

IMS



# MFS Web Enablement Version 9.1.0 User's Guide and Reference



IMS



# MFS Web Enablement Version 9.1.0 User's Guide and Reference

**Note**

Before using this information and the product it supports, read the information in Notices at the end of this book.

**First Edition (July 2005)**

**© Copyright International Business Machines Corporation 2005. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

## Contents

<b>About this book</b> . . . . .	<b>v</b>	<b>Chapter 4. MFS Adapter</b> . . . . .	<b>33</b>
Accessibility . . . . .	v	<b>Chapter 5. Sample MFS style sheets</b> . . . . .	<b>35</b>
How to send your comments . . . . .	v	<b>Chapter 6. Security considerations for MFS Web Enablement</b> . . . . .	<b>39</b>
<b>Chapter 1. Overview of MFS Web enablement</b> . . . . .	<b>1</b>	Secure Sockets Layer support . . . . .	39
How does MFS Web Enablement work? . . . . .	4	WebSphere Application Server user authentication . . . . .	39
Prerequisites for MFS Web Enablement . . . . .	6	<b>Chapter 7. Sample instructions to web enable the IMS IVP Phonebook application</b> . . . . .	<b>41</b>
Limitations of MFS Web Enablement Version 9.1.0 . . . . .	7	Step 1: Parsing the MFS source file with MFS XML Utility . . . . .	41
<b>Chapter 2. Configuration</b> . . . . .	<b>9</b>	Step 2: Generating an instance servlet . . . . .	42
Configuring WebSphere Application Server . . . . .	9	Step 3: Generating a WAR file . . . . .	44
Configuring the WebSphere Application Server resource adapter . . . . .	9	Step 4: Configure MFS Web Enablement support on WebSphere Application Server . . . . .	44
Configuring SSL in WebSphere Application Server . . . . .	10	Step 5: Deploy the application WAR file on WebSphere Application Server . . . . .	44
Setting timeout in WebSphere Application Server . . . . .	11	Step 6: Invoke the instance servlet . . . . .	46
Setting timeout for an enterprise application . . . . .	12	Step 7: Invoke PHONEBOOK . . . . .	46
Configuring logging and tracing for WebSphere Application Server Version 5.1.1 . . . . .	13	Step 8: Logout . . . . .	46
Configuring logging and tracing for WebSphere Application Server Version 6.0 . . . . .	15	<b>Chapter 8. Troubleshooting</b> . . . . .	<b>47</b>
Configuring form-based authentication for MFS Web Enablement in WebSphere Application Server (optional) . . . . .	16	Messages and Codes for MFS Web Enablement . . . . .	47
Configuring a Web browser for MFS Web Enablement . . . . .	19	MFS Servlet messages . . . . .	47
Configuring a Microsoft Internet Explorer browser . . . . .	20	MFS Adapter messages . . . . .	49
Configuring a Mozilla Firefox Web browser . . . . .	23	MFS Importer messages . . . . .	50
<b>Chapter 3. MFS Servlet.</b> . . . . .	<b>27</b>	Logging and tracing in MFS Web Enablement . . . . .	53
Installing the instance servlet Web Application Archive (WAR) file . . . . .	28	<b>Notices</b> . . . . .	<b>55</b>
Invoking the deployed instance servlet . . . . .	30		



---

## About this book

---

### Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products. The major accessibility features in IMS MFS Web Enablement enable you to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Online information for IMS MFS Web Enablement is available in HTML, which is an accessible format. You can access all HTML functions by using a keyboard or keyboard shortcut keys. HTML also allows you to use screen readers and other assistive technologies.

---

### How to send your comments

Your feedback is important in helping us provide the most accurate and highest quality information. If you have any comments about this or any other IMS information, you can do one of the following:

- Go to the IMS Library page at [www.ibm.com/software/data/ims/library.html](http://www.ibm.com/software/data/ims/library.html) and click the Library Feedback link, where you can enter and submit comments.
- Send your comments by e-mail to [imspubs@us.ibm.com](mailto:imspubs@us.ibm.com). Be sure to include the title, the part number of the title, the version of IMS, and, if applicable, the specific location of the text you are commenting on (for example, a page number in the PDF or a heading in the Information Center).





---

## Chapter 1. Overview of MFS Web enablement

IMS MFS Web Enablement Version 9.1.0 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 distributed platforms (Microsoft Windows)

IBM WebSphere Application Server distributed platforms 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 distributed platforms Information Center at [www.ibm.com/software/webservers/appserv/was/library/index.html](http://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 27.

### 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 27.

### **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 33.

### **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.

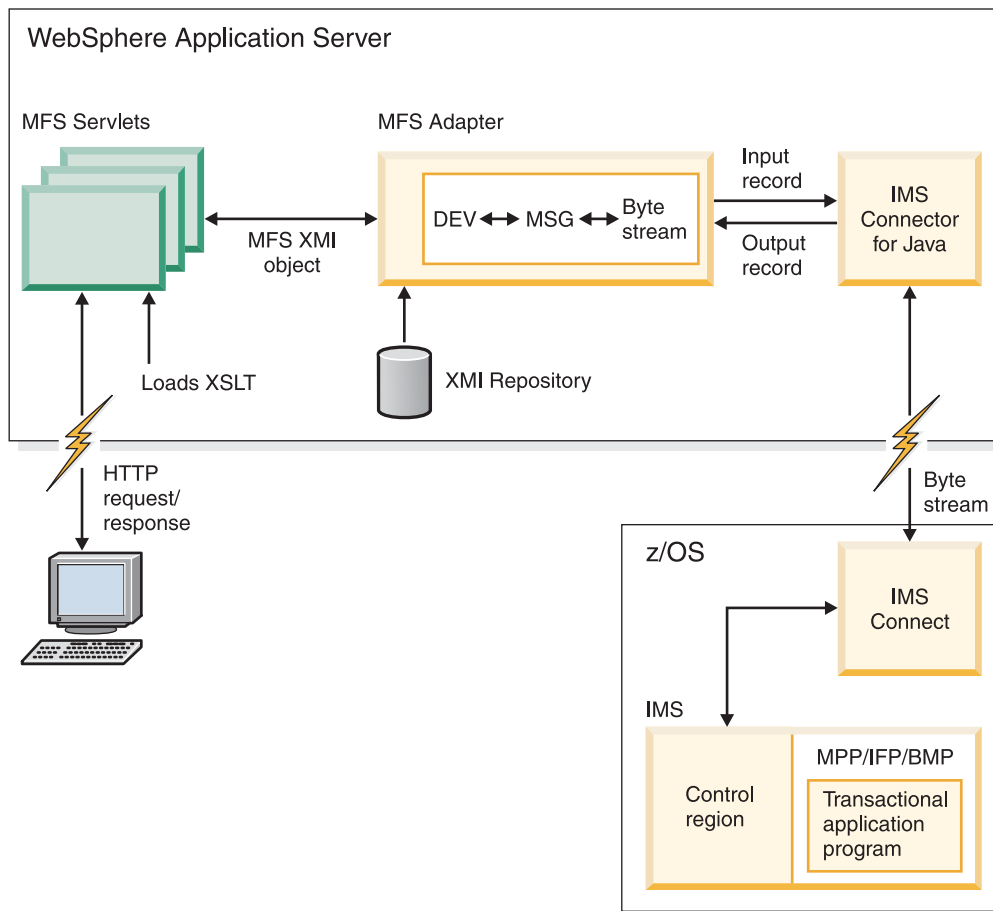
For more information, see the *IMS Connector for Java* documentation at <http://www.ibm.com/software/data/db2imstools/imstools-library.html>.

### **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 ENDMPPPI (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 35.

### **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**

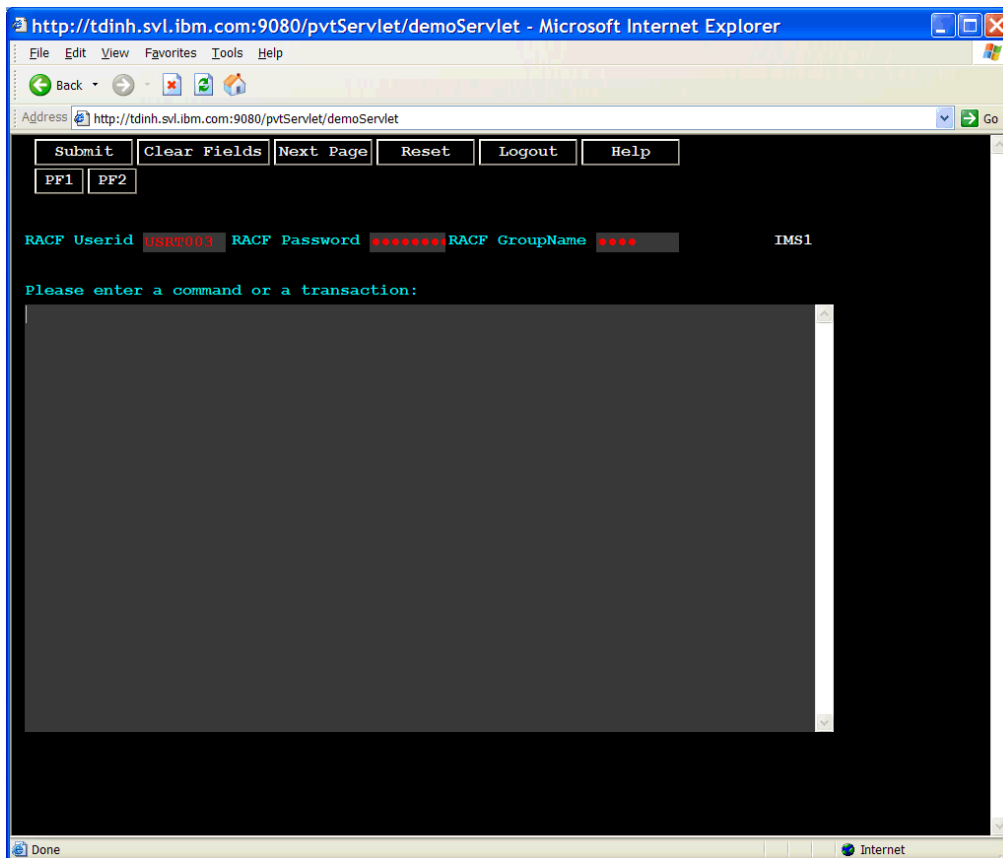
WebSphere Application Server user authentication can be optionally configured so that when clients access a particular servlet for the first time, a login mechanism is activated. See “WebSphere Application Server user authentication” on page 39 for more information about the sample login mechanism.

