TIBCO ActiveMatrix® Management Agent for WebSphere

User's Guide

Software Release 1.2.0
July 2009

Important Information

Contents

Figures

Tables

Preface

TIBCO ActiveMatrix® Management Agent for WebSphere extends policy management to web services deployed in IBM WebSphere Application Server. The management agent is a plug-in component within WebSphere servers, and cooperates with TIBCO ActiveMatrix Policy Manager.

The Service probe for IBM WebSphere Application Server allows monitoring and management of Java EE Web Services and Service References (JAXRPC/JAXWS) which are deployed and running in IBM WebSphere Application Server 6.1.x using TIBCO ActiveMatrix Service Performance Manager.

Readers of this document must already be familiar with TIBCO ActiveMatrix Policy Manager, TIBCO ActiveMatrix Service Performance Manager software and with IBM WebSphere software.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. Please see the readme.txt file for the availability of this software version on a specific operating system platform.

Topics

Changes from the Previous Release of this Guide

This section itemizes the major changes from the previous release of this guide.

Service Probe for IBM WebSphere Application Server

This releases adds this features which can be used to monitor and manage Java EE Web Services and Service References (JAX-RPC/JAX-WS) which are deployed and running in IBM WebSphere Application Server. For detailed information refer to Chapter 1, Introduction.

Sample for Service Probe for IBM WebSphere Application Server

The steps to deploy the sample for Service probe for IBM WebSphere Application Server are added to the Appendix. For details, refer to Appendix A, Samples.

Related Documentation

This section lists documentation resources you may find useful.

TIBCO ActiveMatrix Management Agent for WebSphere Documentation

The following documents form the TIBCO ActiveMatrix Management Agent for WebSphere documentation set:

- *TIBCO ActiveMatrix Management Agent for WebSphere Installation* This book presents instructions for installing the product.
- *TIBCO ActiveMatrix Management Agent for WebSphere User's Guide* This book describes ActiveMatrix Management Agent for WebSphere software, and presents instructions for configuring and using the product.
- *TIBCO ActiveMatrix Management Agent for WebSphere Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

Other TIBCO Product Documentation

You may find it useful to read the documentation for the following TIBCO products:

- TIBCO ActiveMatrix™ Policy Manager
- TIBCO ActiveMatrix™ Service Grid
- TIBCO ActiveMatrix™ Registry
- TIBCO Enterprise Message Service™
- TIBCO Administrator™
- TIBCO ActiveMatrix™ Service Performance Manager

Third Party Documentation

IBM

- IBM WebSphere Application Server
  http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp
  - **Introduction**
    http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_ovrex.html
  - **JAX-RPC**
    http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/org.eclipse.jst.ws.doc.user/concepts/cjaxrpc.html
  - **JAX-WS**

    http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.webservice.wsfp.doc/topics/cjaxws.html

Java

The various Java specfications supported by Management Agent for WebSphere can be obtained from the Java Community Process web site http://jcp.org:

- Java Message Service
- JAX-RPC
- JAX-WS

Web Services Standards

Management Agent for WebSphere supports web services standards sponsored by the following standards bodies:

- World Web Web Consortium web services activity
  http://www.w3.org/2002/ws/
  - WSDL 1.1
    http://www.w3.org/TR/wsdl
- OASIS web services committees
  http://www.oasis-open.org/committees/tc_cat.php?cat=ws
- Web Services Interoperability
  http://www.ws-i.org/

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
TIBCO_HOME	All TIBCO products are installed under the same directory. This directory is referenced in documentation as *TIBCO_HOME*. The value of *TIBCO_HOME* depends on the operating system. For example, on Windows systems, the default value is `C:\tibco`.
`code font`	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use `MyCommand` to start the foo process.
`bold code font`	Bold code font is used in the following ways: • In procedures, to indicate what a user types. For example: Type **`admin`**. • In large code samples, to indicate the parts of the sample that are of particular interest. • In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, `MyCommand` is enabled: MyCommand [**`enable`** | disable]
italic font	Italic font is used in the following ways: • To indicate a document title. For example: See *TIBCO BusinessWorks Concepts*. • To introduce new terms For example: A portal page may contain several *portlets*. Portlets are mini-applications that run in a portal. • To indicate a variable in a command or code syntax that you must replace. For example: `MyCommand` *pathname*
Key combinations	Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C. Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2 Syntax Typographical Conventions

