shows up in the Trace Specification box:


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://eyeseefourjay.svl.ibm.com:9090/admin/secure/logon.do
Go
Links IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer
WebSphere Application Server Version 5 Administrative Console
IBM
Home | Save | Preferences | Logout | Help |
User ID: whode
eyeseefourjay
Servers
Applications
Resources
Security
Environment
System Administration
Troubleshooting
Logs and Trace
Configuration Problems
PMI Request Metrics
Internet

Logging and Tracing > server1 >

Diagnostic Trace Service

Use this page to view and modify the properties of the diagnostic trace service.

Configuration | Runtime

General Properties

Enable Trace	☑ Enable trace with the following specification	Check this box to enable the selected trace service.
Trace Specification	demoServlet=all=enabled Modify...	Use these options to specify tracing details.
Trace Output	○ Memory Buffer Maximum Buffer Size * 8 thousand entries ● File Maximum File Size * 20 MB Maximum Number of Historical Files * 1 File Name * ${SERVER_LOG_ROOT}/trace.log	Use these options to specify the type of output generated by the trace.
Trace Output Format	Basic (Compatible)	Use this field to specify the format of the trace output.

Figure 5. Trace Specification box

7. Click **Save** in the Messages box.
8. Click **Save**.

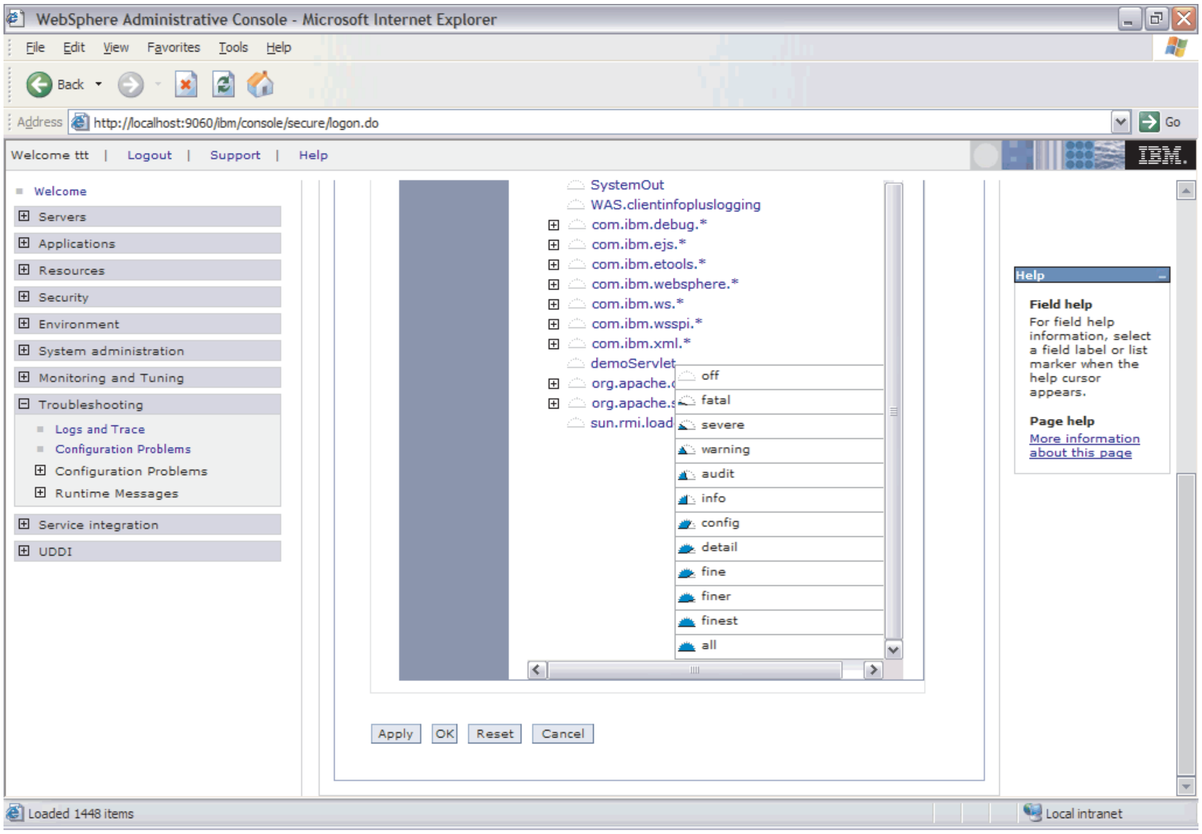
9. Restart WebSphere Application Server.

For more information about logging and tracing, see Logging and tracing.

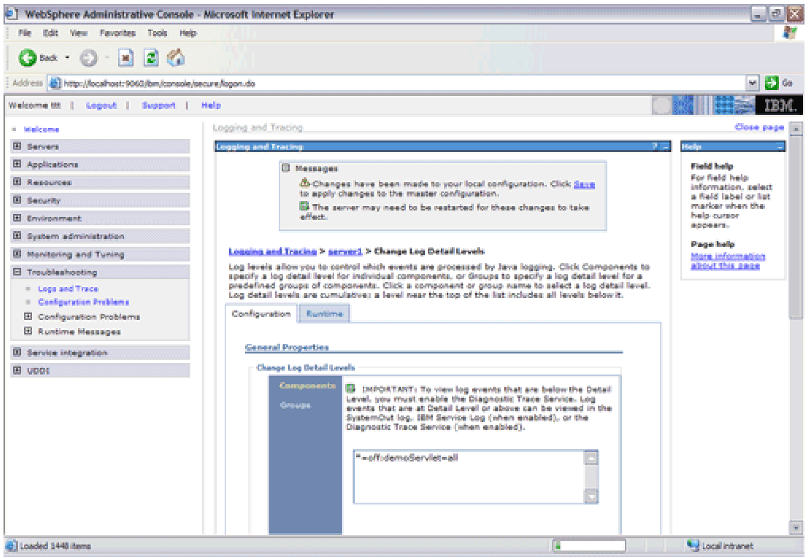
Configuring logging and tracing for WebSphere Application Server Version 6.0

To configure logging and tracing for WebSphere Application Server version 6.0, complete the following steps:

1. With WebSphere Application Server started, go to the WebSphere Application Administrative Console, expand **Troubleshooting** and then click **Logs and Trace**.
2. In the Logging and Tracing view, click the server name that you want to configure (for example, `server 1`).
3. Click **Change Log Detail Level**. A window that lists all of the groups is opened.



4. Select the required trace level and then click **Apply**.
5. Verify that the selection shows up in the Trace Specification Box:



Messages

Changes have been made to your local configuration. Click Save to apply changes to the master configuration.

The server may need to be restarted for these changes to take effect.

Logging and Tracing > server1 > Change Log Detail Levels

Log levels allow you to control which events are processed by Java logging. Click Components to specify a log detail level for individual components, or Groups to specify a log detail level for a predefined groups of components. Click a component or group name to select a log detail level. Log detail levels are cumulative; a level near the top of the list includes all levels below it.

IMPORTANT: To view log events that are below the Detail Level, you must enable the Diagnostic Trace Service. Log events that are at Detail Level or above can be viewed in the SystemOut log, IBM Service Log (when enabled), or the Diagnostic Trace Service (when enabled).

```
*=off:demoServlet=all
```

6. Click **Save** in the Messages box.
7. Click **Save**.
8. Restart WebSphere Application Server.

For more information about logging and tracing, see “Logging and tracing in MFS Web Enablement” on page 53.

Configuring the OMVS environment

To use MFS Web Enablement in an OMVS environment, you must configure your environment. This configuration process involves using FTP to move all the appropriate files to the OMVS environment, and setting permissions to allow read and write access to the files.

To configure your environment:

1. From the workstation where MFS XML Utility was run, use FTP in binary mode to copy the following files to a directory in the OMVS environment:
   - MFS XMI files (which are generated by the MFS XML Utility) from MFS source files.
   - Style sheets
   - MFS JAR files (which are referred to in the Class path of the Resource Adapter Configuration)
2. Change the permissions on both the directory and the files to 755 or 777 so that they both have read and write access.

Configuring form-based authentication for MFS Web Enablement in WebSphere Application Server (optional)

To enable form-based authentication for j_security_check, you need to make several modifications within the web.xml file. The web.xml file needs to be updated with the necessary security and login descriptions.

The updates needed are as follows:

filter and filter-mapping

```
<filter>
<filter-name>MFSWEFilter</filter-name>
<filter-class>MFSWEFilter</filter-class>
</filter>
```

```
<filter-mapping>
<filter-name>MFSWEFilter</filter-name>
<url-pattern>/j_security_check</url-pattern>
</filter-mapping>
```

load-on-startup

```
<load-on-startup>1</load-on-startup>
```

security-role-ref

```
<security-role-ref>
<role-name>AllAuthenticated</role-name>
<role-link>AllAuthenticated</role-link>
</security-role-ref>
```

security-constraint

```
<security-constraint>
<web-resource-collection>
<web-resource-name>secured test resource</web-resource-name>
<url-pattern>demoServlet</url-pattern>
<http-method>GET</http-method>
<http-method>POST</http-method>
</web-resource-collection>
<auth-constraint>
<description>AllAuthenticated Constraint:+:</description>
<role-name>AllAuthenticated</role-name>
</auth-constraint>
</security-constraint>
```

login-config

```
<login-config>
<auth-method>FORM</auth-method>
<form-login-config>
<form-login-page>/WEB-INF/login.jsp</form-login-page>
<form-error-page>/WEB-INF/error.jsp</form-error-page>
</form-login-config>
</login-config>
```

security-role

```
<security-role>
<description>AllAuthenticated role</description>
<role-name>AllAuthenticated</role-name>
</security-role>
```

Note: The order of the elements in the web.xml file does matter and must be as follows:

```
<web-app> <filter> <filter-mapping> <listener> <servlet> <servlet-mapping> <session-config>
<mime-mapping> <welcome-file-list> <error-page> <taglib> <resource-ref> <security-constraint>
<login-config> <security-role> <env-entry> <ejb-ref>
```

The following is a sample web.xml with the updates included:

```
<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>

<filter>
<filter-name>MFSWEFilter</filter-name>
<filter-class>MFSWEFilter</filter-class>
</filter>
<filter-mapping>
<filter-name>MFSWEFilter</filter-name>
<url-pattern>/j_security_check</url-pattern>
</filter-mapping>
<servlet>
<servlet-name>demoServlet</servlet-name>
<servlet-class>demoServlet</servlet-class>
<init-param>
```

```
<param-name>hostName</param-name>
<param-value>ecdvl92.svl.ibm.com</param-value>
</init-param>
<init-param>
<param-name>dataStore</param-name>
<param-value>IMS1</param-value>
</init-param>
<init-param>
<param-name>portNumber</param-name>
<param-value>9999</param-value>
</init-param>
<init-param>
<param-name>MFSXMIRepositoryURI</param-name>
<param-value>file:/c:\xmi</param-value>
</init-param>
<init-param>
<param-name>MFSStyleSheet</param-name>
<param-value>file:/c:/$Projects/MFSXML/source/exampleIEN6.xsl</param-value>
</init-param>
<init-param>
<param-name>traceLevel</param-name>
<param-value>3</param-value>
</init-param>

<init-param>
<param-name>serverPlatform</param-name>
<param-value>1</param-value>
</init-param>
<load-on-startup>1</load-on-startup>
<security-role-ref id="SecurityRoleRef_1">
<role-name>AllAuthenticated</role-name>
<role-link>AllAuthenticated</role-link>
</security-role-ref>

</servlet>

<servlet-mapping>
<servlet-name>demoServlet</servlet-name>
<url-pattern>/demoServlet</url-pattern>
</servlet-mapping>

<security-constraint id="SecurityConstraint_1">
<web-resource-collection id="WebResourceCollection_1">
<web-resource-name>secured test resource</web-resource-name>
<url-pattern>/classes/*</url-pattern>
<http-method>GET</http-method>
<http-method>POST</http-method>
</web-resource-collection>
<auth-constraint id="AuthConstraint_1">
<description>AllAuthenticated Constraint:+:</description>
<role-name>AllAuthenticated</role-name>
</auth-constraint>

<login-config>
<auth-method>FORM</auth-method>
<form-login-config>
<form-login-page>/login.html</form-login-page>
<form-error-page>/error.jsp</form-error-page>
</form-login-config>
</login-config>

<security-role>
<description>AllAuthenticated role</description>
<role-name>AllAuthenticated</role-name>
</security-role>
</web-app>
```

Here are the sample login.jsp and error.jsp pages that can be used for form-based user authentication. The login.jsp and error.jsp pages that you create need to be packaged in the Web application archive file for deployment on WebSphere Application Server. For more information, see:

- ″Developing servlet filters for form login processing″ in the WebSphere Application Server information center inside the WebSphere Application Server Enterprise, Version 5.0.x information
- ″Example: Form login″ in the WebSphere Application Server information center inside the WebSphere Application Server Enterprise, Version 5.0.x information

login.jsp

```
<html>
<head>
<title>Login Page for Example FormBasedAuth</title>
</head>
<body bgcolor="white">
<h2>Custom Login Page</h2>
<hr>

<!--
  This is the custom logon page.  You must use the exact action and form field names
  for a custom logon page.
-->

<form method="POST" action="j_security_check">
  <table border="0" cellspacing="5">
    <tr>
      <th align="right">Username:</th>
      <td align="left"><input type="text" name="j_username"></td>
    </tr>
    <tr>
      <th align="right">Password:</th>
      <td align="left"><input type="password" name="j_password"></td>
    </tr>
    <tr>
      <td align="right"><input type="submit" value="Log In"></td>
      <td align="left"><input type="reset"></td>
    </tr>
  </table>
</form>
</body>
</html>
```

error.jsp

```
<html>
<head>
<title>Error Login Page for Example FormBasedAuth</title>
</head>
<body bgcolor="WHITE">
<h2>Custom Login Page</h2>
<hr>
<h3 style="color: red">Incorrect Username/Password</h3>

<form method="POST" action="j_security_check">
  <table border="0" cellspacing="5">
    <tr>
      <th align="right">Username:</th>
      <td align="left"><input type="text" name="j_username"></td>
    </tr>
    <tr>
      <th align="right">Password:</th>
      <td align="left"><input type="password" name="j_password"></td>
    </tr>
    <tr>
      <td align="right"><input type="submit" value="Log In"></td>
      <td align="left"><input type="reset"></td>
    </tr>
  </table>
</form>
</body>
</html>
```

Configuring a Web browser for MFS Web Enablement

Your Web browser must be properly configured for use with MFS Web Enablement.