---

## **How does MFS Web Enablement work?**

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:



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 user ID, 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 duration of the session. The session is terminated when logged out, browser closed, or timed 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 will be converted to uppercase text by the MFS Adapter. The data will be remain unchanged (mixed cased allowed). The output execution will follow 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 the message:

- 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 DFSM02 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 DFSM03.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 DFSM02.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**

IMS MFS Web Enablement Version 9.1.0 is supported on the distributed platforms for Microsoft Windows 2000 and Microsoft Windows XP. The target runtime server is WebSphere Application Server distributed platforms (Windows only) Version 5.1.1 and Version 6.0 with the latest fixpacks installed.

The following products and tools are required in order to use MFS Web Enablement Version 9.1.0:

- IMS Version 9.1
- IMS Connect Version 9.1.0.1
- IBM WebSphere Application Server distributed platforms (Windows only) Version 5.1.1 and Version 6.0 with the latest fixpacks applied (511 requires both Cumulative fix and Cumulative fix for SDKs)
- IMS Connector for Java Version 9.1.0.1.1
- One of the following Web browsers:
  - Microsoft Internet Explorer Version 6 and later
  - Mozilla Firefox Version 1.0.1 or later

---

## Limitations of MFS Web Enablement Version 9.1.0

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

- The **Back** button and **Refresh** button on the browser are not supported. If you click the **Back** button, the same page will be displayed. If you click the **Refresh** button, the Session ID Error page will be displayed.
- 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 19

---

### 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:

- 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 machine, 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 MFSTDTLang.jar into this new folder.

**Note:** The WAS\_INSTALL\_ROOT directory can be looked up 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 above the IMS resource adapter entry and add each entry on a new line.

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

The following graphic shows the MFS JARS files residing in the WAS\_INSTALL\_ROOT/mfsweb directory, where WAS\_INSTALL\_ROOT is the root install directory of WebSphere Application Server.

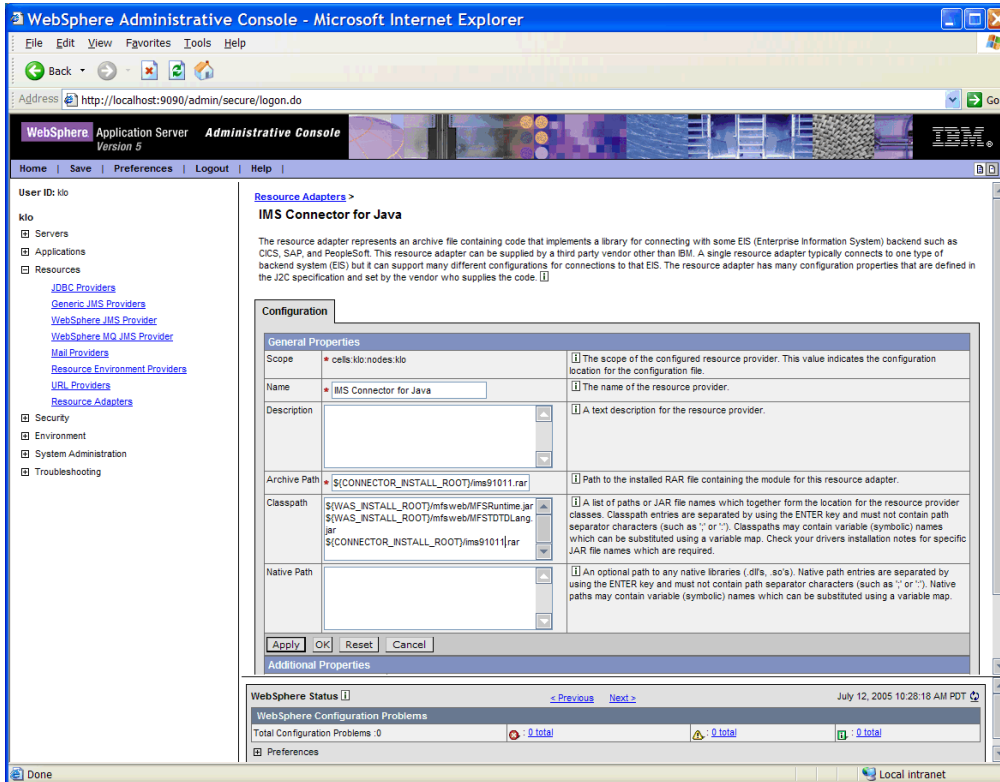


Figure 1. MFS JAR files in the WAS\_INSTALL\_ROOT/mfsweb directory

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 22 or “Enabling SSL for Mozilla Firefox (optional)” on page 25.

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:

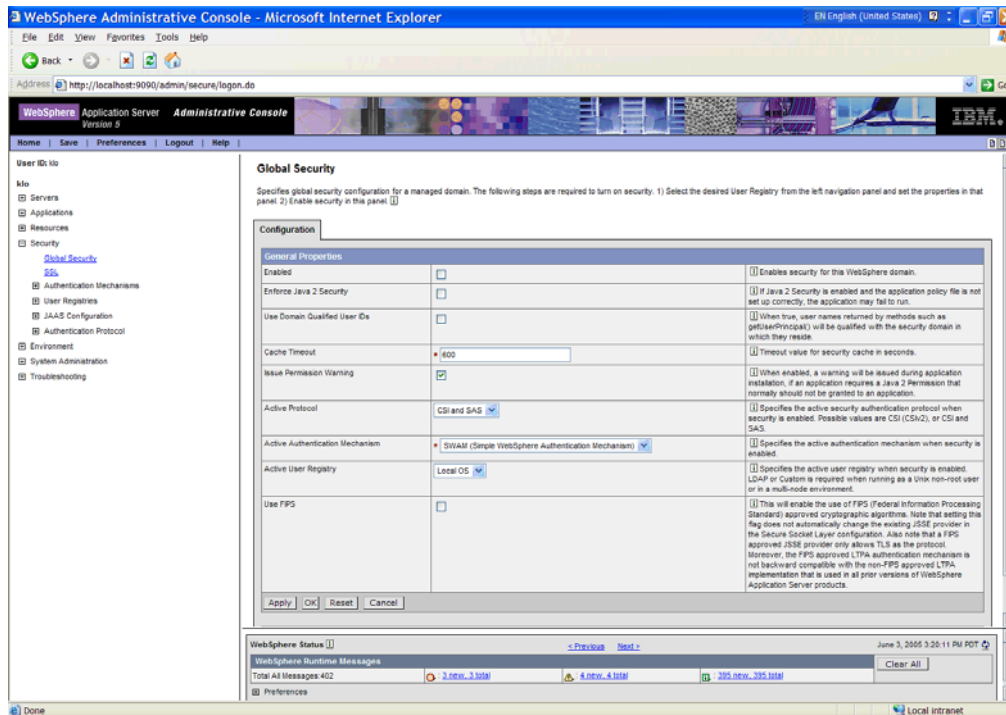


Figure 2. Configuring SSL in the WebSphere Application Server Administrative Console

**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](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:

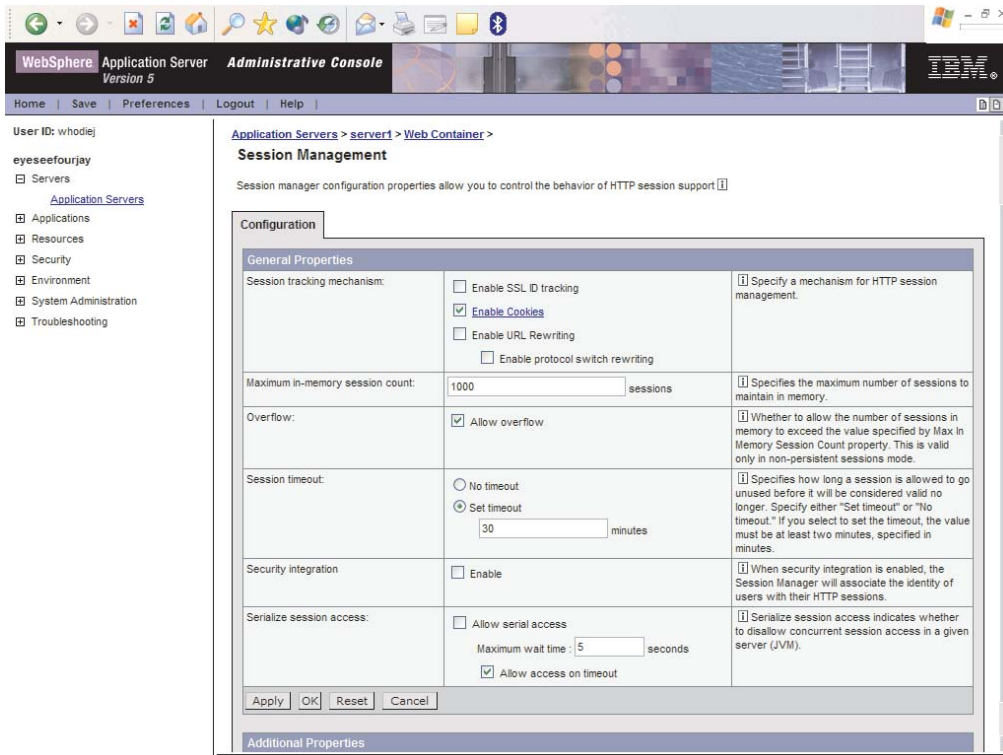


Figure 3. Setting timeout in the WebSphere Application Server

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:

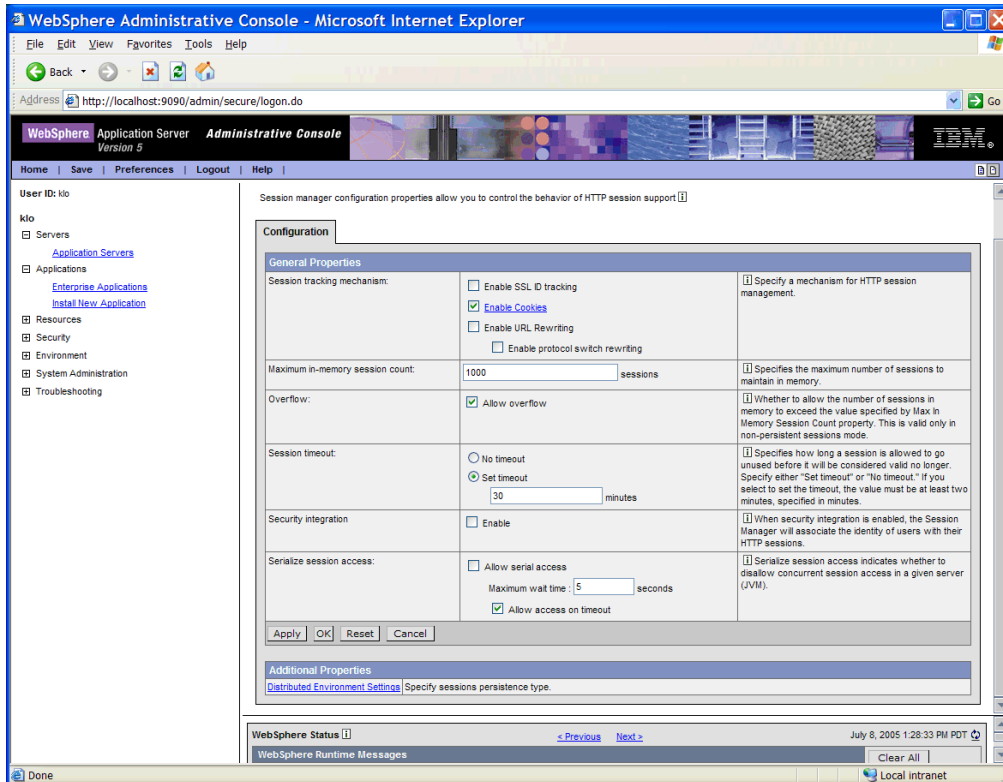


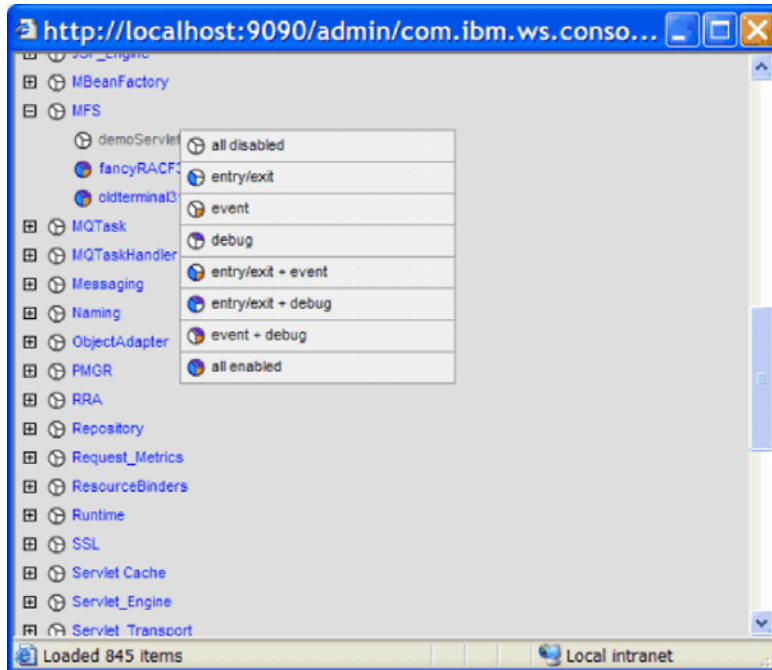
Figure 4. Setting timeout for an enterprise application

## 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**.

- In the **Diagnostic Trace Service** window, modify the Trace Specification by clicking **Modify...** This opens a window that lists all of the groups.



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

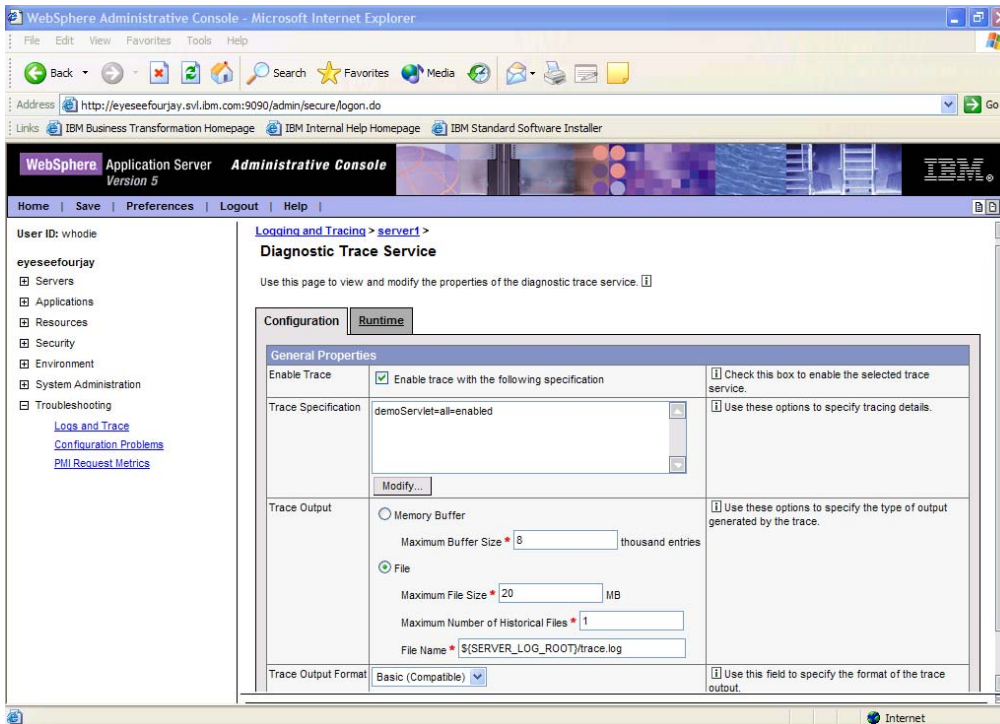


Figure 5. Trace Specification box

- Click **Save** in the Messages box.
- Click **Save**.

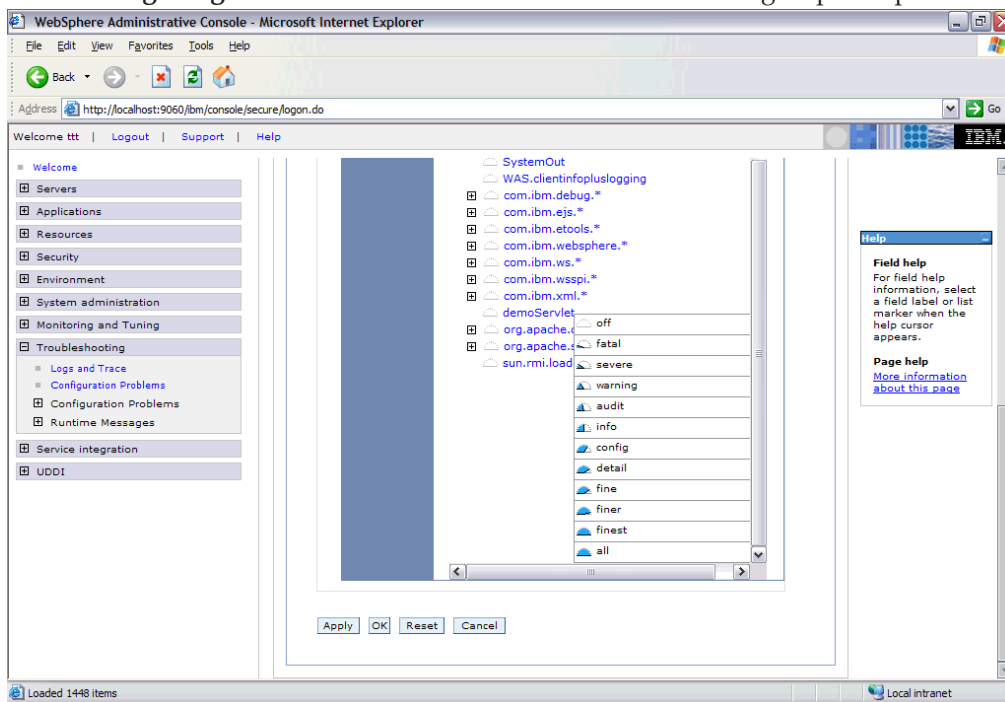
## 9. Restart WebSphere Application Server.