Convention	Use
`[ ]`	An optional item in a command or code syntax. For example: `MyCommand [optional_parameter] required_parameter`
`|`	A logical 'OR' that separates multiple items of which only one may be chosen. For example, you can select only one of the following parameters: `MyCommand para1 | param2 | param3`

`{ }`	A logical group of items in a command. Other syntax notations may appear within each logical group.
	For example, the following command requires two parameters, which can be either the pair `param1` and `param2`, or the pair `param3` and `param4`.
	`MyCommand {param1 param2} | {param3 param4}`
	In the next example, the command requires two parameters. The first parameter can be either `param1` or `param2` and the second can be either `param3` or `param4`:
	`MyCommand {param1 | param2} {param3 | param4}`
	In the next example, the command can accept either two or three parameters. The first parameter must be `param1`. You can optionally include `param2` as the second parameter. And the last parameter is either `param3` or `param4`.
	`MyCommand param1 [param2] {param3 | param4}`

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, please contact TIBCO Support as follows.

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

  http://www.tibco.com/services/support

- If you already have a valid maintenance or support contract, visit this site:

  https://support.tibco.com

Entry to this site requires a user name and password. If you do not have a user name, you can request one.

Chapter 1 Introduction

This chapter presents Management Agent for WebSphere, its capabilities, prerequisites and general operation.

Topics

Product Overview

TIBCO ActiveMatrix® Management Agent for WebSphere extends policy management to web services deployed in IBM WebSphere Application Server. The management agent is a plug-in component within WebSphere servers, and cooperates with TIBCO ActiveMatrix Policy Manager.

Management agents installed within WebSphere instances play the same role as the node agents built into ActiveMatrix Service Grid nodes, and they operate in a similar way.

In the XACML usage model, each management agent acts as a policy decision point (PDP) and a policy enforcement point (PEP). Meanwhile, Policy Manager software provides the policy administration point (PAP) and policy repository point (PRP) components.

The Service probe for IBM WebSphere Application Server allows monitoring and management of Java EE Web Services and Service References (JAXRPC/JAXWS) which are deployed and running in IBM WebSphere Application Server 6.1.x using TIBCO ActiveMatrix Service Performance Manager.

Prerequisites & Dependencies

General

Management Agent for WebSphere requires the following software:

- TIBCO ActiveMatrix Policy Manager 3.0
- IBM WebSphere Application Server 6.1.0 with latest fix pack

Management Agent for WebSphere is certified only with WebSphere Application Server as well as with IBM WebSphere Application Server Network Deployment, and not with other editions of the WebSphere product.

JMS

Management Agent for WebSphere requires TIBCO Enterprise Message Service™ 4.4.3 (or later) as the JMS provider for SOAP/JMS based web services.

Platform

Management Agent for WebSphere supports the following operating system and hardware platforms:

- Windows 2003 x86
- Windows XP
- Solaris SPARC 64-bit and 32-bit
- Solaris x86 64-bit
- AIX 5.3 (Power PC 64-bit)

Deployment Prerequisite

When you deploy your Java web services in a WebSphere application server, you must *not* disable the WebSphere installation option **Create MBeans for resources** (this option is enabled by default).

Capabilities

WebSphere Environment

Management Agent for WebSphere supports WebSphere profiles that define a single stand-alone server or a federated server. A federated server can be created using the WebSphere Application Server Network Deployment product.

WebSphere Package

Management Agent for WebSphere interoperates with WebSphere Application Server 6.1.0. with latest fix pack and WebSphere Application Server Network Deployment 6.1.0 with latest fix pack.

Protocols and Transports

Management Agent for WebSphere manages JAX-RPC 1.1 and JAX-WS 2.0 web services.

For JAX-RPC 1.1, Management Agent for WebSphere supports both SOAP/HTTP and SOAP/JMS.

For JAX-WS 2.0, Management Agent for WebSphere supports only HTTP transports (because WebSphere does not support JMS transports for JAX-WS).