The following topics provide additional information:
- "Configuring a Microsoft Internet Explorer browser"
- "Configuring a Mozilla Firefox Web browser" on page 22

Configuring a Microsoft Internet Explorer browser

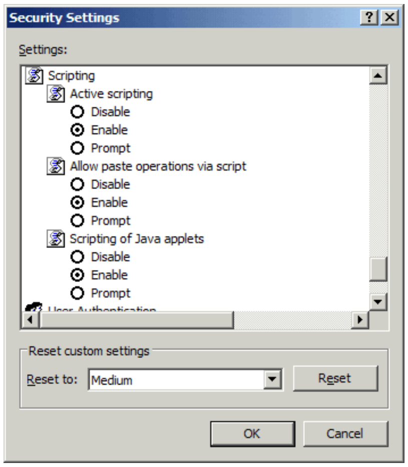
To configure Microsoft Internet Explorer for use with MFS Web Enablement, you must enable Java script and cookies, and optionally, SSL.

The following topics provide additional information:
- "Enabling Java script for Microsoft Internet Explorer"
- "Enabling cookies for Microsoft Internet Explorer" on page 21
- "Enabling SSL for Microsoft Internet Explorer (optional)" on page 21

Enabling Java script for Microsoft Internet Explorer

Java script must be enabled in Microsoft for use with MFS Web Enablement.

To enable Java script:
1. Open your Microsoft Internet Explorer browser and select **Tools** -> **Internet Options**.
2. Select the **Security** tab.
3. Select **Internet** or **Local Internet**, depending your network.
4. Click the **Custom Level** button.
5. Scroll down to **Scripting** -> **Active Scripting** and click **Enable**:



Scripting
Active scripting
○ Disable
● Enable
○ Prompt
Allow paste operations via script
○ Disable
● Enable
○ Prompt
Scripting of Java applets
○ Disable
● Enable
○ Prompt

Reset to: Medium

Figure 6. Enabling active scripting

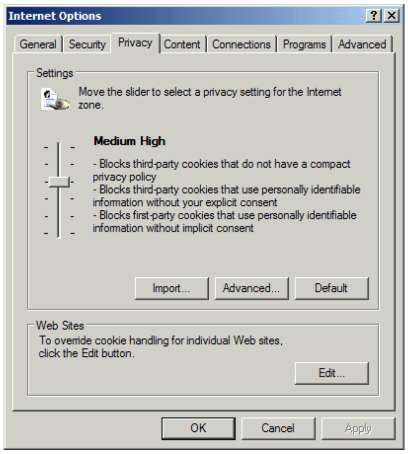
6. Click **OK** to save the setting.
7. Click **OK**.

Enabling cookies for Microsoft Internet Explorer

Cookies must be enabled for use with MFS Web Enablement.

To enable cookies:

1. Open your Microsoft Internet Explorer browser and select **Tools** -> **Internet Options**.
2. Select the **Privacy** tab.
3. Ensure that your privacy settings are set to at least **Medium High**.



Move the slider to select a privacy setting for the Internet zone.

Medium High

- Blocks third-party cookies that do not have a compact privacy policy
- Blocks third-party cookies that use personally identifiable information without your explicit consent
- Blocks first-party cookies that use personally identifiable information without implicit consent

To override cookie handling for individual Web sites, click the Edit button.

Figure 7. Microsoft Internet Explorer privacy settings

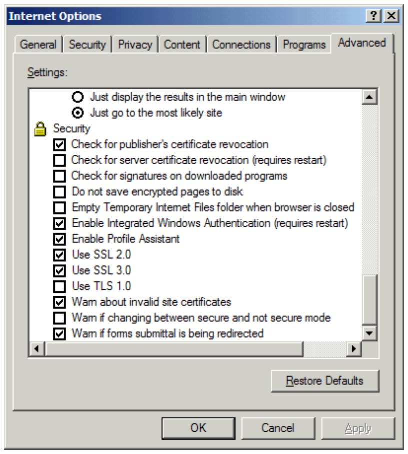
4. Click **OK** to save the setting.

Enabling SSL for Microsoft Internet Explorer (optional)

SSL enablement is optional. If you choose to enable SSL, you will need to have it enabled in both your Web browser and in WebSphere Application Server. For more information about SSL enablement in WebSphere Application Server, see "Configuring SSL in WebSphere Application Server" on page 10.

To enable SSL:

1. Open your Microsoft Internet Explorer browser and select **Tools** -> **Internet Options**.
2. Select the **Advanced** tab.
3. Scroll down to **Security** and make sure that the **Use SSL 2.0** and **Use SSL 3.0** check boxes are selected.



○ Just display the results in the main window
● Just go to the most likely site
Security
☑ Check for publisher's certificate revocation
☐ Check for server certificate revocation (requires restart)
☐ Check for signatures on downloaded programs
☐ Do not save encrypted pages to disk
☐ Empty Temporary Internet Files folder when browser is closed
☑ Enable Integrated Windows Authentication (requires restart)
☑ Enable Profile Assistant
☑ Use SSL 2.0
☑ Use SSL 3.0
☐ Use TLS 1.0
☑ Warn about invalid site certificates
☐ Warn if changing between secure and not secure mode
☑ Warn if forms submittal is being redirected

Figure 8. Enabling SSL

4. Click **OK**.

Configuring a Mozilla Firefox Web browser

To configure Mozilla Firefox for use with MFS Web Enablement, you must enable Java script and cookies, and optionally, SSL.

The following topics provide additional information:

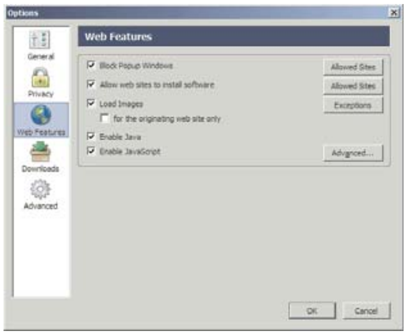
- "Enable Java script for Mozilla Firefox"
- "Enabling cookies for Mozilla Firefox" on page 23
- "Enabling SSL for Mozilla Firefox (optional)" on page 23

Enable Java script for Mozilla Firefox

To use MFS Web Enablement in Mozilla Firefox you must enable Java script.

To enable Java script:

1. Open your Mozilla Firefox browser and select **Tools** -> **Internet Options**.
2. Click **Web Features** and make sure that the **Enable Java** and **Enable JavaScript** check boxes are selected.



☑ Block Popup Windows
☑ Allow web sites to install software
☑ Load Images
☐ for the originating web site only
☑ Enable Java
☑ Enable JavaScript

Figure 9. Enabling Java and JavaScript

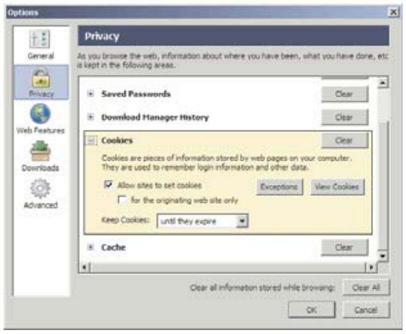
3. Click **OK**.

Enabling cookies for Mozilla Firefox

To use MFS Web Enablement in Mozilla Firefox you must enable cookies.

To enable cookies:

1. Open your Mozilla Firefox browser and select **Tools** -> **Internet Options**.
2. Click **Privacy** and make sure that the **Allow sites to set cookies** check box is selected.



As you browse the web, information about where you have been, what you have done, etc is kept in the following areas.

Cookies are pieces of information stored by web pages on your computer. They are used to remember login information and other data.

☑ Allow sites to set cookies

□ for the originating web site only

Keep Cookies: until they expire

Clear all information stored while browsing:

Figure 10. Enabling cookies

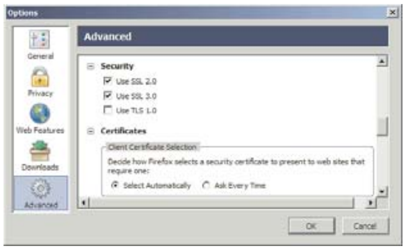
3. Click **OK**.

Enabling SSL for Mozilla Firefox (optional)

SSL enablement is optional. If you choose to enable SSL, you will need to have it enabled in both your Web browser and in WebSphere Application Server. For more information about SSL enablement in WebSphere Application Server, see "Configuring SSL in WebSphere Application Server" on page 10.

To enable SSL:

1. Open your Mozilla Firefox browser and select **Tools** -> **Internet Options**.
2. Click **Advanced**.
3. Scroll down to **Security** and make sure that the **Use SSL 2.0** and **Use SSL 3.0** check boxes are selected.



☑ Use SSL 2.0

☑ Use SSL 3.0

□ Use TLS 1.0

Decide how Firefox selects a security certificate to present to web sites that require one:

● Select Automatically ○ Ask Every Time

Figure 11. Enabling SSL

4. Click **OK**.

Chapter 3. MFS Servlet

The MFS Servlet works with HTTP session objects to:

- Load state information associated with the unique session ID. Create a new session if the request comes in with new session ID.
- Manage and update the state information in each HTTP request.
- Invalidate sessions when an HTTP session becomes unbound (upon logout, browser closed, or session timeout).

The instance servlet, which is generated from the MFS XML Utility, extends the MFS Servlet. The instance servlet name and initialization parameters are recorded in the web.xml file in the Web application archive (WAR) file. See the *MFS XML Utility User's Guide and Reference* for more information.

Upon receiving the HTTP request, the MFS Servlet retrieves the state information for the client. The MFS Servlet recognizes the following types of requests:

Transaction (submit)
The MFS Servlet sends the user's input data in one or more physical pages to the MFS Adapter. If multiple physical pages input (MPPI) is specified for the device page in the MFS source file, the MFS Servlet displays one physical page at a time to collect the data belonging to the same device page and sends them all at once to the MFS Adapter.

RACF information
The RACF information is used by the MFS Adapter to create a host connection. The MFS Servlet stores the latest RACF information. If the request comes from the blank page, the MFS Servlet first checks to see if the request contains different RACF user ID, group name, and password values. If yes, the MFS Servlet updates the information and terminates the existing (conversational) connection.

MFS function keys
The MFS Servlet supports function keys defined from PF1 to PF36. If the request is a function key request, the MFS Servlet either fills the literal value into a device field, as specified in the function key definition, or performs the specified control function. The literal value can be field data, format, or exit commands. The supported control functions are next physical page and end multiple physical pages input.

MFS paging
The MFS Servlet supports the next page paging request similar to PA1 on a 3270 terminal. The MFS Servlet keeps track of the current logical and physical page position and displays one physical page at a time for every Next Page request. If the MFS Servlet receives a Next Page request on the last physical page on a logical page, then it returns the next logical page's first physical page and iterates through until the last physical page of the last logical page. When a next page request is received on the last logical and physical page, the same page is displayed.