For more information about logging and tracing, see [Logging and tracing](#).

# Configuring logging and tracing for WebSphere Application Server Version 6.0

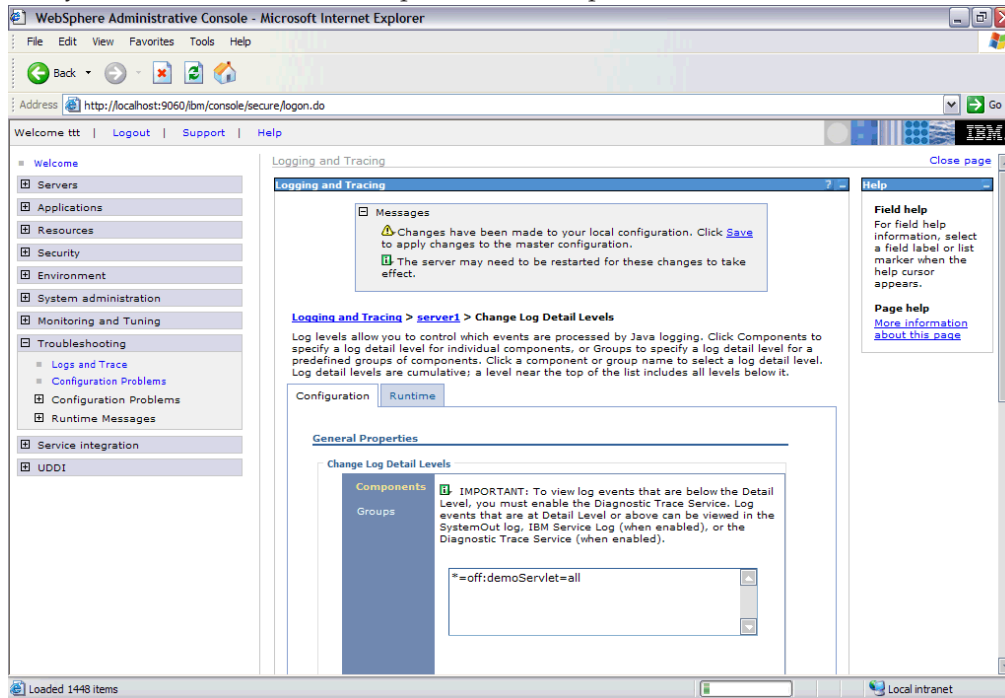
To configure logging and tracing for WebSphere Application Server version 6.0, complete the following steps:

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:



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 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>ecdv192.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.xml</param-value>
</init-param>
<init-param>
<param-name>traceLevel</param-name>
<param-value>3</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” on page 20
- “Configuring a Mozilla Firefox Web browser” on page 23

## 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 22

### 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**:

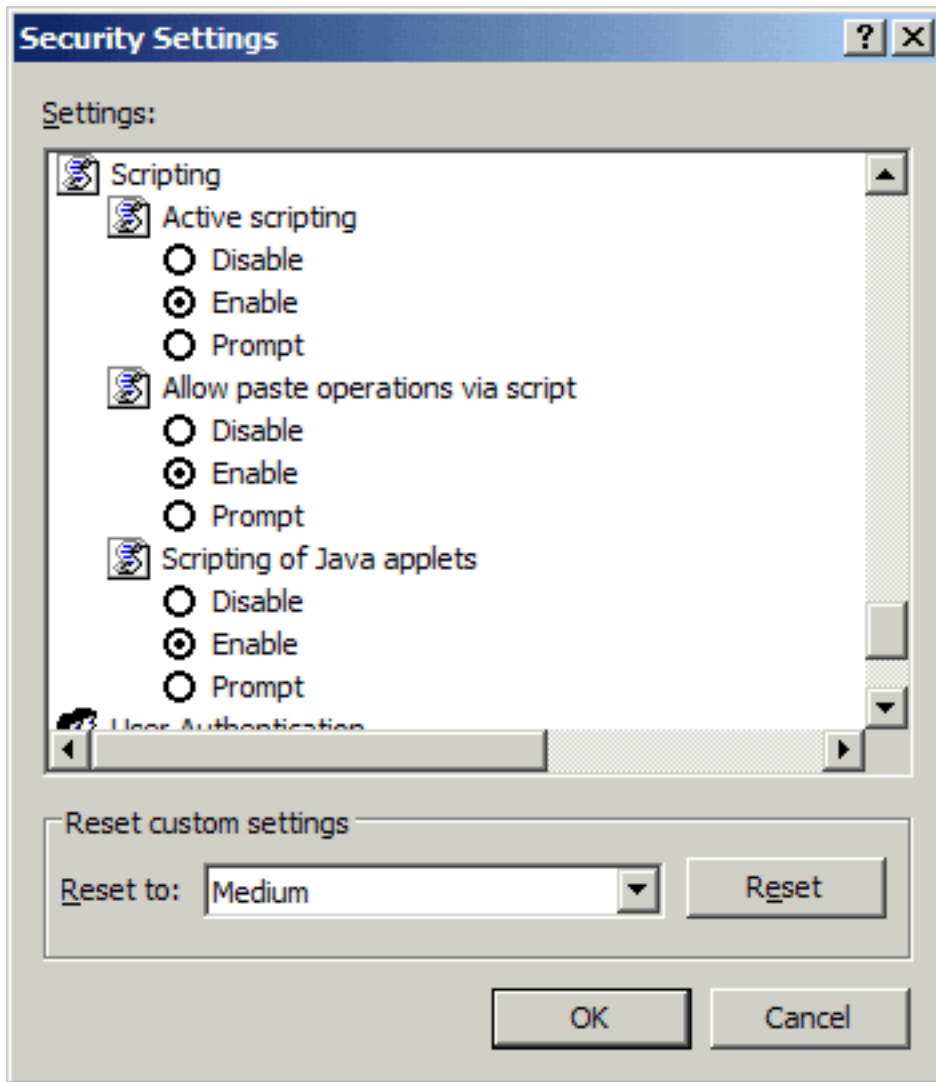


Figure 6. Enabling active scripting

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**.

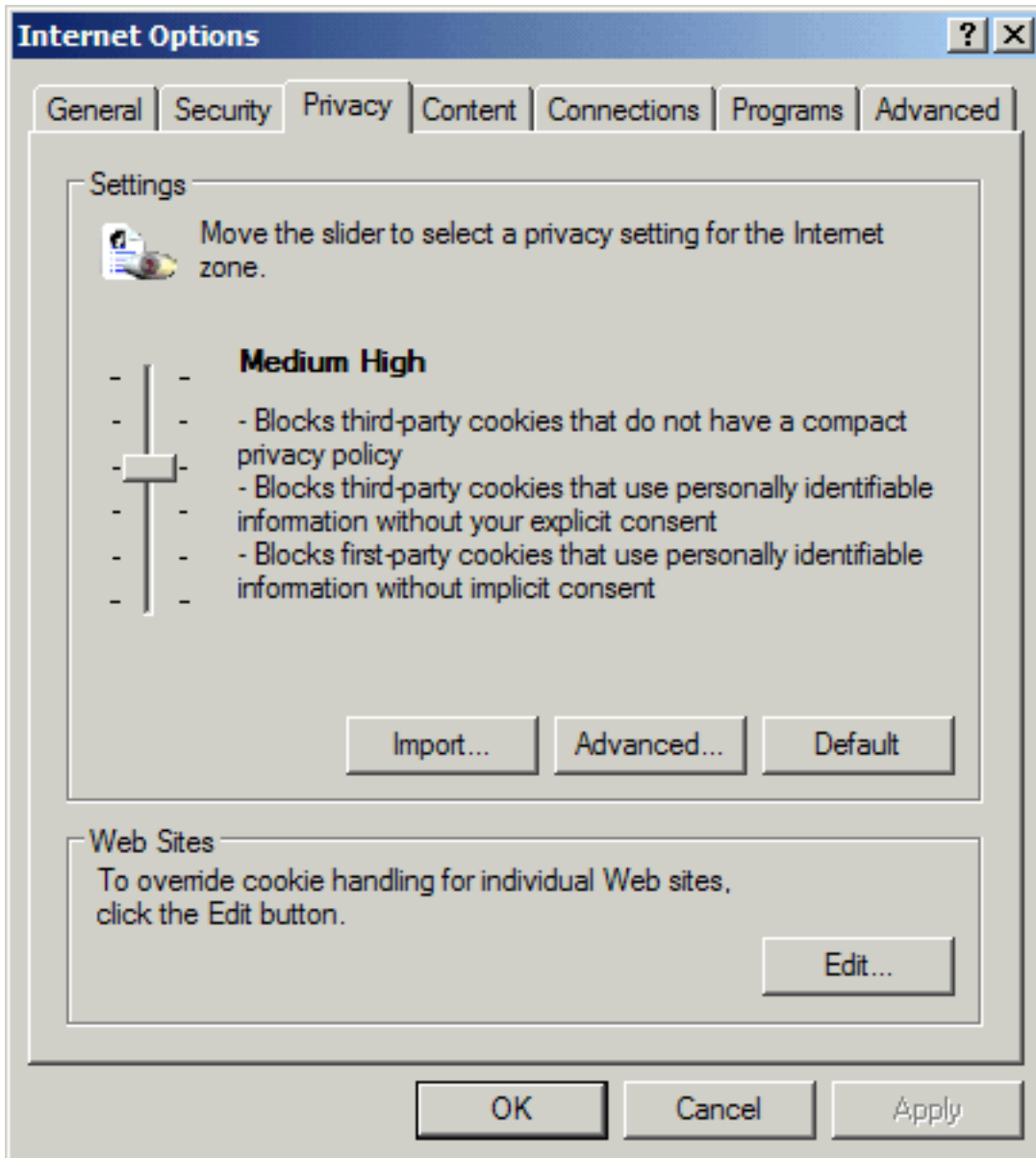


Figure 7. Microsoft Internet Explorer privacy settings

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.