Deployment Structure

Management Agent for WebSphere deploys two components within each WebSphere application server instance:

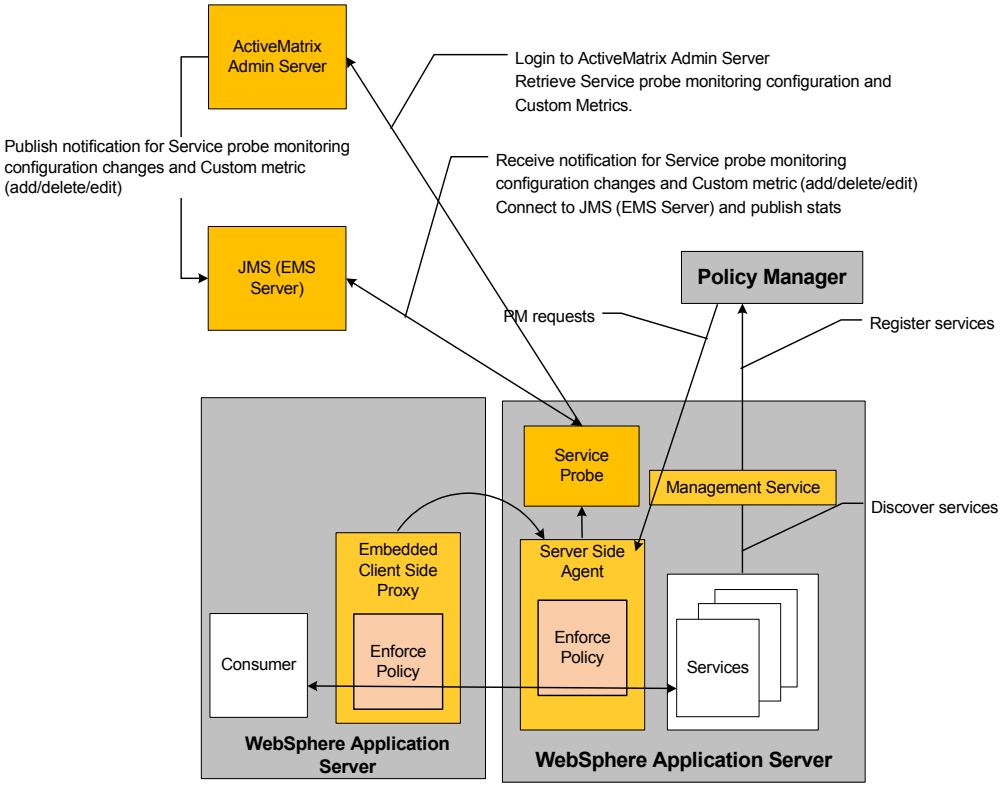
- **Agent** The agent is a message handler, which intercepts messages and enforces policies for web services deployed within the application server.
- **Management Service** The management service has two main tasks:
  - It discovers the endpoints of services deployed within the application server, and registers those endpoints with Policy Manager.
  - It acts as a communications intermediary between Policy Manager and the agent. Examples include requests to manage endpoints; requests to add, delete, enable and disable policies; requests to query log data.

Complementary policies for client-side policy enforcement has been deprecated. The Embedded Client-Side Proxy feature has been introduced to apply explicit client-side policies.

- **Service Probe for IBM WebSphere Application Server** The main task of Service Probe is to:
  - Allow monitoring and management of Java EE Web Services and Service References (JAX-RPC/JAX-WS) which are deployed and running in IBM WebSphere Application Server 6.1.x using TIBCO ActiveMatrix Service Performance Manager.

The following Figure 1 illustrates this structure, and the communication flows involved in these tasks.