MFS formatting and IMS conversational EXIT command
The MFS Servlet supports IMS commands, including formatting (`/FOR` or `/FORMAT`) for loading the specified module name for display, and conversational command (`/EXIT`) for terminating conversational transactions. For more information, see "MFS Servlet messages" on page 47.

Reset Upon receiving the reset request, the MFS Servlet clears the current state and redirects the user to the initial blank page.

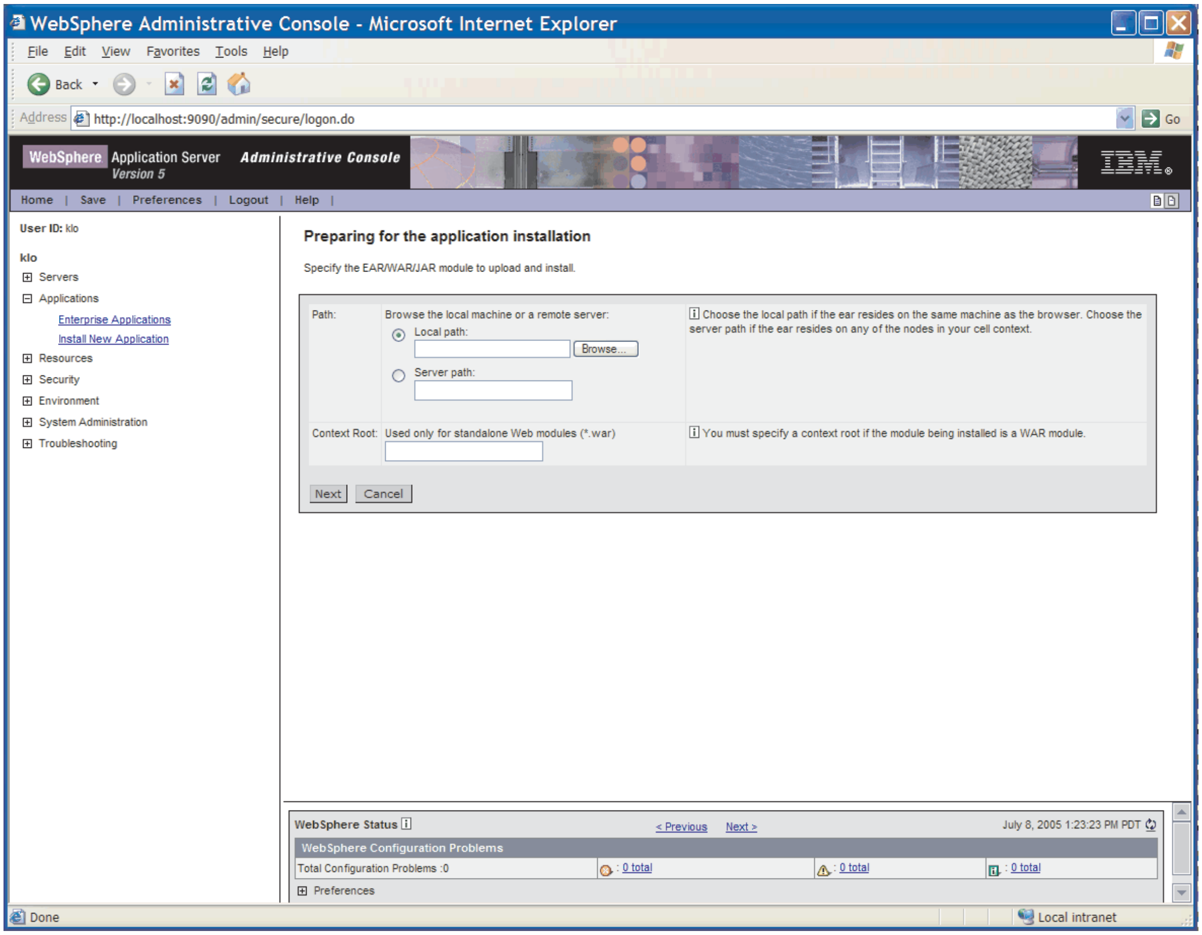
Logout
Upon receiving the logout request, the MFS Servlet invokes MFS Adapter to end the conversation if in a conversation, closes the connection, dumps all state data associated with the session ID, and redirects the user to the logout page.

Note: For better performance, you should always Logout when finished using the MFS-based IMS transactions. Logging out releases objects held in memory immediately.

Installing the instance servlet Web Application Archive (WAR) file

To install the instance servlet Web Application Archive (WAR) file:

1. Start WebSphere Application Server.
2. Open the WebSphere Administrative Console.
3. Expand **Applications** and select **Install New Application**.
4. Click the **Browse** button, under Local path, to select the WAR file you want to deploy.



http://localhost:9090/admin/secure/logon.do

Preparing for the application installation

Specify the EAR/WAR/JAR module to upload and install.

Browse the local machine or a remote server:

Choose the local path if the ear resides on the same machine as the browser. Choose the server path if the ear resides on any of the nodes in your cell context.

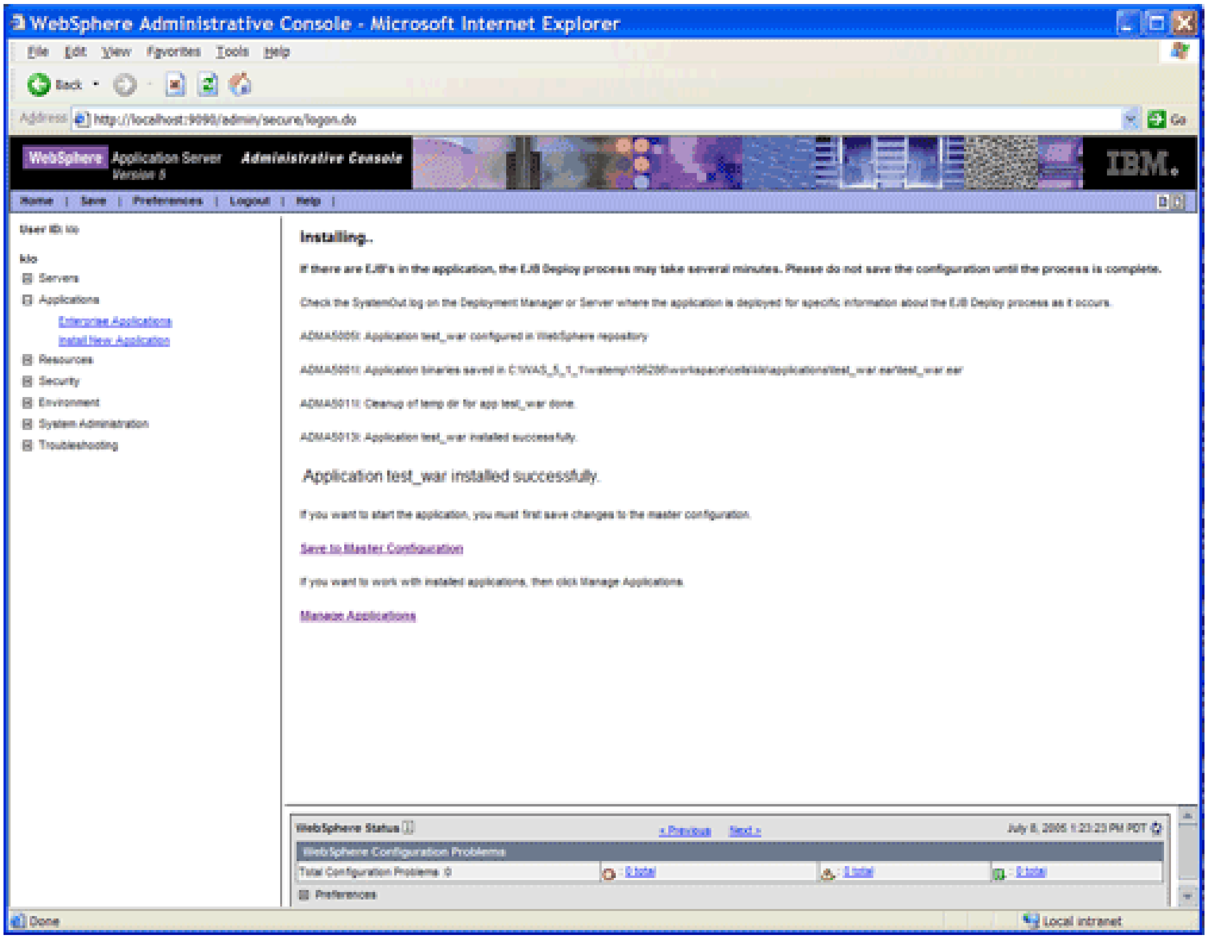
Used only for standalone Web modules (*.war)

You must specify a context root if the module being installed is a WAR module.

July 8, 2005 1:23:23 PM PDT

Total Configuration Problems :0 | 0 total | 0 total | 0 total

5. Enter the Context Root. The text you enter will be a part of the URL.
6. Click the **Next** button to go to the next screen and accept the default values.
7. Click the **Next** button to go to the Application Security Warnings window.
8. Click the **Continue** button to go to the Install New Application window and accept the default values.
9. Click the **Next** button and accept the defaults three more times.
10. Click the **Finish** button. You should get the message `Application [`*`application name`*`] installed successfully`.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://localhost:9090/admin/secure/logon.do
Go
WebSphere Application Server Version 5 Administrative Console
IBM.
Home | Save | Preferences | Logout | Help |
User ID: klo
klo
Servers
Applications
Enterprise Applications
Install New Application
Resources
Security
Environment
System Administration
Troubleshooting
WebSphere Status
< Previous Next >
July 8, 2005 1:23:23 PM PDT
WebSphere Configuration Problems
Total Configuration Problems: 0
0 total
0 total
0 total
Preferences
Done
Local intranet

Installing..

If there are EJB's in the application, the EJB Deploy process may take several minutes. Please do not save the configuration until the process is complete.

Check the SystemOut.log on the Deployment Manager or Server where the application is deployed for specific information about the EJB Deploy process as it occurs.

ADMA5005I: Application test_war configured in WebSphere repository

ADMA5001I: Application binaries saved in C:\WAS_5_1_1\wstemp\106288\workspace\cells\klo\applications\test_war.ear\test_war.ear

ADMA5011I: Cleanup of temp dir for app test_war done.

ADMA5013I: Application test_war installed successfully.

Application test_war installed successfully.

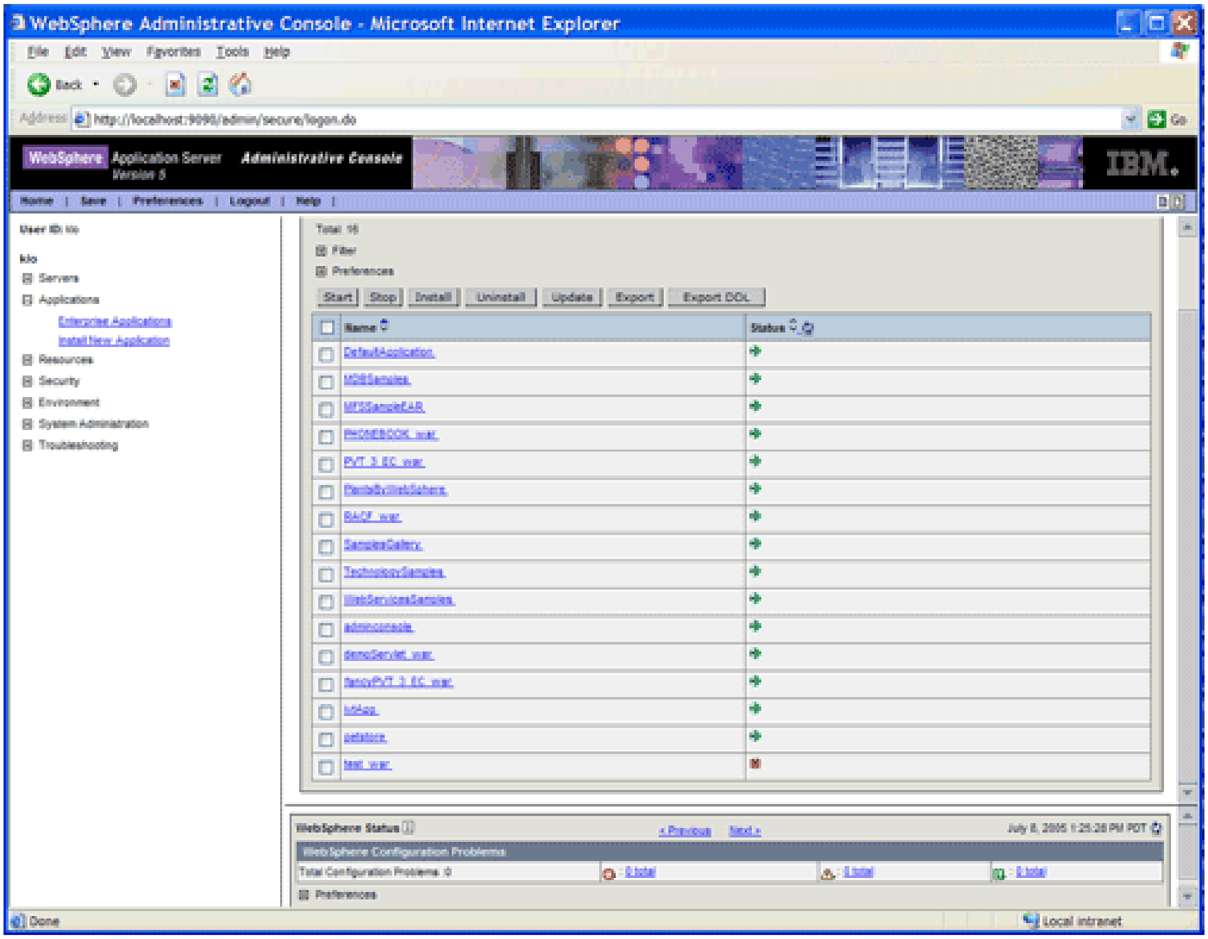
If you want to start the application, you must first save changes to the master configuration.