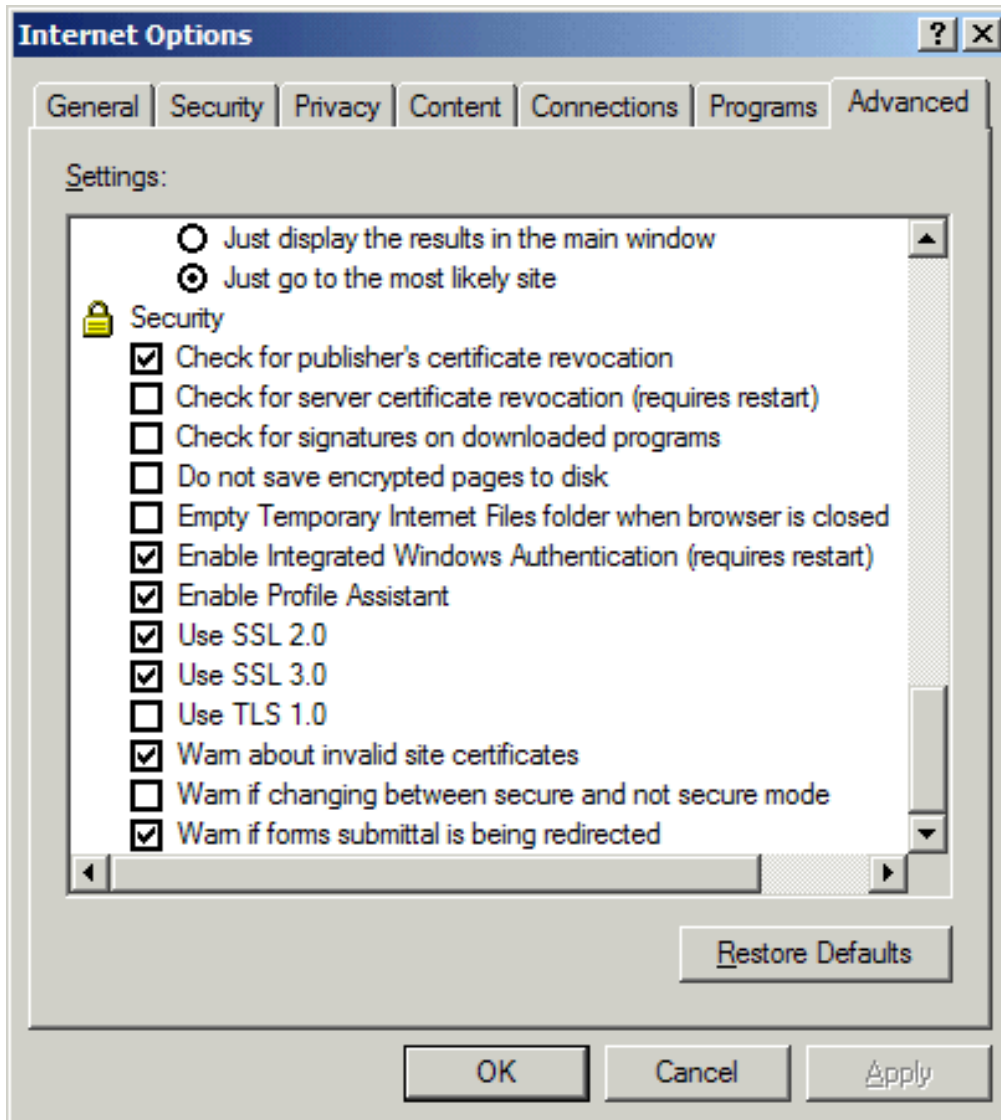


Figure 8. Enabling SSL

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 24
- “Enabling SSL for Mozilla Firefox (optional)” on page 25

### 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.

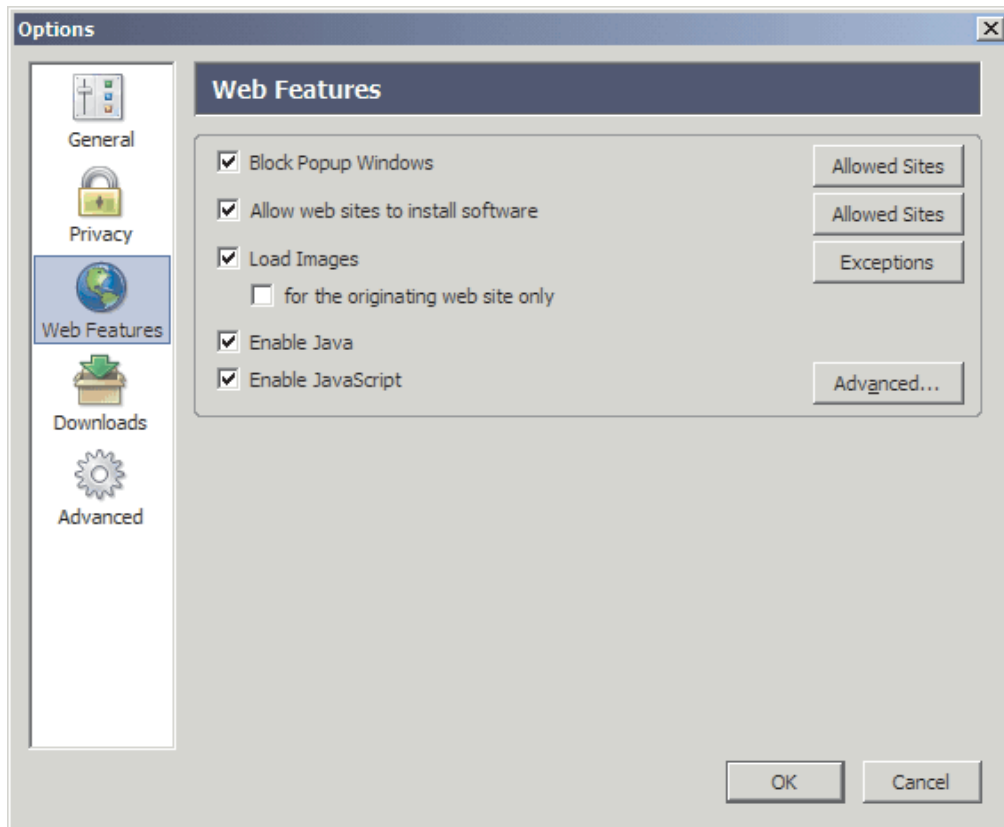


Figure 9. Enabling Java and JavaScript

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.



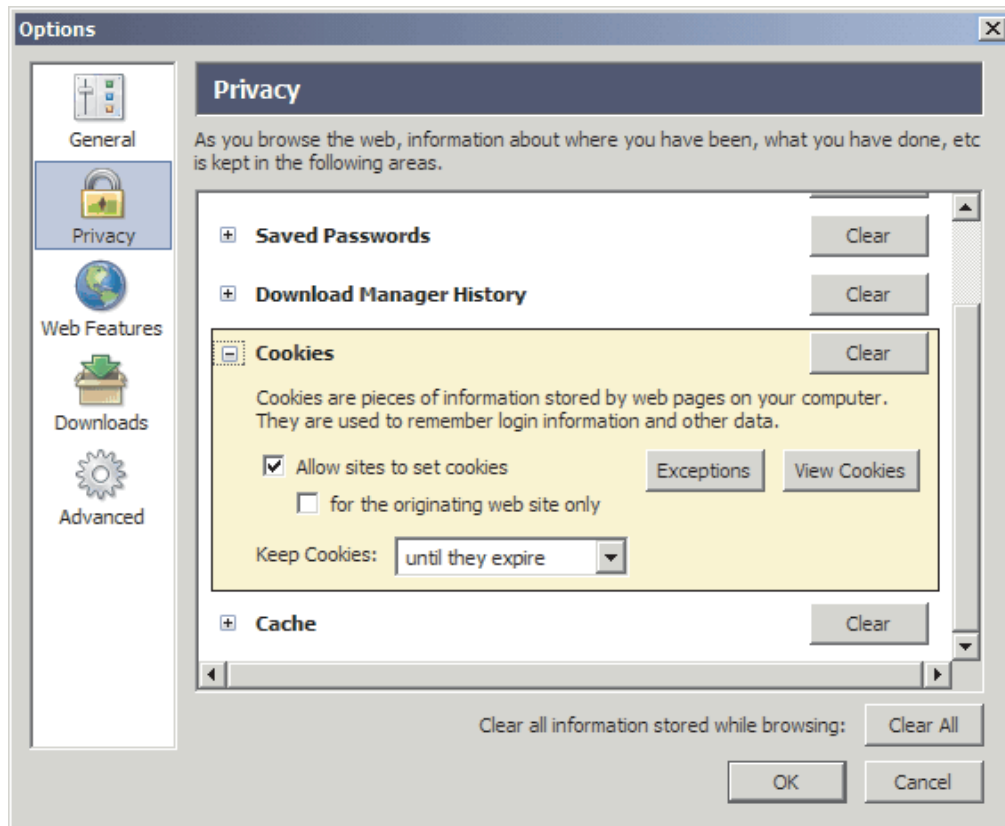


Figure 10. Enabling cookies

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.