Figure 1 Deployment Structure



Combination Scenarios

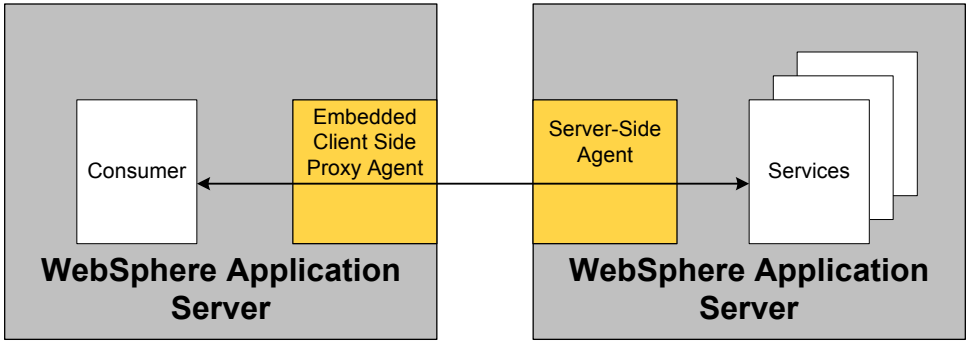
Management Agent for WebSphere enables interoperation with services deployed using TIBCO ActiveMatrix Service Grid and TIBCO BusinessWorks. This section presents several use cases of Management Agent for WebSphere in combination with these other products.

WebSphere Calls WebSphere

Figure 2 highlights two roles of the management agent:

- **Embedded Client Side Proxy Agent** intercepts outbound request messages from a consumer and inbound reply or fault messages returning from the provider.
- **Server-side agent** intercepts inbound request messages to a provider and outbound reply or fault messages returning to the consumer.

Figure 2 WebSphere Calls WebSphere

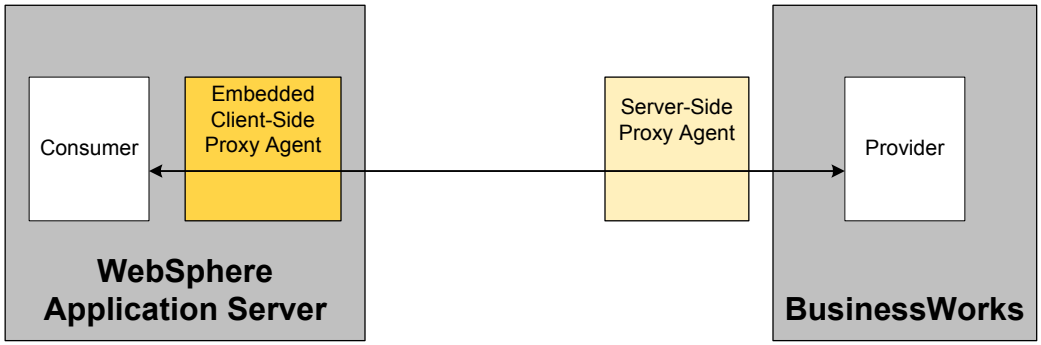


WebSphere Calls BusinessWorks

Providers deployed in BusinessWorks (as a stand-alone product) require an instance of TIBCO ActiveMatrix Policy Agent to act as a provider-side proxy agent.

Embedded client side proxy agents do not negotiate with other agents to automatically apply complementary client-side policies. For a complete explanation, see Chapter 6, Cryptography in *TIBCO ActiveMatrix Policy Manager Policy Reference*.

Figure 3 WebSphere Calls BusinessWorks

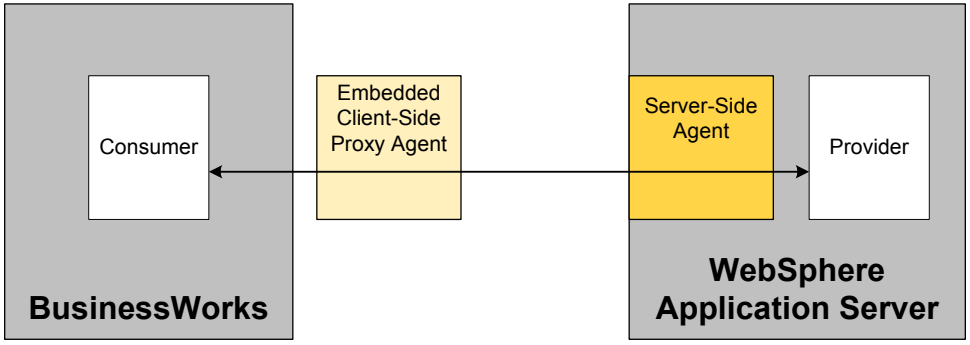


BusinessWorks Calls WebSphere

Consumers deployed in BusinessWorks (as a stand-alone product) require an instance of TIBCO ActiveMatrix Policy Agent to act as a client-side proxy agent.