Save to Master Configuration

If you want to work with installed applications, then click Manage Applications.

Manage Applications

11. Click on the **Save to Master Configuration** link to go to the Save window.
12. Click the **Save** button.
13. To start the application, select **Enterprise Applications**.
14. Select the WAR file that you installed, and click the **Start** button.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://localhost:9090/admin/secure/logon.do
Go
WebSphere Application Server Version 5 Administrative Console
IBM.
Home | Save | Preferences | Logout | Help |
User ID: klo
klo
Servers
Applications
Enterprise Applications
Install New Application
Resources
Security
Environment
System Administration
Troubleshooting
Total: 16
Filter
Preferences
Start
Stop
Install
Uninstall
Update
Export
Export DDL
WebSphere Status
< Previous
Next >
July 8, 2005 1:25:28 PM PDT
WebSphere Configuration Problems
Total Configuration Problems: 0
0 total
0 total
0 total
Preferences
Done
Local intranet

□	Name	Status
□	DefaultApplication	➡
□	MDBSamples	➡
□	MFSSampleEAR	➡
□	PHONEBOOK_war	➡
□	PVT_3_EC_war	➡
□	PlantsByWebSphere	➡
□	RACF_war	➡
□	SamplesGallery	➡
□	TechnologySamples	➡
□	WebServicesSamples	➡
□	adminconsole	➡
□	demoServlet_war	➡
□	fancyPVT_3_EC_war	➡
□	ivtApp	➡
□	petstore	➡
□	test_war	■

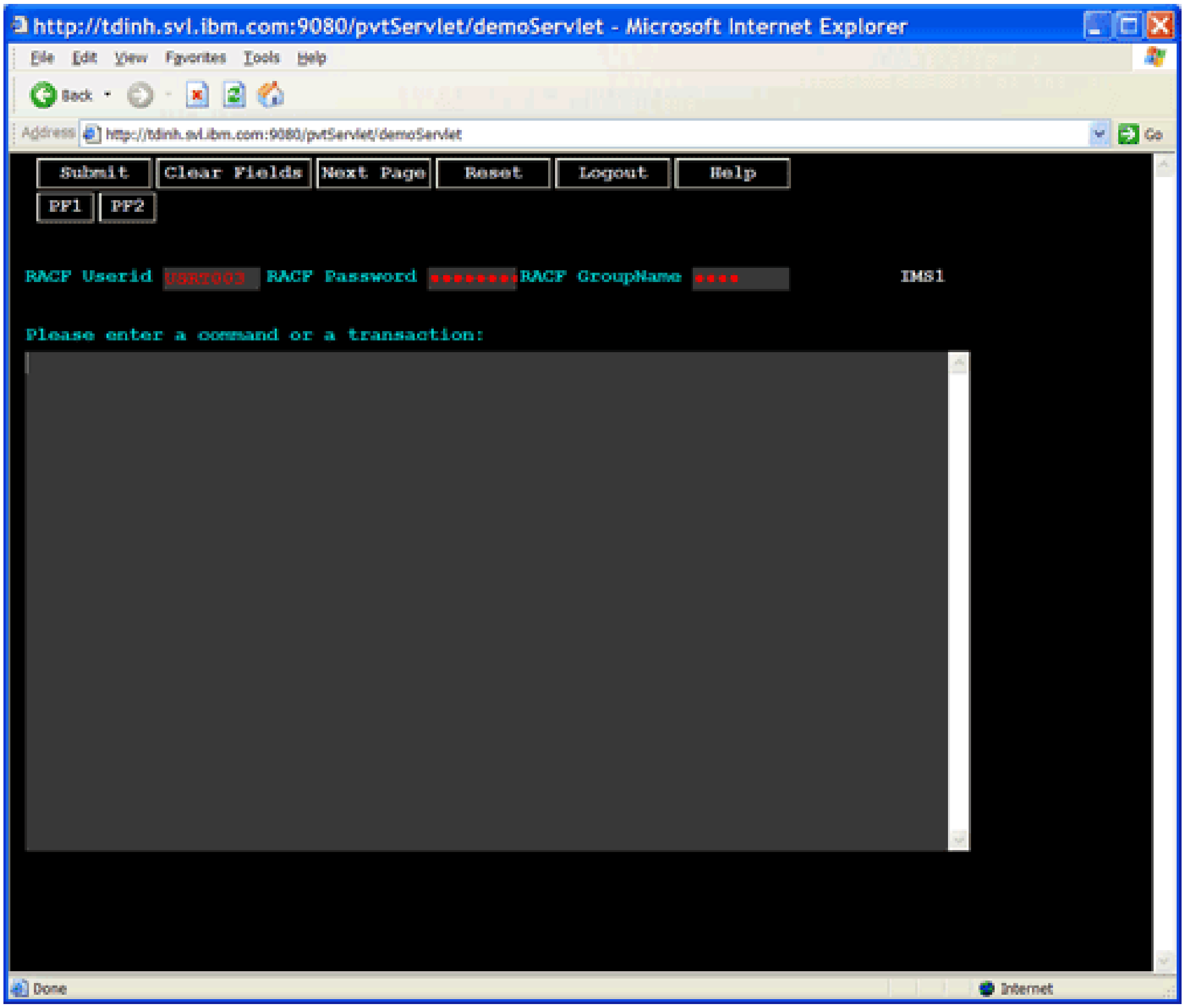
15. You should get the message `Application [`*`application name`*`] on server [`*`server name`*`] and node [node name] started successfully`. The application status color should change from red to green.

Invoking the deployed instance servlet

Before invoking the deployed instance servlet, make sure that your Web server is running.

To start the instance servlet:

1. From a Microsoft Internet Explorer or Mozilla Firefox browser, enter the URL of the instance servlet (for example `https://test.ibm.com:9081/test/testServlet`). If user authentication is set on, proceed to step 2, if not, proceed to step 4.
2. Enter your user ID and password.
3. If the user authentication is on, the Security Alert prompt appears. Click **Yes** to indicate that you want to proceed.
4. After WebSphere Application Server authenticates, you are then redirected to the initial blank page:


http://tdinh.svl.ibm.com:9080/pvtServlet/demoServlet - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://tdinh.svl.ibm.com:9080/pvtServlet/demoServlet
Go
Submit
Clear Fields
Next Page
Reset
Logout
Help
PF1
PF2
Done
Internet

RACF Userid USRT003 RACF Password •••••••• RACF GroupName •••• IMS1

Please enter a command or a transaction:

Figure 12. Initial blank page

5. You can now enter your RACF information, command, or transaction request in the same way that you would using a 3270 type terminal.

   Note: The RACF information displayed defaults to the information specified in the web.xml file. However, you can choose to overwrite the default values. The RACF information is used for creating a connection to IMS Connect. The servlet is timed out in fixed intervals. Make sure that you properly log out to release the connection when you are finished.

Note: The MFS Web Enablement support restricts multiple browsers from sharing the same session ID. The MFS Servlet tracks the state of each client by checking the index associated with every request and disallows requests with old index number. For example, if after opening a new browser window that shares the same session ID which increments the index number, the new browser window will take control.

Browser-specific tips:

- From Internet Explorer: If you open a new browser window by pressing Control + N or by clicking on **File** -> **New** -> **Window**, and then go back to using the previous browser window, you will get the `Session ID Error` page. However, this restriction does not apply to browser windows that are opened by double-clicking on the Microsoft Internet Explorer icon, because a new session ID will be associated with every new instance of the browser.

- From Mozilla Firefox: If you open a new browser window to access the same instance servlet, and then go back to using the previous browser window, you will get the `Session ID Error` page. This restriction does not apply when you are using multiple Mozilla browsers, each invoking a different instance servlet.

Chapter 4. MFS Adapter

The MFS Adapter runs inside WebSphere Application Server and transforms data between MFS device data and message data. The MFS Adapter is invoked by the MFS Servlet. Using the Eclipse Modeling Framework (EMF), the MFS Adapter loads the appropriate MFS XMI resource from the repository, invokes the transformer routine to handle the data conversion, and submits the IMS transaction using IMS Connector for Java's Common Client Interface (CCI) method calls.

Based on the information contained in the DIF/MID XMI file, the transformer routine first maps the input device data into message data, then into an input byte array. The input byte array is sent across using IC4J. Upon successful execution, the output byte array comes back on the return route. The MFS Adapter then loads the DOF/MOD XMI file, specified in the mapName (capable of handling the case with the application program switches the MODNAME), and invokes the transformer routine to first map the output byte array into message data, then into output device data. The resulting data object is returned to the MFS Servlet.

Note: MFS Web Enablement does not support asynchronous send-only message requests.

The MFS Adapter transformer routine implements both the J2EE CCI Record and Streamable interfaces. The javax.resource.cci.Record interface is the base interface for the representation of an input or output to the execute methods defined on a J2EE interaction. The javax.resource.cci.Streamable interface enables a resource adapter to extract data from an input record or set data into an output record as a stream of bytes.

The following table describes various scenarios that the MFS Adapter supports:

Table 1. Supported MFS Adapter scenarios

Scenario	**Description**
Scenario 1: MFS Adapter receives format request	Loads and returns specified modname XMI file (load DFSMO3 if not found)
Scenario 2: MFS Adapter receives exit request	Ends the conversation and returns the status using DFSMO2
Scenario 3: Adapter receives transaction request	Loads and parses using input MID's XMI file.s
Scenario 4: MFS Adapter receives transaction response where the IMS application does not replace MODNAME	Loads and processes using input MID's next MOD XMI file (default is DFSMO2 if unspecified)
Scenario 5: MFS Adapter receives transaction response where the IMS application replaces MODNAME	Loads and processes using MOD XMI file specified in the InteractionSpec's mapName
Scenario 6: MFS Adapter receives transaction response where the output byte array begins with "DFS"	Loads and processes using DFSMO1 (for single segment output) or DFSMO5 (for multiple segment output) XMI file
Scenario 7: MFS Adapter receives transaction response where a runtime exception occurred (MFS Adapter, IMS Connect, IMS Connector for Java, or IMS)	Loads and processes using DFSMO2

Chapter 5. Sample MFS style sheets

Important: The sample MFS style sheets are provided for demonstration purposes and are customizable. They are provided on an as-is basis and no support is provided. The portions of the style sheets that are customizable are notated with comments. Modifying any non-recommended portions can result in unexpected behavior and a possible runtime failure.

The MFS Servlet loads an MFS style sheet to render HTML pages as output. The MFS style sheets supply information on how to render the data in a Web browser.

The MFS style sheets provide functionality similar to that of using a 3270 type terminal, including:

- A **Submit** button on the top of the page that is analogous to pressing the Enter key on a 3270 type terminal.
- The **Next Page** button which is equivalent to the PA1 function. Clicking this button advances you to the next physical page. When you get to the last physical page, clicking the **Next Page** button simply displays the same page.
- The PF keys PF1 through PF36 are displayed as buttons on the HTML pages. Only PF keys with literal data (transaction code and two commands: `/FOR` or `/FORMAT` and `/EXIT`) and two control functions: NEXTPP (next physical page), and ENDMPPI (end multiple physical pages input) are supported.
- A **Clear Fields** button that clears the contents of all input fields.
- A **Reset** button that allows you to return to the blank page.
- A **Logout** button that closes all connections and exits.
- A **Help** button opens the *MFS Web Enablement Version 9.3.0 User's Guide and Reference*.
- Attribute bytes support, including:

  Protected
  Data cannot be entered into this field. Setting this attribute to "true" changes it into a label text field.