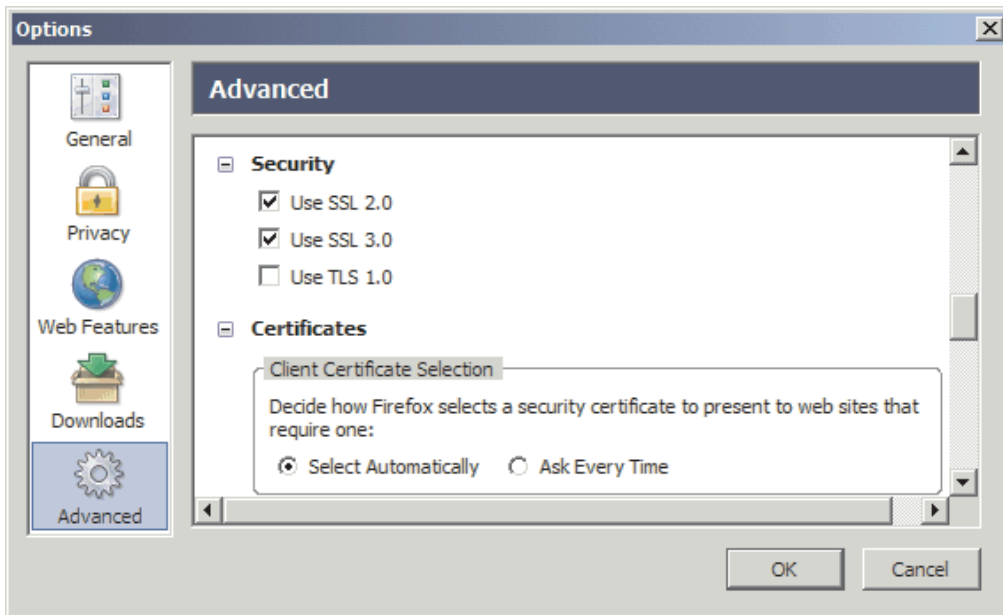


Figure 11. Enabling SSL

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 the user to a blank page.

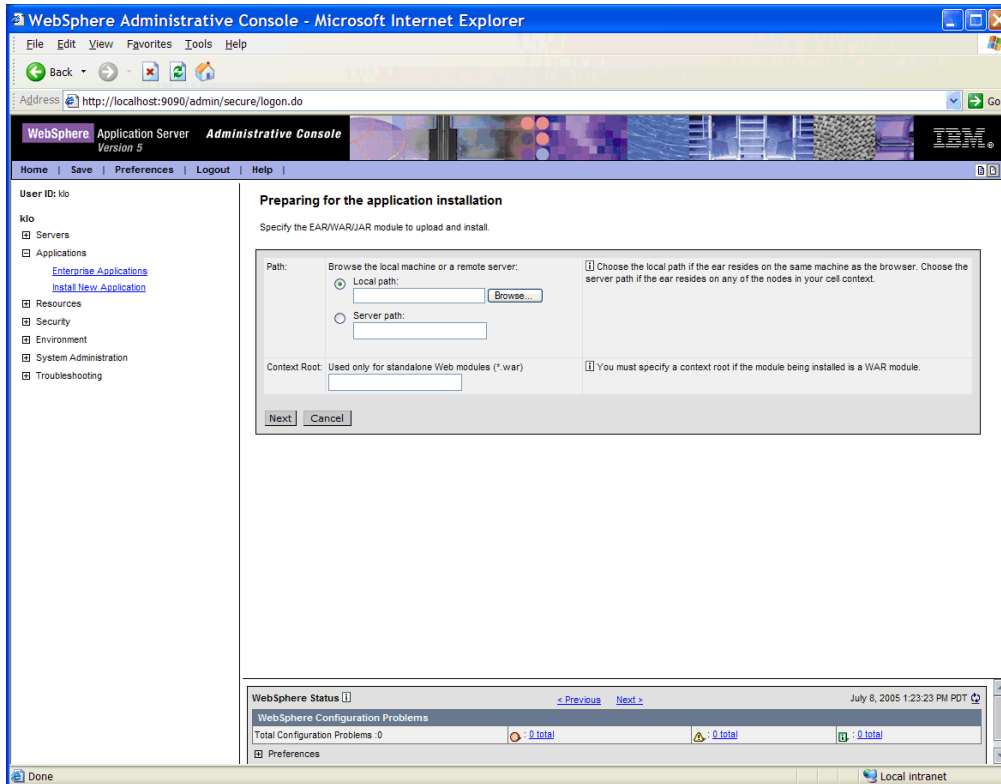
### Logout

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.

## Installing the instance servlet Web Application Archive (WAR) file

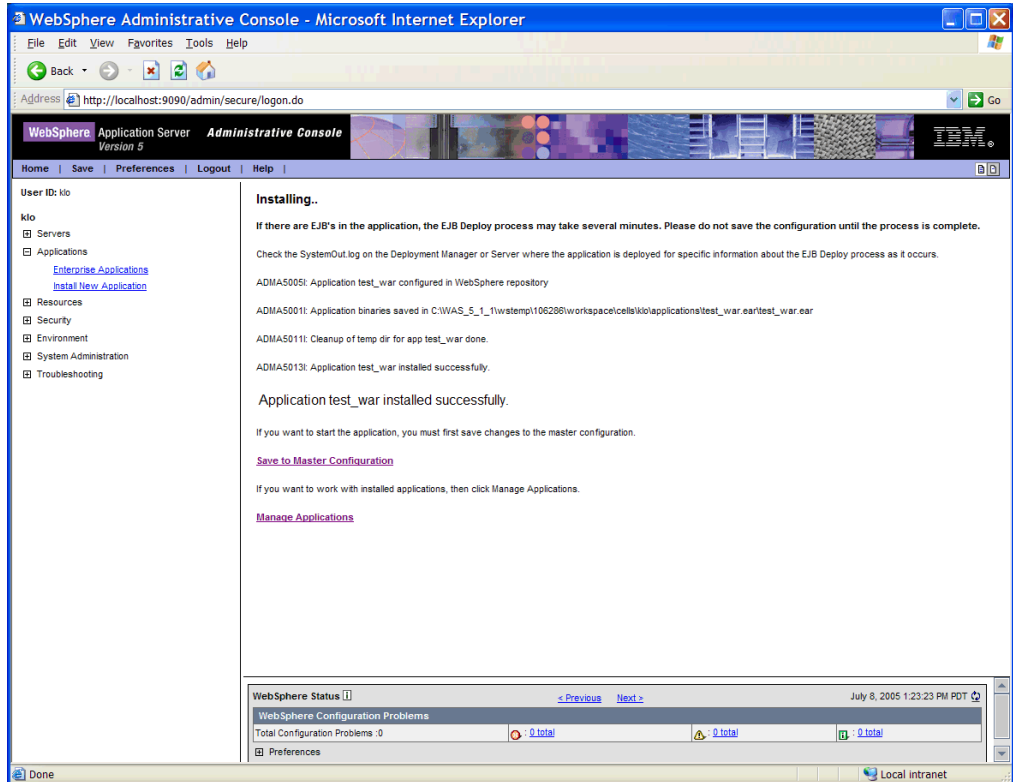
To install the instance servlet Web Application Archive (WAR) file:

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.



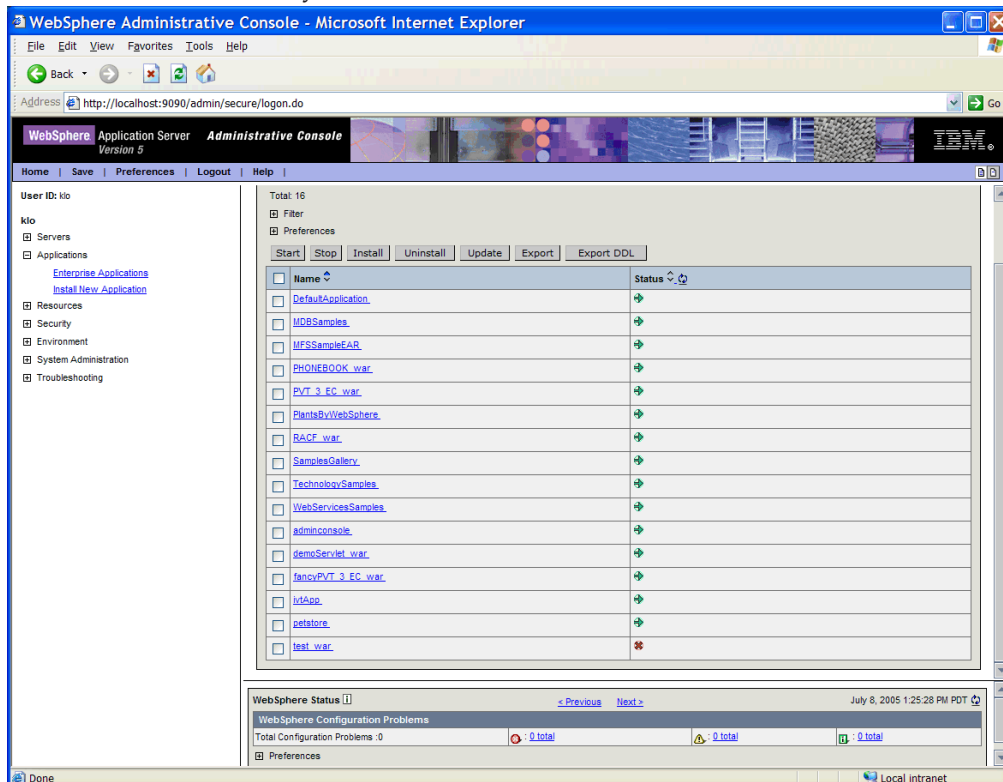
5. Enter the Context Root. The text you enter here will be a part of the URL.
6. Click the **Next** button to go to the next screen and accept the default values.
7. Click the **Next** button to go to the Application Security Warnings window.
8. Click the **Continue** button to go to the Install New Application window and accept the default values.
9. Click the **Next** button and accept the defaults three more times.