Proxy agents do not negotiate with other agents to automatically apply complementary client-side policies. For a complete explanation, see Chapter 6, Cryptography in *TIBCO ActiveMatrix Policy Manager Policy Reference*.

Figure 4 BusinessWorks Node Calls WebSphere



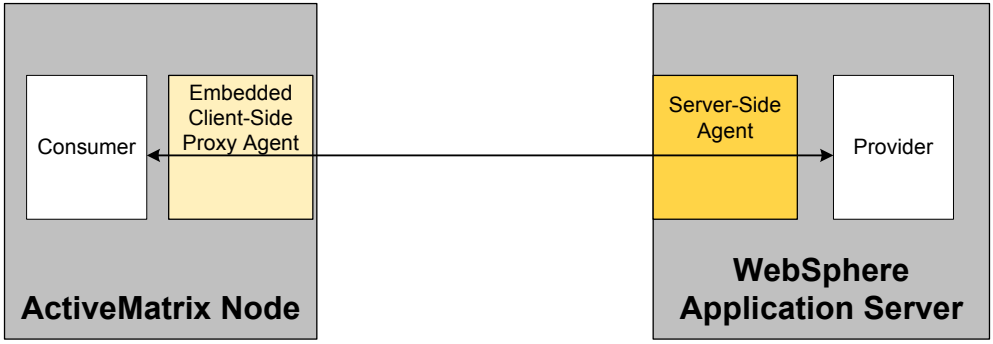
ActiveMatrix Node Calls WebSphere

Consumers deployed in ActiveMatrix Service Grid require an instance of TIBCO ActiveMatrix Policy Agent to act as a client-side proxy agent.

Proxy agents do not negotiate with other agents to automatically apply complementary client-side policies. For a complete explanation, see Chapter 6, Cryptography in *TIBCO ActiveMatrix Policy Manager Policy Reference*.

This is true only for ActiveMatrix Service Grid version 2.0.x and below. In the ActiveMatrix Service Grid 2.1, you do not need the instance of Policy Agent (shown in Figure 5), as it already has an embedded client side proxy agent. This agent, like proxy agent, does not negotiate with other agents to automatically apply the complementary policies.

Figure 5 ActiveMatrix Node Calls WebSphere



WebSphere Calls ActiveMatrix Node

On consumer side, the Embedded Client Side Proxy Agent feature exposes the service reference endpoint on which you can apply the explicit client-side policies. Refer to the section , Using Embedded Client Side Proxy Feature in Chapter 3, Tasks.

Figure 6 WebSphere Calls ActiveMatrix Node

Availability of Policy Types

Management Agent for WebSphere supports the following policy types:

- Logging policies—no restrictions
- Authentication policies—no restrictions
- Authorization policies—no restrictions
- Censor Response policies—no restrictions
- Cryptographic policies:
  - Explicit receiving policies
  - Explicit forwarding policies
  - Encrypt Request Element
- Credential Mapping policies

Unavailable Policy Types

Management Agent for WebSphere does not support the following policy type:

- Routing policies

TIBCO ActiveMatrix Policy Manager

Service Status Indicators

The Policy Manager console displays information about WebSphere services and the policies that pertain to them.

Policy Manager indicates the status of a service with a colored dot (green or red).

TIBCO ActiveMatrix Service Performance Manager

Service Performance Manager Dashboard

The rules are triggered, actions are enabled and alerts are sent based on the condition defined on the services. The dashboard shows all the output data related to rules, services, and infrastructure.

For detailed information on "Service Performance Manager Dashboard", refer to the *TIBCO ActiveMatrix Service Performance Manager User's Guide* .

Service Probe

- Publish server start and stop events on the TIBCO ActiveMatrix Service Performance Manager JMS Queue with server and machine details.
- Provide machine and JVM information along with the server details.
- Discover all modules in application, all Services and Service References in a module and publish on the TIBCO ActiveMatrix Service Performance Manager JMS Queue.
- Discover all modules in application, all Services and Service References in a module and publish on the TIBCO ActiveMatrix Service Performance Manager JMS Queue, when a new application gets installed.
- Publish the Service and Service Reference execution stats along with the custom metrics stats (if custom metric is registered).
- Publish Services and Service References Stop event when the Application is stopped, and Uninstall event when Application is uninstalled from the Server.