  Modified
  Data in this field can be modified. Setting this attribute to "true" changes it into an input text field.

  High-intensity
  Data displayed in this field appears in bold font (default).

  Non-displayable
  Data entered into this field is non-displayable. In the case of label text field, the foreground color is set to the background color. In the case of input text field, the input type is set to hidden.
- Extended Attribute bytes support:

  Highlighting
  - **Default:** This field gets the default font and color assignments.
  - **Blink:** This field is blinking.
  - **Reverse video:** This field's foreground and background colors are reversed.
  - **Underline:** This field is underlined.

  Color Sets a field's color. Eight colors are used:

Table 2. Colors of the MFS sample style sheets

Color name	Color displayed on a classic 3270 type terminal simulation	Color displayed on a stylized 3270 type terminal
Blue	Blue	Blue
Red	Red	Red
Green	Lime-green	rgb(33,70,40)
Turquoise	Aqua	rgb(52,126,124)
Yellow	Yellow	rgb(244,122,0)
Pink	Fuchsia	rgb(160,50,140)
Default	Aqua	rgb(100,50,0)
Neutral	White	rgb(111,111,111)

Outlining

Sets a border around a field:

- **Box:** Sets the border over, under, left, and right. This overrides other outlining extended attributes.
- **Over:** Sets the border on the top of the field.
- **Under:** Sets the border on the bottom of the field.
- **Left:** Sets the border to the left of the field.
- **Right:** Sets the border to the right of the field.

The attributes of the MFS style sheets that you can customize are:

- Font attributes:
  - Color
  - Family
  - Size
  - Weight
- Background color
- Button style
- JavaScript™ code can be added

You can specify additional graphics in the style sheet and add them into the WebSphere application archive (WAR) file. Do not modify the rest of the code in the style sheets.

You can modify the style sheet to add a drop-down list. The following sample code is a style sheet to create a drop-down list for a specific field given the field values are defined in a specified XML format. With drop-down lists, users can choose from a list rather than entering text in a field. The sample code should be added in between the code for creating *textarea* and the code for creating input field as shown in the following sample:

```
- End of creating textarea code:
</xsl:otherwise>
</xsl:choose>
</xsl:element>   <!-- end of element "textarea" -->

</xsl:when><!-- create drop-down list -->
<xsl:variable name="formatName">
<xsl:value-of select="//MFS:MFSFormat/@label"/>
</xsl:variable>
<xsl:variable name="commandName">
<xsl:value-of select="document(concat('file:///xmi/',$formatName, '.xml'))/options/key"/>
```

```
</xsl:variable>
<xsl:choose>
<xsl:when test="@label= $commandName">
<xsl:element name="label">
<xsl:attribute name="for">
<xsl:value-of select="document(concat('file:///xmi/',$formatName, '.xml'))/options/key"/>
</xsl:attribute>
</xsl:element>
<xsl:element name="select">
<xsl:attribute name="name">
<xsl:value-of select="document(concat('file:///xmi/',$formatName, '.xml'))/options/key"/>
</xsl:attribute>
<xsl:attribute name="id">
<xsl:value-of select="document(concat('file:///xmi/',$formatName, '.xml'))/options/key"/>
</xsl:attribute>
<xsl:for-each select="document(concat('file:///xmi/',$formatName, '.xml'))/options/list/option">
<xsl:element name="option">
<xsl:attribute name="value">
<xsl:value-of select="@value"/>
</xsl:attribute>
<xsl:value-of select="@value"/>
</xsl:element>
</xsl:for-each>
</xsl:element>

</xsl:when>

- Beginning of creating input field code:
<xsl:otherwise>
<xsl:element name="input">
<xsl:attribute name="name"><xsl:value-of select="@label"/></xsl:attribute>
<xsl:choose>
<xsl:when test="attributes/@intensity='nondisplayable'">
<xsl:attribute name="type">hidden</xsl:attribute>
```

In addition to modifying the code in the style sheet, you must also create an external XML file that has the same name as the MFS format label, but with an XML extension. Include the following elements in the XML file:

- Key name: the device field label name (the field on the device in which you want the pull-down list)
- A list of all the choices for the pull-down list

The following sample shows the code in a drop-down list XML file, IVTNOF.xml, for the IMS installation verification procedure (IVP) application:

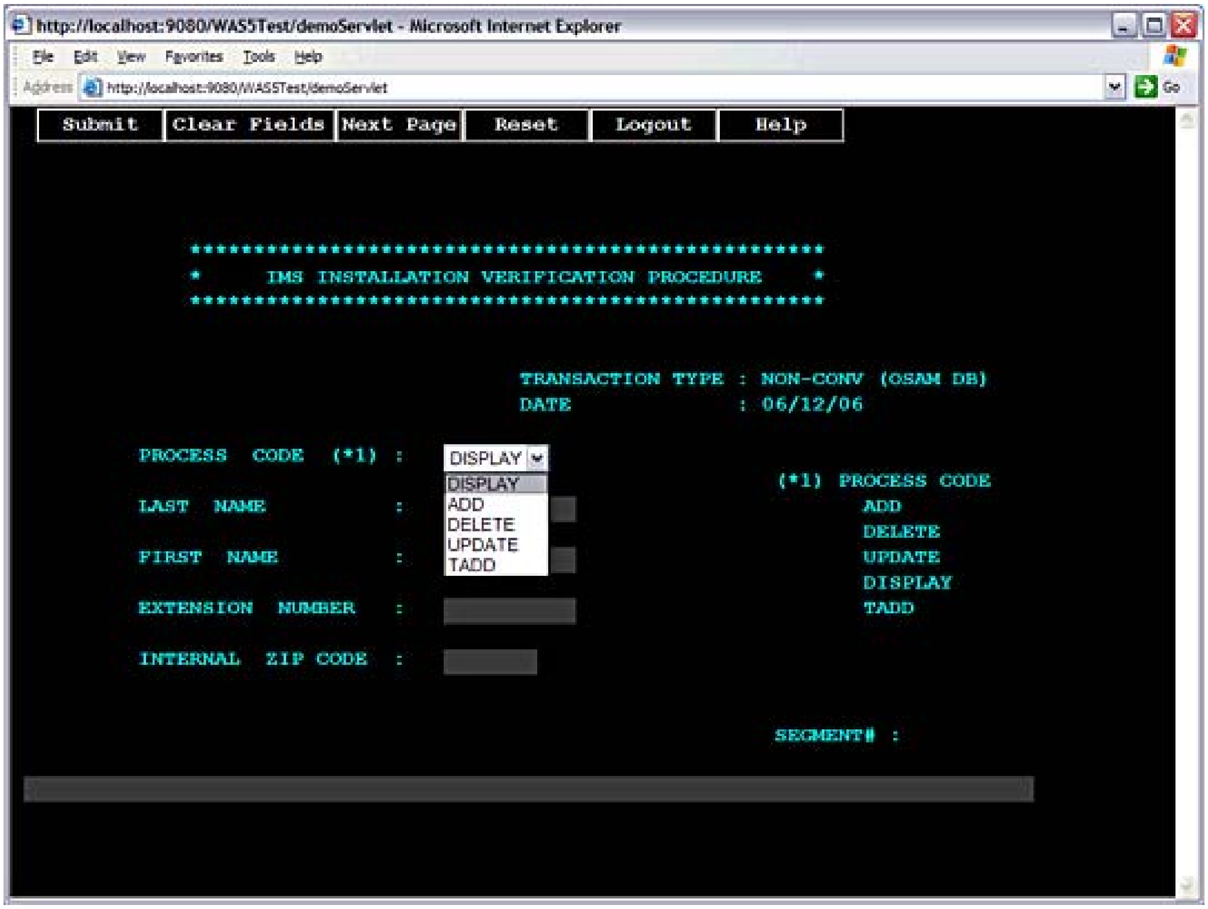
Note: The XML Transformer will generate system error messages if the referenced external XML files are missing.

```
<?xml version='1.0'?>
<options>
  <key>CMD</key>
  <list>
 <option value="DISPLAY"/>
 <option value="ADD"/>
 <option value="DELETE"/>
 <option value="UPDATE"/>
 <option value="TADD"/>
  </list>
</options>
```

Two types of sample MFS style sheets provided:

Classic 3270 type terminal simulation

The following page is rendered with the sample classic 3270 type terminal simulation, and shows the IMS installation verification procedure (IVP):

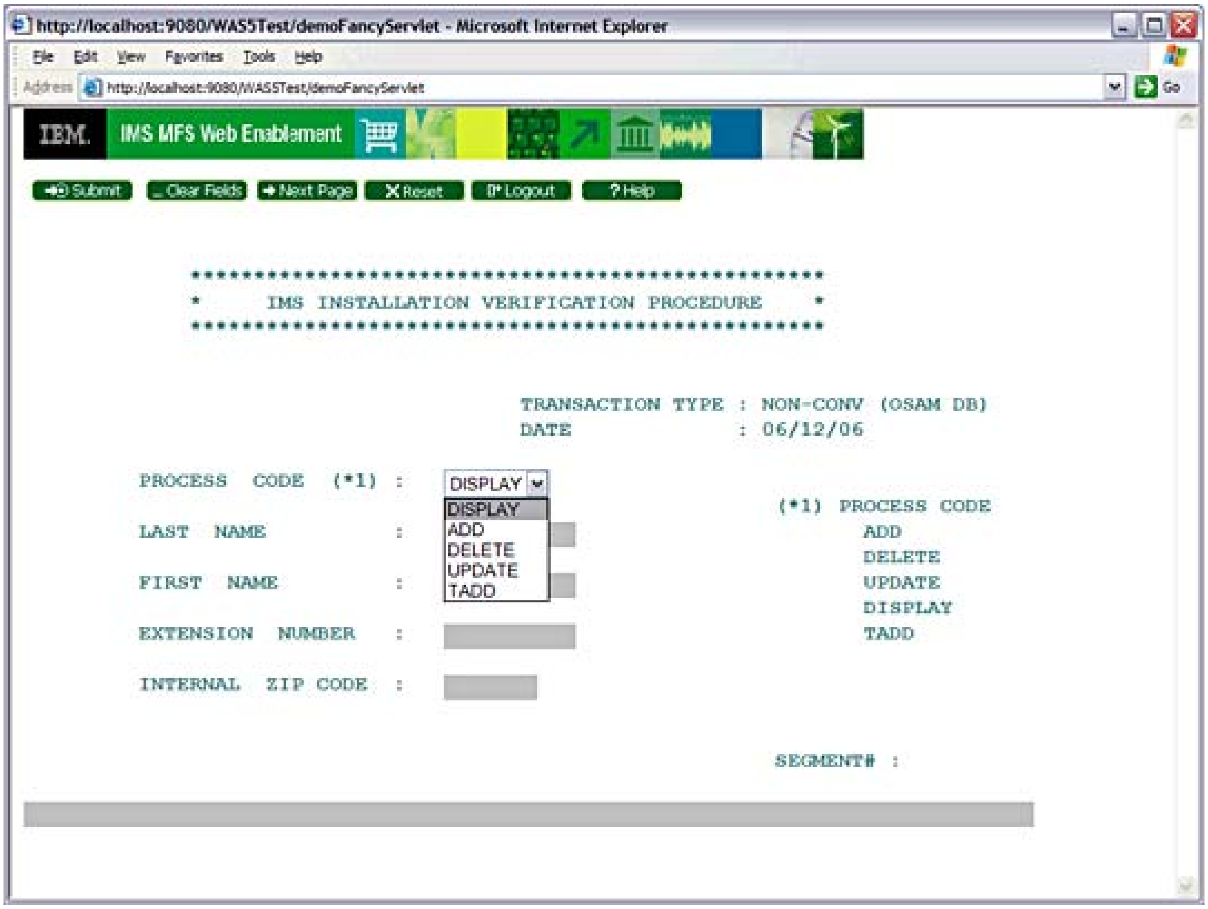

http://localhost:9080/WAS5Test/demoServlet - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:9080/WAS5Test/demoServlet
Go
Submit
Clear Fields
Next Page
Reset
Logout
Help
DISPLAY
ADD
DELETE
UPDATE
TADD

```
     ***************************************************
     *     IMS INSTALLATION VERIFICATION PROCEDURE     *
     ***************************************************

                          TRANSACTION TYPE : NON-CONV (OSAM DB)
                          DATE             : 06/12/06

 PROCESS  CODE  (*1) :  DISPLAY
                                          (*1) PROCESS CODE
 LAST  NAME          :                         ADD
                                               DELETE
 FIRST  NAME         :                         UPDATE
                                               DISPLAY
 EXTENSION  NUMBER   :                         TADD

 INTERNAL  ZIP CODE  :

                                          SEGMENT# :
```

Figure 13. Classic 3270 type terminal simulation

Stylized 3270 type terminal

Displays a Web page interface.


http://localhost:9080/WAS5Test/demoFancyServlet - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:9080/WAS5Test/demoFancyServlet
Go
IBM. IMS MFS Web Enablement
Submit
Clear Fields
Next Page
Reset
Logout
? Help

```
*****************************************************
*     IMS INSTALLATION VERIFICATION PROCEDURE      *
*****************************************************

                           TRANSACTION TYPE : NON-CONV (OSAM DB)
                           DATE             : 06/12/06

PROCESS  CODE  (*1) :   DISPLAY
                                              (*1) PROCESS CODE
LAST  NAME          :                               ADD
                                                    DELETE
FIRST  NAME         :                               UPDATE
                                                    DISPLAY
EXTENSION  NUMBER   :                               TADD

INTERNAL  ZIP CODE  :

                                              SEGMENT# :
```

DISPLAY
ADD
DELETE
UPDATE
TADD

Figure 14. Stylized 3270 type terminal

Chapter 6. Security considerations for MFS Web Enablement

The following section describes the security considerations for MFS Web Enablement.