- Click the **Finish** button. You should get the message Application [*application name*] installed



successfully.

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



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 invoke the instance servlet:

1. From a Microsoft Internet Explorer or Mozilla Firefox browser, enter the URL of the instance servlet (for example `https://WebServer:Port/ContextRoot/MyServlet`). If user authentication is set on, proceed to step 2, if not, proceed to step 4.
2. Enter your valid user ID and password.
3. If the user authentication is set 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:

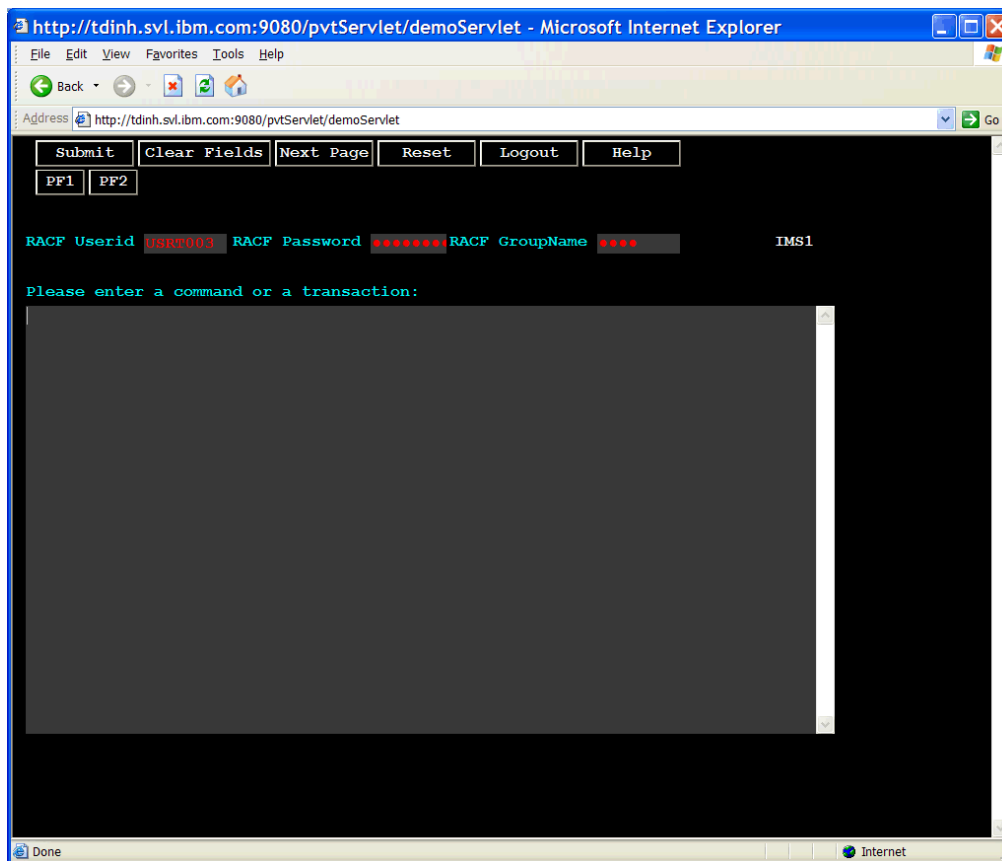


Figure 12. Initial blank page

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.

Here are some more details:

- 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 performs data transformation 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 describes various scenarios that the MFS Adapter supports:

**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 PF 36 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 ENDMPPPI (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 displays the *MFS Web Enablement Version 9.1.0 User's Guide and Reference*.
- Attribute bytes support, including:

### **Protected**

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

### **Modified**

Data in this field can be modified. Setting this attribute to "true" makes 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, including:

### **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 1. 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.

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):

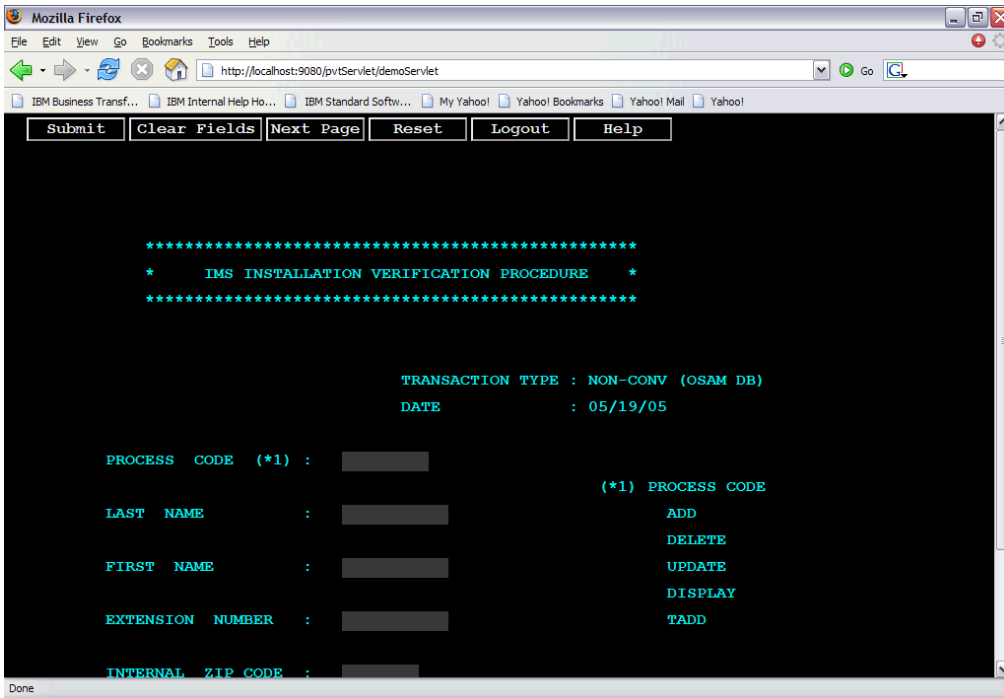


Figure 13. Classic 3270 type terminal simulation

## Stylized 3270 type terminal

Displays a Web page interface.

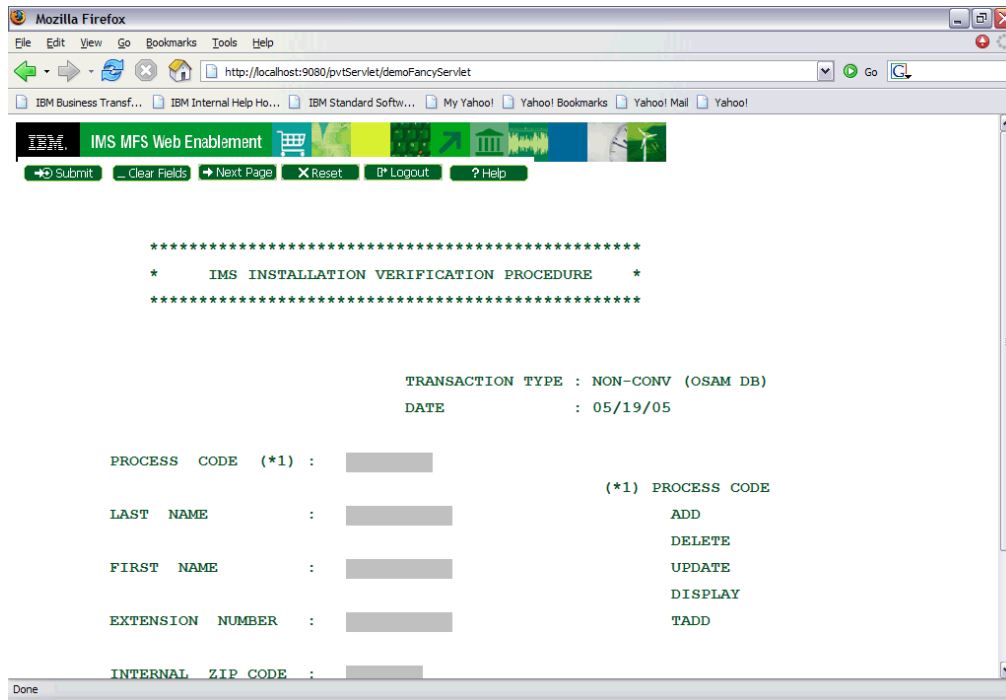


Figure 14. Stylized 3270 type terminal

The customizable attributes of the MFS style sheets are:

- Font attributes:
  - Color
  - Family
  - Size
  - Weight
- Background color
- Button style
- Java script 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.



---

## 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”

---

### 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.xml` and `IVTNOMI1.xml` 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 MS950)
- 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 N):
>> You entered N by default
```

6. Press **Enter** to specify the source codepage as MS950 (default based on system locale):

```
Specify codepage for source (our system default is MS950):
>> You entered MS950 by default.
```

7. Press **Enter** to specify the host codepage as Cp037 (for EBCDIC United States):

```
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 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\
5. 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
6. Specify c:\\$Projects\MFSXML\source\sample3270.xml 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.xml  
>> You entered file:/c:\\$Projects\MFSXML\source\sample3270.xml
7. Specify ecdb31.svl.ibm.com as the 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
8. Specify 9999 as the host port number and press **Enter**:  
Specify a port number ('?' for help): 9999  
>> You entered 9999
9. Specify IMS1 as the IMS datastore name and press **Enter**:  
Specify IMS1 as the IMS datastore name ('?' for help): IMS1  
>> You entered IMS1
10. 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
11. 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
12. The instance servlet is generated and compiled in the output directory that is specified:  
Generating servlet.....completed  
Servlet is being compiled.....completed.
13. 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
14. 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 a 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**:

Please 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, running through step 2 in advance is mandatory.

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

\*Please 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 is going to be generated with the following instance servlets.