WebSphere Custom Security Permissions

The management service component of Management Agent for WebSphere requires custom security permissions. `PolicyAgentManagementService_<appsvr_profile_name>.ear` contains a configuration file named `was.policy`, which grants these permissions to the management service application code:

```
permission java.security.AllPermission;
permission com.ibm.websphere.security.WebSphereRuntimePermission "AdminPermission";
permission com.ibm.websphere.management.AdminPermission "getAdminService";
permission com.ibm.websphere.management.AdminPermission "getMBeanFactory";
permission com.tivoli.jmx.MBeanServerPermission "MBeanServerFactory.*";
permission com.tivoli.jmx.MBeanServerPermission "MBeanServer.*";
```

Chapter 2 Configuration

This chapter presents the steps required to configure Management Agent for WebSphere.

Topics

Navigating the Installation Directories

Table 3 presents the organization of the directories associated with Management Agent for WebSphere software.

Table 3 Management Agent for WebSphere Installation Directories (Sheet 1 of 3)

Directory	Content
Directories associated with this product (ActiveMatrix Management Agent for WebSphere)	
tibco	*TIBCO_HOME*, the root for TIBCO software
amma-was	ActiveMatrix Management Agent for WebSphere software
1.2	Release
bin	Configuration utilities and configuration files
docs	Management Agent for WebSphere documentation
lib	Contains the Management Agent installer jar
logs	Created when you run ManagementAgentConfig or RevertManagementAgent utility
serviceprobe	Contains wasserviceagent.properties
samples	Contains WebSphere service samples
templates	Unconfigured .ear file for the management service component, amma-was-config.xml and template file
release.txt	Contains the release and the build number
uninstaller_archives	Uninstaller scripts and data

Table 3 Management Agent for WebSphere Installation Directories (Sheet 2 of 3)

`instances`	The configuration utility creates the `instances` directory, and a subdirectory for each configured WebSphere application server instance (identified by the profile name). `ManagementAgentConfig` creates the file `pluginagent.properties` in the WebSphere application server profile directory. Each subdirectory under `instances` contains configuration data for a specific WebSphere profile. Do not place other subdirectories or files in this tree.
appsvr_profile_name	
`cell`	
cellname	
`node`	
node name	
`server`	
servername	
`agent cache`	The agent caches configuration data, managed endpoint data, policy data, logging configuration, log data, database configuration in this directory.
`webapps`	Configured .ear file for the management service component
`pluginagent.properties`	Contains information required for Management Agent
Directories associated with WebSphere Application Server	
`IBM`	*ibm_home*, the root for IBM software
`WebSphere`	
`AppServer`	*was_home*, the root for WebSphere Application Server

Table 3 Management Agent for WebSphere Installation Directories (Sheet 3 of 3)

bin	ManagementAgentConfig modifies startServer by appending the three (tibco.agent.property.file,com.ibm.websphere.webservices.saaj.accessSOAPBody and com.amberpoint.util.Extension.dir) system properties in the java execution command. It saves the backup copy (with extension .bak) in the same directory before modification.
plugins	ManagementAgentConfig copies the TIBCO OSGI plug-ins required by the Management Agent for WebSphere into this directory.
profiles	WebSphere Users define application server profiles in this directory.
appsvr_profile_name	Profile for a particular application server
bin	ManagementAgentConfig modifies setupCmdline, to set the TIBCO_AGENT_PROP_FILE environment variable. It saves the backup copy (with extension .bak) in the same directory before modification.
properties	ManagementAgentConfig creates **serverPaths.properties** file under this directory.

Configuration Checklist

Table 4 is a checklist for configuring Management Agent for WebSphere software. It summarizes each step briefly; further detail is available in subsequent sections of this book.