The following topics provide additional information:
- “Secure Sockets Layer support”
- “WebSphere Application Server user authentication” on page 38

Secure Sockets Layer support

Secure Sockets Layer (SSL) support, also known as HTTPS support, enables the secure transmission of data using 128-bit encryption between the client browser and the WebSphere Application Server.

Note: SSL support between WebSphere Application Server and IMS Connect is not supported.

SSL is based on the public key mechanism and is capable of utilizing many different encryption algorithms, such as: RSA, DES, Triple DES, and Blowfish.

Server SSL support is usually handled automatically when the “http” portion of a URL is replaced with “https”. For example `http://login.ibm.com` changes to `https://login.ibm.com`. A client can also request SSL on their behalf by using `https://` instead of `http://`.

For more information, see “Configuring SSL in WebSphere Application Server” on page 10.

WebSphere Application Server user authentication

Applications programmers can create site-specific login forms by using WebSphere Application Server’s form-login type. The existing J2EE specification defines form-login as one of the authentication methods for Web applications. WebSphere Application Server extends J2EE by also providing a form-logout mechanism.

The form login and logout process works as follows:

1. A user attempts to use a resource (for example, a WAR file) that is secured with a form-login authentication method.
2. The user is redirected to the form-login page, which takes the user to an HTML form that collects the authentication information.
3. The user enters a user ID and password into the form and then submits it.
4. The submission triggers a special WebSphere Application Server servlet that authenticates the user.
5. If the user authenticates successfully, the originally requested resource can be accessed.

To configure the MFS Web Enablement WAR file for form-login, see “Configuring form-based authentication for MFS Web Enablement in WebSphere Application Server (optional)” on page 16.

Chapter 7. Sample instructions to Web enable the IMS IVP Phonebook application

This section provides step-by-step guidelines to Web enable, generate, deploy, and invoke the IMS IVP Phonebook application with the MFS Web Enablement PhoneBook MFS source file, which is at http://www.ibm.com/software/data/ims/toolkit/mfswebsupport/index.html.

Step 1: Parsing the MFS source file with MFS XML Utility

In step 1, you parse the source file dfsivf1.mfs with the MFS XML Utility to generate the IVTNO.xmi and IVTNOMI1.xmi files.

To parse the MFS source files:

1. From the MFS XML Utility window, choose selection 1 and press **Enter**:
   ```
   Please enter your selection here: 1
   Step 1: Generate XMI files that represent MID/DIF and MOD/DOF of the MFS source
           This step requires the following information:
           -MFS source files
           -Device Characteristics Table file (Optional)
           -Whether source files are in text or binary format (default to text)
           -Codepage for source files (default to Cp1252)
           -Codepage for host environment (default to Cp037)
           -Device type to format (default to 3270-A02)
           -Device feature to enable (default to ignore)
           -Output directory for generated XMI files (default to installation directory)
   ```
2. Press **Enter** to run in novice mode:
   ```
   Enter arguments here or press enter to run novice mode. Type '/help' for more
   information or 'q' to quit anytime:
   >>

   Beginning Step 1: Generating XMI files...
   ```
3. Specify `c:\MFSXMLUtility+\dfsivf1.mfs` as the MFS source files and press **Enter**:
   ```
   Specify MFS source files or directory containing MFS source files: c:\MFSXMLUtility+\dfsivf1.mfs
   You selected c:\MFSXMLUtility+\dfsivf1.mfs
   ```
4. Press **Enter** to indicate no device characteristics table:
   ```
   Specify device characteristics table (Optional):
   No device characteristics table selected
   ```
5. Press **Enter** to indicate the default value of `n`:
   ```
   Is the source in binary mode (y/n; default is no):
   >> You entered no by default
   ```
6. Press **Enter** to specify the source codepage as Cp1252 (default based on system locale):
   ```
   Specify codepage for source (our system default is Cp1252):
   >> You entered MS950 by default.
   ```
7. Press **Enter** to specify the host codepage as `Cp037` (Windows Latin 1):
   ```
   Specify codepage for host (Default is Cp037):
   >> You entered Cp037 by default.
   ```
8. Specify `PHONEBOOK` as the output directory and press **Enter** :
   ```
   Specify output directory (Default is C:\MFSXMLUtilityGA\): PHONEBOOK
   >> You entered C:\MFSXMLUtilityGA\PHONEBOOK\

   Parsing files...
   ```
9. Press **Enter** to select the default device type:

```
CChoose one of the following device features (Default is Ignore):
1)Ignore
=>
You selected Ignore

parsing c:\MFSXMLUtility+\dfsivf1.mfs

Writing to C:\MFSXMLUtilityGA\device_types.log completed
```

10. Press **Enter** to specify that you do not want to see the parse output:

```
Parse successfully. Would you like to see the parse output? (y|n; default is no):

Parse output log will be created.


Writing to C:\MFSXMLUtilityGA\parse.log completed


The following XMI files were generated:
 C:\MFSXMLUtilityGA\PHONEBOOK\IVTNOMI1.xmi
 C:\MFSXMLUtilityGA\PHONEBOOK\IVTNO.xmi

XMI files generated.
```

11. Type y and press **Enter** to indicate that you want to save your input values to a batch file:

```
Do you wish to save your input values to a batch file? (y|n ; Default is no)y
Writing batch file to C:\MFSXMLUtilityGA\PHONEBOOK\IVTNOMI1_step1.txt...
Step 1 batch file created

The following batch file was generated:
 C:\MFSXMLUtilityGA\PHONEBOOK\IVTNOMI1_step1.txt

Step 1 completed.
```

Step 2: Generating an instance servlet

In step 2 you generate the PHONEBOOK instance servlet.

To generate an instance servlet:

1. From the MFS XML Utility window, choose selection 2 and press **Enter**:

```
Please enter your selection here: 2

Step 2: Generate and compile instance servlet used during runtime for the backend MFS application;
        This step requires the following information:
        -Name of the this instance servlet
        -Location of XMI repository on web server (default to last value)
        -Name and location of stylingsheet on local machine to copy to web server
         (default to last value)
        -Host name or IP address of IMS (default to last value)
        -Port number of host (default to last value)
        -IMS datastore name (default to last value)
        -RACF username (optional)
        -RACF group (optional)
        -RACF password (required if RACF username is specified)
        -Trace Level of IMS Connect for Java (default to 0)

Begin Servlet Generation....
```

2. Press **Enter** to generate the servlet in novice mode:

```
Press Enter to generate servlet in novice mode, otherwise enter servlet arguments
for expert mode or type '/help' or 'q':
>>
```

3. Enter PHONEBOOK as the name of your instance servlet and press **Enter**:

```
Please enter the name of this instance servlet:PHONEBOOK
```

4. Specify the number 1 or 2 to indicate the operating system.

   ```
   Please select the platform where WebSphere Application Server is located:
   1) WINDOWS
   2) Other Systems (for example, z/OS, AIX)
   >> 2
   You chose Other Systems.
   ```

   If you select UNIX® systems, the XMI repository URI and stylesheet URI are appended with file:// instead of file:/.

5. Specify the output directory for your instance servlet and press Enter (the last output directory specified or the MFS XML Utility installation directory):

   ```
   Specify output directory (default is C:\MFSXMLUtilityGA\PHONEBOOK\):
   You have selected an existing directory! Files with the same name will be over-written
   without warnings!
   Continue? (y|n; default is yes)
   >> You entered C:\MFSXMLUtilityGA\PHONEBOOK\
   ```

6. Specify `/c:\xmi` as the file path URI of the XMI repository on WebSphere Application Server and press **Enter**:

   ```
   Specify target location of XMI repository on web server ('?' for help): c:\xmi
   >> You entered file:/c:\xmi
   ```

7. Specify `c:\$Projects\MFSXML\source\sample3270.xsl` as the target location of your style sheet on WebSphere Application Server and press **Enter**:

   ```
   Specify location of styling sheet ('?' for help): c:\$Projects\MFSXML\source\sample3270.xsl
   >> You entered file:/c:\$Projects\MFSXML\source\sample3270.xsl
   ```

8. Specify your IMS hostname or IP address and press **Enter**:

   ```
   Specify IMS hostname or IP address ('?' for help): ecdb31.svl.ibm.com
   >> You entered ecdb31.svl.ibm.com
   ```

9. Specify 9999 as the host port number and press **Enter**:

   ```
   Specify a port number ('?' for help): 9999
   >> You entered 9999
   ```

10. Specify IMS1 as the IMS datastore name and press **Enter**:

    ```
    Specify IMS1 as the IMS datastore name ('?' for help): IMS1
    >> You entered IMS1
    ```

11. Enter your RACF credentials or skip this step by pressing **Enter**:

    ```
    Note that the following RACF information will be used if no RACF information is
    specified during runtime.
    Specify RACF user name (Optional; '?' for help):
    No value entered
    ```

12. Specify 3 for the trace level for IMS Connector for Java and press **Enter**:

    ```
    Specify trace level for IMS Connector for Java from 0 to 3 (default is 0; '?' for help): 3
    >> You entered trace level 3
    ```

13. The instance servlet is generated and compiled in the output directory that is specified:

    ```
    Generating servlet......completed
    Servlet is being compiled.......completed.
    ```

14. Type `Y` to save your input values for later execution in batch mode and press **Enter**:

    ```
    Do you wish to save your input values to a batch file (RACF information will NOT be saved)?
    (y|n ; Default is no)y
    Writing batch file to C:\MFSXMLUtilityGA\PHONEBOOK\PHONEBOOK_step2.txt...

    Batch file created
    ```

15. The deployment descriptor and web.xml files are generated:

    ```
    Generating servlet deployment descriptor......generated.

    Starts to put files in the WEB-INF directory...
    Compile servlet to be packaged into WAR file....
    Servlet is being compiled.......completed.
    ```

```
The following servlet file was generated:
 C:\MFSXMLUtilityGA\PHONEBOOK\PHONEBOOK.java
The following servlet class file was generated:
 C:\MFSXMLUtilityGA\PHONEBOOK\PHONEBOOK.class
The following batch file was generated:
 C:\MFSXMLUtilityGA\PHONEBOOK\PHONEBOOK_step2.txt
The following segment of web.xml file was generated:
 C:\MFSXMLUtilityGA\PHONEBOOK\PHONEBOOKWeb.xml
Step 2 completed.
```

Step 3: Generating an Application WAR file

In step 3 you generate PB.war file.