- 1) .\WEB-INF\classes\PHONEBOOK.class
- 2) .\WEB-INF\classes\PHONEBOOK.java

2. Enter the 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.

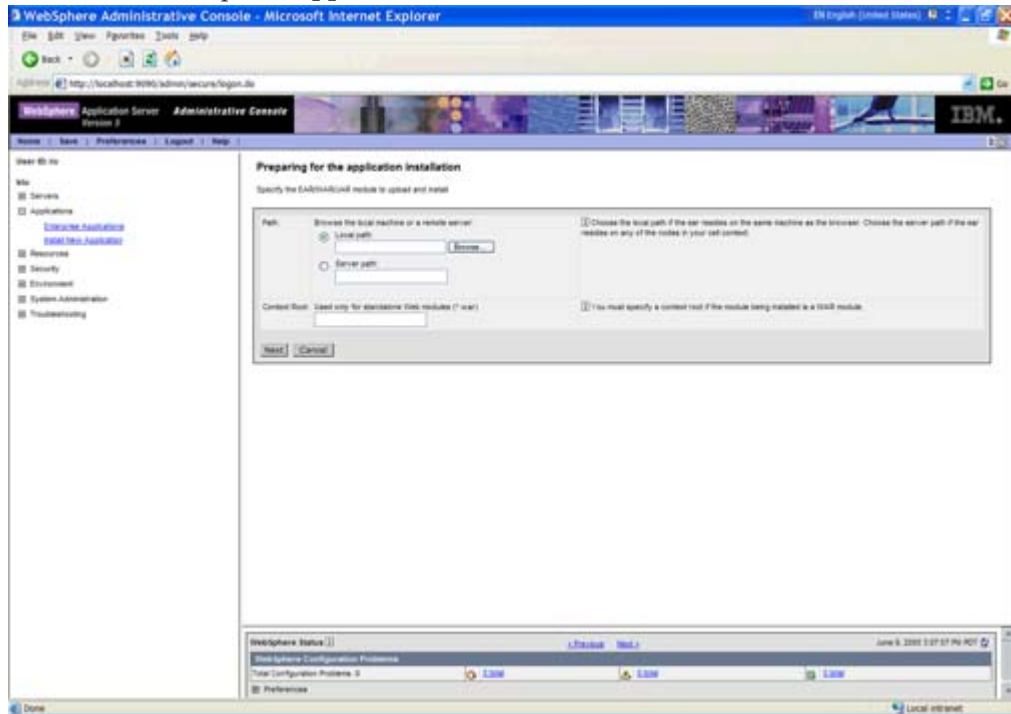
**Note:** It is important that the WebSphere Application Server administrator saves all changes and restarts WebSphere Application Server whenever making changes to the IMS resource adapter.

---

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

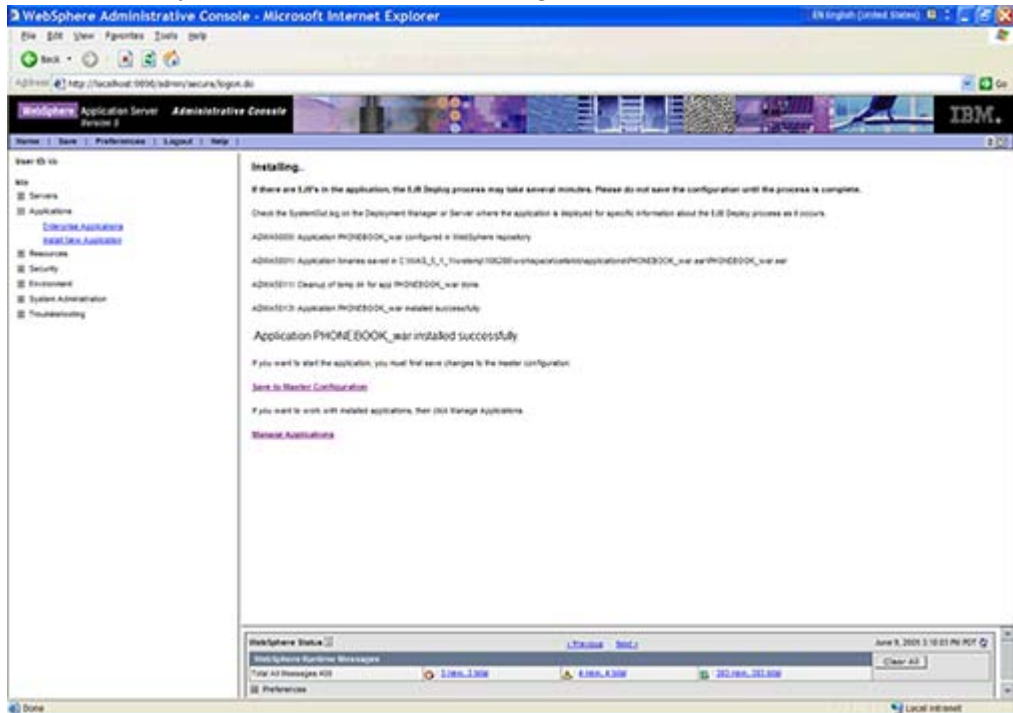
To deploy the PB.war file onto WebSphere Application Server:

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**



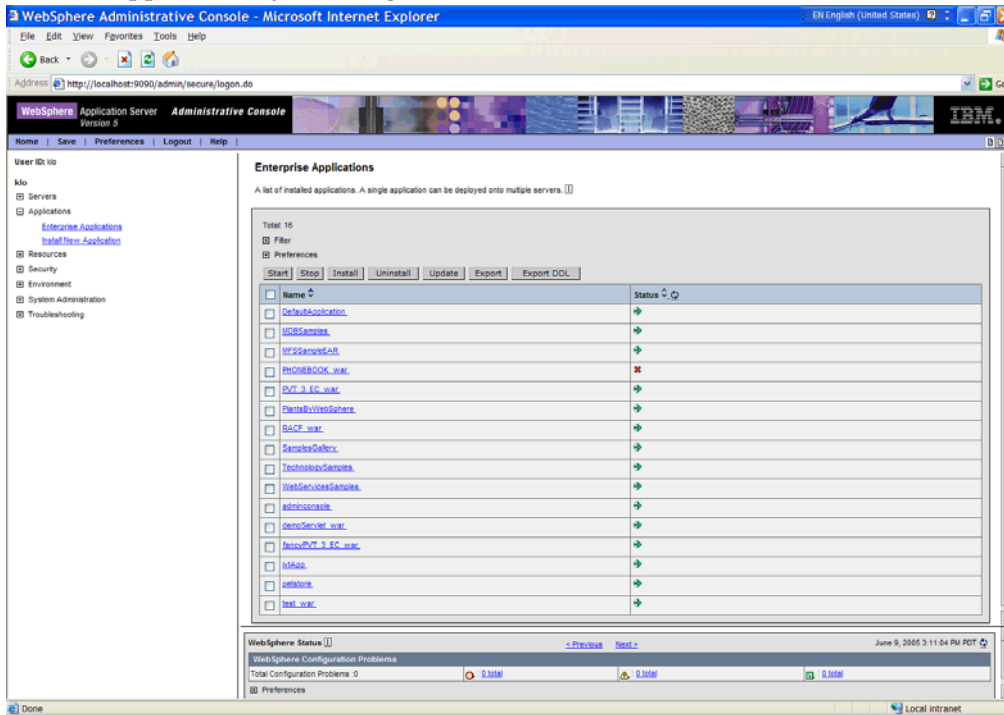
Application.

2. Click the **Browse** button 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 the **Next** button to go to the next screen, keep the default values.
5. Click the **Next** button to go to the **Application Security Warnings** window.
6. Click the **Continue** button to go to the **Install New Application** window, keep the default values.
7. Click the **Next** button to go through the next three screens.
8. Click the **Finish** button and you should receive the message Application PB\_war installed



successfully.

9. Click the **Save** button.
10. Start the application by selecting **Enterprise Applications**.



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 installation, and `PHONEBOOK` is the instance servlet that is generated from step 2.

---

## Step 7: Invoke PHONEBOOK

To invoke the PHONEBOOK application sample:

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

---

## Step 8: Logout

To logout:

Click the **Logout** button.

---

## 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 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," 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

Requested XMI not found: *modName*

## Explanation

The XMI can not be found in the XMI repository.

## **System action**

The system default DFSMO3 page is displayed.

## **User response**

Use the MFS XML Utility to create and transfer the missing XMI file into the MFS XMI repository location specified in the application's deployment descriptor web.xml file.

## **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.

---

## Notices

Note to U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

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 Documentation 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 Documentation. The furnishing of this Documentation 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, 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 OR CONDITIONS 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:

*Lab Director  
IBM Canada Ltd. Laboratory  
8200 Warden Avenue  
Markham, Ontario, Canada L6G 1C7*

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 Documentation 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.

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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may 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 illustrates 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. 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 IBM's application programming interfaces.

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

(C) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (C) Copyright IBM Corp. 2000, 2005. All rights reserved.

## **Programming interface information**

Programming interface information is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

**Warning:** Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.



## Trademarks and service marks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, or other countries, or both:

- AIX
- CICS
- DB2
- DB2 Extenders
- DB2 Universal Database
- e-business
- HotMedia
- IBM
- IBM Cloudscape
- IMS
- iSeries
- MQSeries
- OS/390
- S/390
- VisualAge
- WebSphere
- z/OS

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

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

UNIX is a registered trademark of The Open Group

Rational and ClearCase are trademarks of International Business Machines Corporation and Rational Software Corporation, in the United States, other countries or both.

Other company, product, and service names, which may be denoted by a double asterisk(\*\*), may be trademarks or service marks of others.

(C) Copyright IBM Corporation 2000, 2005. All Rights Reserved.







Printed in USA