Table 4 Installation Checklist

Step	Notes
1. Install the product	See *TIBCO ActiveMatrix Management Agent for WebSphere Installation*.
2. Service Probe for IBM WebSphere Application Server	See *TIBCO ActiveMatrix Management Agent for WebSphere Installation*.
3. Record the Management Agent for WebSphere Configuration Plan, page 20	Record information for conveniently configuring the management agent.
4. Configuring the Management Agent, page 28	
5. Arranging the Shared Secret for the Management Agents, page 30	
6. Configuring the Database, page 31	Select the database to serve the management agent instance. Prepare the database, and configure the management agent to use the database to store logs from logging policies.
7. Start the agents by starting the corresponding WebSphere application servers.	See Starting the Management Agent and Service Probe for IBM WebSphere Application Server on page 34.
Other Configuration Instructions You might also need to configure the following additional items, depending on the WebSphere applications you deploy.	
• Configuring a JMS-Based JAX-RPC Service, page 39	
• Enabling the Security Context API: Administrative Task, page 67	

Record the Management Agent for WebSphere Configuration Plan

Use the forms in this section to record decisions you make as part of your installation plan. You will use this information in subsequent tasks.

The following parameter descriptions apply to the items in Table 5 on page 24.

WebSphere Application Server Home

Record the full directory pathname of installation directory for WebSphere Application Server software; for example, `/IBM/WebSphere/AppServer`.

In the Management Agent for WebSphere documentation set, the variable *was_home* refers to this location.

Application Server Profile Name

You can configure at most one ActiveMatrix management agent for each WebSphere application server. For each management agent, record the name of corresponding application server in Table 5 on page 24.

Admin Username and Password

If an application server is protected by an administrative username and password, you must supply the those values when configuring the corresponding management agent—but to protect the security of this information, do *not* record it here.

If the Application server is a Federated application server and the Deployment Manager associated with it is protected by an administrative username and password, then you must enter the credentials of the Deployment Manager when configuring the management agent.

If the `ManagementAgentConfig` utility fails due to invalid username and password, re-run the `ManagementAgentConfig` utility with valid credentials.

WebSphere Host Name

Each application server instance has a *default host object*, which consists of a hostname and port number. This object describes a locator for communication into services within the application server. In particular, Policy Manager components communicate with the management agent at this address.

Record the fully-qualified hostname from the WebSphere default host object. (You specified this hostname when you configured the WebSphere application server profile.)

Policy Manager Connects Using SSL

Policy Manager connects to management agents to deliver policy instructions and to query for log data. These requests use either HTTP protocols or HTTPS (SSL) protocols. Each management agent listens for one of these protocols, and ignores the other. To properly configure a management agent to receive these requests, you must first determine which protocol Policy Manager uses for these requests.

If the Policy Manager connects using SSL, then circle *yes* in Table 5. Otherwise, circle *no*.

If you circle *yes*, then you must prepare a keystore for the management agent containing its identity, and a truststore for Policy Manager containing the agent's public certificate.

For further information, see Chapter 5, SSL, on page 57; especially Configuring SSL for Policy Manager Requests on page 60.

WAS Requires Client Authentication from Policy Manager

The WebSphere application server could be configured to require client authentication. If so, then when Policy Manager connects using SSL, it must supply its identity. (If Policy Manager does not connect using SSL, then this requirement is not relevant.)

If the application server is configured to require client authentication, circle *yes* in Table 5. Otherwise, circle *no*.

If you circle *yes*, then you must prepare a keystore for Policy Manager containing its identity, and a truststore for the management agent containing the Policy Manager's public certificate.

For further information, see Chapter 5, SSL, on page 57; especially Configuring SSL for Policy Manager Requests on page 60.

HTTP or HTTPS Transport Port

An application server accepts inbound requests on this port, and routes them to its deployed services.

Record one of two possible port numbers, depending on the value you specified for Policy Manager Connects Using SSL (see above):

- If you circled *yes*, then record the HTTPS transport port.
- If you circled *no*, then record the HTTP transport port.

(You specified these ports when you configured the WebSphere application server profile.)

SOAP Connector Port

An application server uses a connector port to route SOAP requests among its deployed services. In particular, the management agent connects to a JMX interface at this port to discover the deployed services.

Record the SOAP connection port. (You specified this port when you configured the WebSphere application server profile).

For a **Standalone Application Server** get this value from the following `serverindex.xml` file:
`<was_home>/profiles/profile_name/config/cells/cell_name/nodes/node_name`.