To generate a WAR file:

1. From the MFS XML Utility window, choose selection 3 and press **Enter**:

```
Enter your selection here: 3
Step 3: Generate WAR (Web Application aRchive) file containing one or more instance servlets
        In order to generate WAR file, you must first complete step 2.
        This step requires the following information:
        -Previously generated instance servlet class file(s) in the WEB-INF\classes directory
        -Previously generated deployment descriptor (web.xml) in the WEB-INF directory

        *Examine the content of the web.xml file in C:\MFSXMLUtilityGA\WEB-INF\ and make
         any necessary additions.*
        *Note that the only web.xml file that will be packaged into the WAR file is in
         C:\MFSXMLUtilityGA\WEB-INF\

This WAR file will be generated with the following instance servlets.
1) .\WEB-INF\classes\PHONEBOOK.class
2) .\WEB-INF\classes\PHONEBOOK.java
```

2. Enter PB for the name of your WAR file and press **Enter**:

```
Enter the name of this WAR file: PB
```

3. Indicate if you would like to include additional files in your WAR file, for example GIF or JPG files, (default is no) and press **Enter**:

```
Do you want to package additional files such as pictures with this WAR file? (y|n; default is no)
```

4. The WAR file is generated:

```
Adding manifest
Adding⌂WEB-INF/ (reading=0)(writing=0)(saving 0%)
Adding⌂WEB-INF/classes/ (reading=0)(writing=0)(saving 0%)
Adding⌂WEB-INF/classes/PHONEBOOK.class (reading=379)(writing=270)(saving 28%)
Adding⌂WEB-INF/classes/PHONEBOOK.java (reading=614)(writing=404)(saving 34%)
Adding⌂WEB-INF/web.xml (reading=1060)(writing=373)(saving 64%)
WAR file generated.

The following WAR file was generated:
 C:\MFSXMLUtilityGA\WAR\PB.war
Step 3 completed.
```

Step 4: Configure MFS Web Enablement support on WebSphere Application Server

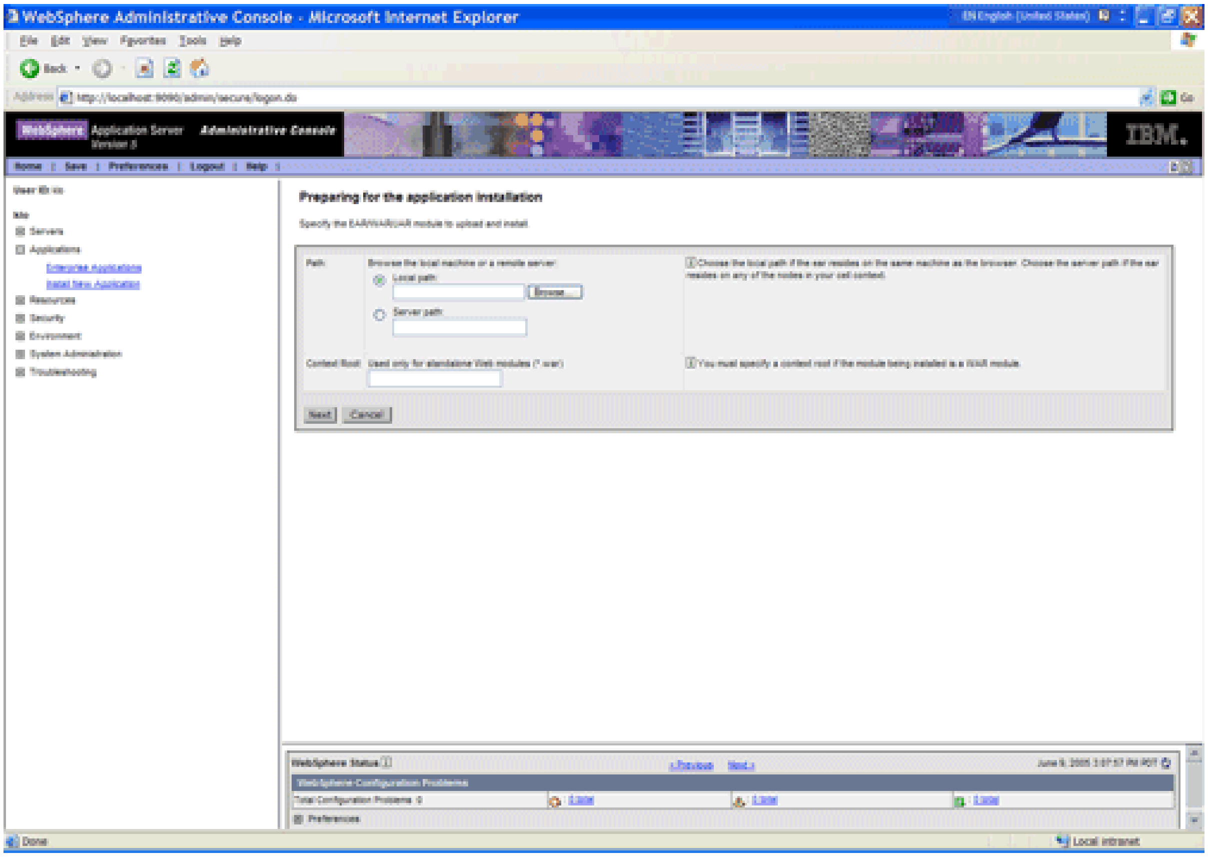
Refer to “Configuring the WebSphere Application Server resource adapter” on page 9 to configure MFS Web Enablement support on WebSphere Application Server.

Note: It is important that the WebSphere Application Server administrator saves all changes and restarts WebSphere Application Server whenever making changes to the IMS resource adapter.

Step 5: Deploy the application WAR file on WebSphere Application Server

To deploy the PB.war file onto WebSphere Application Server:

1. With WebSphere Application Server started, go to your WebSphere Application Server Administrative Console and expand **Applications** from the left-hand side and then click **Install New Application**.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://localhost:9090/admin/secure/logon.do
Go
WebSphere Application Server Version 5
Administrative Console
IBM.
Home | Save | Preferences | Logout | Help
Servers
Applications
Enterprise Applications
Install New Application
Resources
Security
Environment
System Administration
Troubleshooting
Browse...
Next
Cancel
WebSphere Status
Previous
Next
WebSphere Configuration Problems
Preferences
Done
Local intranet

Preparing for the application installation

Specify the EAR/WAR/JAR module to upload and install.

Path: Browse the local machine or a remote server:

● Local path:

○ Server path:

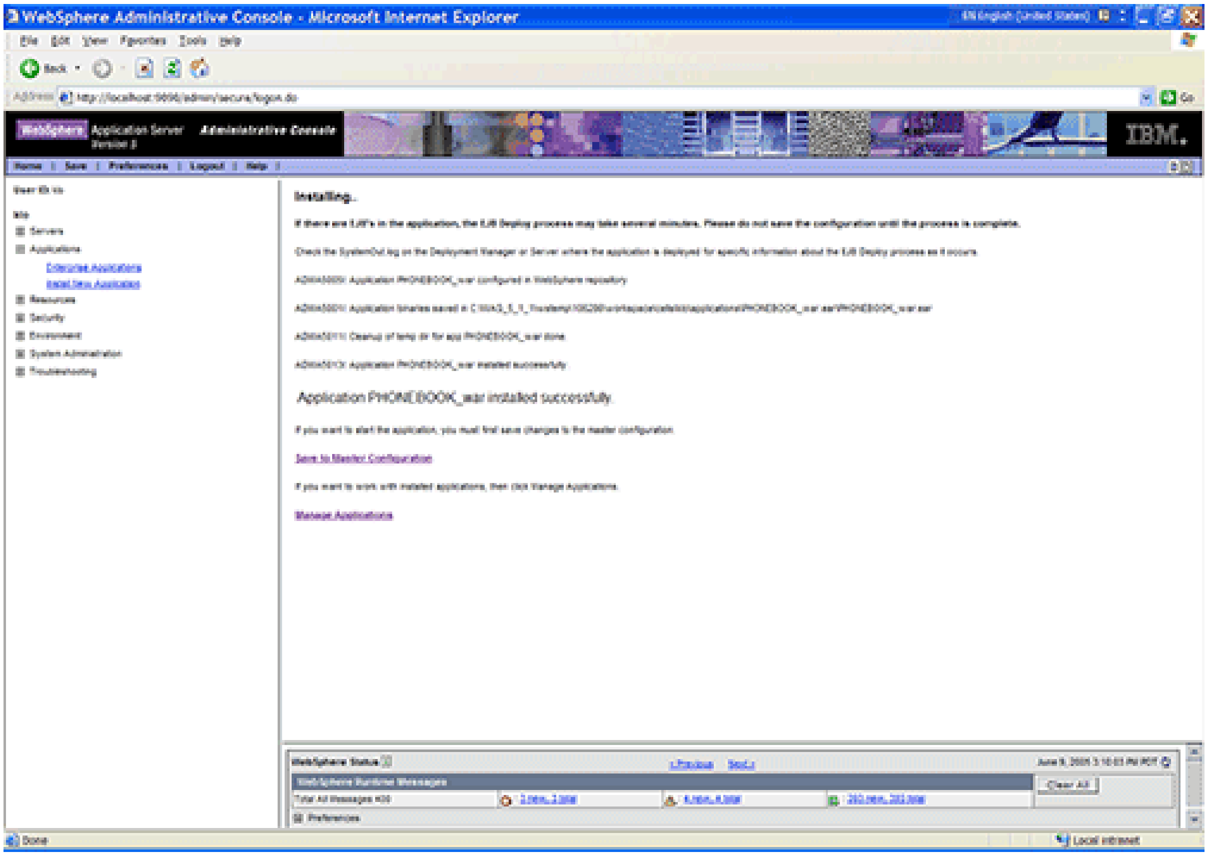
Choose the local path if the ear resides on the same machine as the browser. Choose the server path if the ear resides on any of the nodes in your cell context.

Context Root: Used only for standalone Web modules (*.war)

You must specify a context root if the module being installed is a WAR module.

Total Configuration Problems: 0

2. Click **Browse** under **Local path** to select the PB.war file that was generated in Step 3.
3. Enter the **Context Root** (for example: demo). The text you enter here will be a part of the URL.
4. Click **Next** to go to the next screen, keep the default values.
5. Click **Next** to go to the **Application Security Warnings** window.
6. Click **Continue** to go to the **Install New Application** window, keep the default values.
7. Click **Next** to go through the next three screens.
8. Click **Finish** and you should receive the message `Application PB_war installed successfully`.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://localhost:9090/admin/secure/logon.do
Go
WebSphere Application Server Version 5
Administrative Console
IBM.
Home | Save | Preferences | Logout | Help |
Servers
Applications
Enterprise Applications
Install New Application
Resources
Security
Environment
System Administration
Troubleshooting
WebSphere Status
< Previous Next >
WebSphere Runtime Messages
Clear All
Preferences
Done
Local intranet

Installing...

If there are EJB's in the application, the EJB Deploy process may take several minutes. Please do not save the configuration until the process is complete.

Check the SystemOut.log on the Deployment Manager or Server where the application is deployed for specific information about the EJB Deploy process as it occurs.

ADMA5005I: Application PHONEBOOK_war configured in WebSphere repository

ADMA5001I: Application binaries saved in C:\WAS_5_1_Toolkit\temp\[illegible]\applications\PHONEBOOK_war.ear\PHONEBOOK_war.ear

ADMA5011I: Cleanup of temp dir for app PHONEBOOK_war done.

ADMA5013I: Application PHONEBOOK_war installed successfully.

Application PHONEBOOK_war installed successfully.

If you want to start the application, you must first save changes to the master configuration.

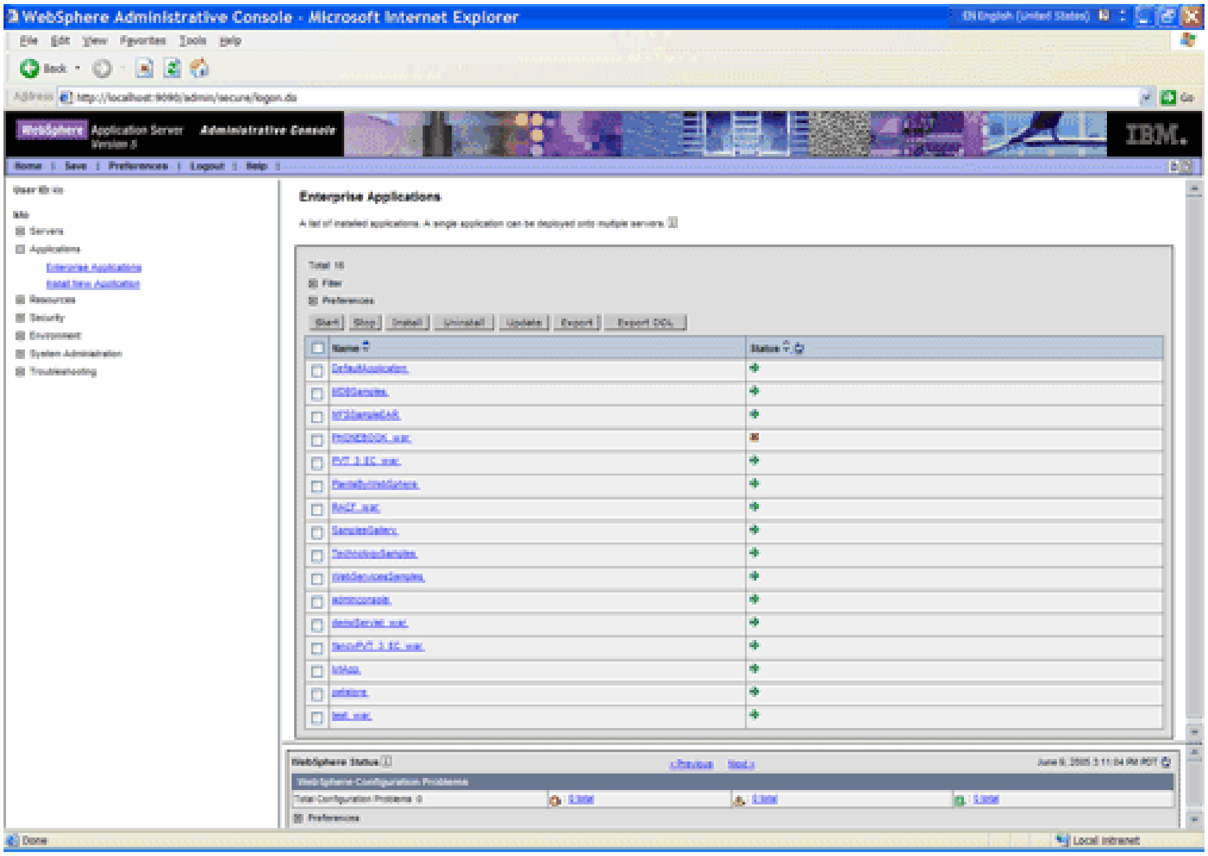
Save to Master Configuration

If you want to work with installed applications, then click Manage Applications.

Manage Applications

9. Click the **Save** button.

10. Start the application by selecting **Enterprise Applications**.



Enterprise Applications

A list of installed applications. A single application can be deployed onto multiple servers.

Total 16

□	Name	Status
□	DefaultApplication	➜
□	MDBSamples	➜
□	MTSSampleEAR	➜
□	PHONEBOOK_war	✖
□	PVT_3_EC_war	➜
□	PlantsByWebSphere	➜
□	RACF_war	➜
□	SamplesGallery	➜
□	TechnologySamples	➜
□	WebServicesSamples	➜
□	adminconsole	➜
□	demoServlet_war	➜
□	SecurePVT_3_EC_war	➜
□	ivtApp	➜
□	pdtdocs	➜
□	test_war	➜

WebSphere Status

WebSphere Configuration Problems

Total Configuration Problems: 0

WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:9090/admin/secure/logon.do
WebSphere Application Server Version 6 Administrative Console
Home | Save | Preferences | Logout | Help
Servers
Applications
Enterprise Applications
Install New Application
Resources
Security
Environment
System Administration
Troubleshooting
Filter
Preferences
Start
Stop
Install
Uninstall
Update
Export
Export DDL
< Previous
Next >
June 6, 2005 3:11:04 PM PDT
Done
Local intranet

11. Select the check box of the Web application archive (WAR) file that you just installed and click the **Start** button.

12. You should receive the message `Application PB_war on server server 1 and node [node name] started successfully` and the application status should become green.

Step 6: Invoke the instance servlet

- With WebSphere Application Server started, open a Web browser and enter the URL `http://localhost:9080/demo/PHONEBOOK`.

  Note: The `localhost:9080` is the WebSphere Application Server node, `demo` is the context root you entered during the Web application archive (WAR) file, and `PHONEBOOK` is the instance servlet that is generated in step 2.

Step 7: Invoke PHONEBOOK

To invoke the PHONEBOOK application sample:

1. Enter /FOR IVTNO and click the **Submit** button
2. Enter `DISPLAY` in PROCESS CODE field.
3. Enter `LAST1` in LAST NAME field.
4. Click the **Submit** button.
5. Verify the output is correct.

Step 8: Logout

To logout:

Click the **Logout** button.

Note: For improved performance, you should always logout when finished using the MFS-based IMS transactions. Logging out releases objects held in memory immediately.

Chapter 8. Troubleshooting

This section provides troubleshooting information for MFS Web Enablement.

The following topics provide additional information:

- "Messages and Codes for MFS Web Enablement"
- "Logging and tracing in MFS Web Enablement" on page 53

Messages and Codes for MFS Web Enablement

You might encounter the following possible types of messages and codes when you use IMS MFS Web Enablement:

MFS XML Utility messages and codes
The error messages that occur during the parsing of MFS source files and the generating of instance servlets. For more information on MFS XML Utility messages and codes, see the *MFS XML Utility Guide and Reference*.

IMS MFS Web Enablement messages and codes
The error messages that occur within IMS MFS Web Enablement. The error messages the occur within the execution of transforming and HTTP request and response to and from a byte array. An error can occur, for example, during the processing of an MFS Servlet. The error message is displayed using system default DFSMO2.xmi.

For more information on the MFS Web Enablement messages and codes, see "MFS Servlet messages" and "MFS Adapter messages" on page 49. See "Logging and tracing in MFS Web Enablement" on page 53 for information about how to configure the logging of detailed traces into the log file.

IMS Connector for Java and IMS Connect messages and codes
When an error occurs sending or receiving a byte array, IMS Connector for Java returns ICO exceptions for internal errors or HWS exceptions for IMS Connect errors. The error message is displayed using system default DFSMO2.xmi.

For more information on the IMS Connect messages, see *IMS Messages and Codes, Volume 1*. For more information on the IMS Connector for Java messages, see the IMS Connector for Java documentation at http://www.ibm.com/software/data/db2imstools/imstools-library.html.

IMS DFS messages
IMS might report an error situation when running the IMS transaction by sending a DFS message (messages with the "DFS" prefix). The messages are displayed with the system default DFSMO1.xmi (in the case of single segment data) or DFSMO5.xmi (in the case of multiple segment data). For more information on DFS messages, see the *IMS Messages and Codes, Volume 2*.

MFS Servlet messages

The topic documents the messages that are issued by MFS Servlet.

IXFS001E

`Unable to load resource from the specified path: URI`

Explanation

Unable to find the XMI file in the folder which was specified in the setup.

System action

An error message is displayed.

User response

Place the XMI file in the specified path or folder, and rerun the step.

IXFS002E

```
XSLT transformation failure. Please see log for more detail.
```

Explanation

Unable to format the HTML page because of errors in the style sheet, or unable to locate the style sheet.

System action

An error message is displayed.

User response

Modify the style sheet or verify the location of the style sheet.

IXFS003E

```
Input does not contain valid request
```

Explanation

Invalid module name or transaction code was entered.

System action

The system default DFSMO2 page is displayed.

User response

Reissue the format request or transaction code.

IXFS004E

```
Invalid PFKey defined
```

Explanation

Literal values defined for the PFKey are not supported. See Chapter 1, “Overview of MFS Web enablement version 9.3,” on page 1 for the supported PF key literals.

System action

The system default DFSMO2 page is displayed.

User response

Do not continue using this PFKey. Reissue the format request or transaction code to restart.

IXFS005E

DFS291 *hh:mm:ss* INPUT MUST BEGIN FROM FIRST PHYSICAL PAGE

Explanation

Input data was entered from somewhere other than the first physical page with multiple physical page input specified. The input data is not submitted.

System action

The input data is ignored.

User response

Reissue the format request or transaction code to start over again. Reenter the data, starting from the first physical page.

MFS Adapter messages

This topic documents the messages that are issued by the MFS Adapter.

IXFT001E

`Unsupported message option 3 used in resource name.`

Explanation

Message option 3 is not supported.

System action

The system default DFSMO2 page is displayed.

User response

Do not continue using this transaction. Contact the system administrator for a supported MFS metadata XMI file.

IXFT002E

`Unsupported MFS bypass appMapName`

Explanation

The output data cannot be displayed properly without loading an MFS metadata XMI file. The transaction cannot be continued.

System action

The system default DFSMO2 page is displayed.

User response

Do not continue using this transaction. Contact the system administrator for a supported MFS metadata XMI file.

IXFT003E

`Invalid XMI file` *`[fileName]`* `contains unresolved mapping in the message fields to device fields.`

Explanation

The MFS Adapter detects dangling unreferenced device fields. The specified XMI file is invalid.

System action

The system default DFSMO2 page displays.

User response

Regenerate the XMI file using version 9.3 of the MFS XML Utility.

MFS Importer messages

IXFI001E

`The copy file was not found.`

Explanation

An MFS file required by a COPY statement was not found in the specified directory.

System action

The message is issued and the parser is stopped.

User response

Copy the missing MFS file into the specified directory.

IXFI002W

`Missing stack.`

Explanation

An attempt was made to perform UNSTACK from a non-existent STACK ID.

System action

The message is issued and execution continues.

User response

Correct the MFS source and restart the importer.

IXFI003W

`The device characteristics file could not be opened.`

Explanation

The importer could not open the device characteristics file for reading.

System action

The message is issued and execution continues.

User response

Make sure that the device characteristic file exists and has the correct file access mode.

IXFI004W

```
The device characteristics file was invalid.
```

Explanation

An I/O exception occurred when the device characteristics file was read.

System action

The message is issued and execution continues.

User response

Make sure that the contents of the device characteristic file are correct and in binary format.

IXFI005W

```
The external URI is invalid.
```

Explanation

A midname, modname, or table name was expected while an external reference for the NXT or OCT parameter was loading. The midname, modname, or table name are missing.

System action

The message is issued and the MFS Importer generates an empty default reference for the midname, modname, or table name to resolve the relationship. Execution continues.

User response

A forward reference in the MFS source files can produce this message. This error occurs when the information needed to complete the parsing of the MFS source resides in another file. Ensure that the associated XMI file for the MID, MOD, or TABLE name in the warning is produced and fully populated (not empty).

IXFI006W

```
A default object was generated.
```

Explanation

This error refers to an unresolved relationship. This error occurs when the parser needs to generate an empty XMI file so that cross-XMI file relationships can be set correctly.

System action

The message is issued and execution continues.

User response

Correct the error in the MFS source and restart the MFS Importer.

IXFI007W

```
An unresolved relationship occurred.
```

Explanation

This error refers to an unresolved relationship. This error occurs when the information provided in the source file is incorrect. For example, when an MFLD references a non-existent or invalid DFLD.

System action

The message is issued and execution continues.

User response

Correct the error in the MFS source and restart the MFS Importer.

IXFI008W

```
The device is unsupported.
```

Explanation

A non-3270 device type was found in the source.

System action

The message is issued and processing continues.

User response

None.

IXFI009E

```
The encoding is unsupported.
```

Explanation

An unsupported encoding was specified.

System action

The message is issued and the MFS Importer stops.

User response

Choose a supported encoding from the drop-down list.

IXFI010W

```
The parser overwrote an XMI file.
```

Explanation

The MFS source contained a definition for a MID/MOD or TABLE statement that was already defined in the XMI repository. The new definition will overwrite the old definition.

System action

The message is issued and processing continues.

User response

None.

IXFI011E

```
Parse error: com.ibm.etools.mfs.importer.ParseException: Encountered "L" at line X, column Y.
Was expecting one of:
    "LPAGENO" ...
    "TIME" ...
    "DATE1" ...
    "YYDDD" ...
    "DATE2" ...
    "MMDDYY" ...
    "DATE3" ...
    "DDMMYY" ...
    "DATE4" ...
    "YYMMDD" ...
    "DATE1Y4" ...
    "YYYYDDD" ...
    "DATEJUL" ...
    "DATE2Y4" ...
    "MMDDYYYY" ...
    "DATEUSA" ...
    "DATE3Y4" ...
    "DDMMYYYY" ...
    "DATEEUR" ...
    "DATE4Y4" ...
```

Explanation

The MFS source contains an unsupported keyword or phase. For example, LTMSG, LTNAME, and LTSEQ.

System action

The message is issued, and the MFS Importer stops.

User response

Modify the MFS source to remove the unsupported keyword or phase.

Logging and tracing in MFS Web Enablement

The MFS Servlet and the MFS Adapter provide an option to log runtime messages into the server's trace file. The WebSphere Application Server administrator can enable tracing by following the instructions in the WebSphere Application Server logging and tracing configuration.

you can view the logging information in the trace log file using a text editor (such as Notepad). The trace log file records tracing events for user-selected applications. All MFS traces are recorded in the following format:

```
[timestamp] servlet_name  [session id] class.method Message: data
```

In the event trace:

timestamp
The time the event is logged.

servlet_name
The name of the instance servlet.

session id
The unique http session ID identifies the client browser that initiates the request.

class.method
The name of the class and method that logs the event.

data The trace data.

The administrator can specify different trace levels and combinations:

Debug
Detailed debugging of messages.

Entry + Exit
Method entrance and exiting indicator messages.

Event General event messages.

To avoid heavy logging, enable the debug-level trace logging only when necessary. The WebSphere Application Server administrator must keep track of and clean up the trace files.

For more information, see "Configuring logging and tracing for WebSphere Application Server Version 5.1.1" on page 13.

Chapter 9. Notices

This information was developed for products and services offered in the U.S.A.

IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. COPYRIGHT LICENSE: This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

(your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. Copyright IBM Corp. 2003, 2005. All rights reserved. Trademarks

Chapter 10. Trademarks

IBM, IMS, WebSphere, and z/OS are trademarks of International Business Machines Corporation in the United States, other countries, or both.

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

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