The `serverindex.xml` file contains an element with a `serverType` attribute value as "`APPLICATION_SERVER`". This element consists of a child element with `endPointName` attribute value as "`SOAP_CONNECTOR_ADDRESS`". This element has a child element with one of the attribute as port. The value for this attribute must be used as the soap connector port.

For a **Federated Application Server** get this value from the following `serverindex.xml` file:
`<was_home>/profiles/profile_name/config/cells/cell_name/nodes/node_name`

The `serverindex.xml` file contains an element with a `serverType` attribute value as "`NODE_AGENT`". This element consists of a child element with `endPointName` attribute value as "`SOAP_CONNECTOR_ADDRESS`". This element has a child element with one of the attribute as port. The value for this attribute must be used as the soap connector port.

Policy Manager Host

Each management agent connects to a Policy Manager instance running on a specific host computer.

Record the fully qualified hostname of the Policy Manager central services host computer. (You specified this hostname when you configured Policy Manager.)

Connect to Policy Manager Using SSL

Management agents connect to Policy Manager to register WebSphere services. These requests use either HTTP protocols or HTTPS (SSL) protocols. To properly configure a management agent to send these requests, you must first determine which protocol to use for these requests.

If the management agent connects using SSL, then circle *yes* in Table 5. Otherwise, circle *no*.

If you circle *yes*, then you must prepare a keystore for Policy Manager containing its identity, and a truststore for the management agent containing the Policy Manager's public certificate.

Policy Manager Requires Client Authentication from Management Agent

Policy Manager can be configured to require client authentication. If so, then when the management agent connects using SSL, it must supply its identity. (If the management agent does not connect using SSL, then this requirement is not relevant.)

If Policy Manager is configured to require client authentication, circle *yes* in Table 5. Otherwise, circle *no*.

If you circle *yes*, then you must prepare a keystore for the management agent containing its identity, and a truststore for Policy Manager containing the management agent's public certificate.

Policy Manager HTTP or HTTPS Port

Policy Manager accepts inbound requests on its central services port.

Record the central services port that you specified when you configured Policy Manager.

Record Management Agent Instances

Record the information for configuring each management agent instance in Table 5. For further details about each item, see the descriptions above.

To facilitate later use of these values, Table 5 divides the information pertaining to each instance into three sections (highlighted by color) corresponding to the three input screens of the configuration utility's graphic user interface.

Table 5 Management Agent for WebSphere Instances Plan (Sheet 1 of 4)

#	Record Information	
1	**WebSphere Application Server Home**	
	Application Server	
	WebSphere Host Name	
	Policy Manager Connects Using SSL *yes no*	**WAS Requires Client Authentication from Policy Manager** *yes no*
	HTTP or HTTPS Port	
	SOAP Connector Port	
	Policy Manager Host Name	
	Connect to Policy Manager Using SSL *yes no*	**Policy Manager Requires Client Authentication from Management Agent** *yes no*
	Policy Manager HTTP or HTTPS Port	
2	**WebSphere Application Server Home**	
	Application Server	
	WebSphere Host Name	
	Policy Manager Connects Using SSL *yes no*	**WAS Requires Client Authentication from Policy Manager** *yes no*
	HTTP or HTTPS Port	
	SOAP Connector Port	
	Policy Manager Host Name	
	Connect to Policy Manager Using SSL *yes no*	**Policy Manager Requires Client Authentication from Management Agent** *yes no*
	Policy Manager HTTP or HTTPS Port	

Table 5 Management Agent for WebSphere Instances Plan (Sheet 2 of 4)

#	Record Information	
3	**WebSphere Application Server Home**	
	Application Server	
	WebSphere Host Name	
	Policy Manager Connects Using SSL *yes no*	**WAS Requires Client Authentication from Policy Manager** *yes no*
	HTTP or HTTPS Port	
	SOAP Connector Port	
	Policy Manager Host Name	
	Connect to Policy Manager Using SSL *yes no*	**Policy Manager Requires Client Authentication from Management Agent** *yes no*
	Policy Manager HTTP or HTTPS Port	
4	**WebSphere Application Server Home**	
	Application Server	
	WebSphere Host Name	
	Policy Manager Connects Using SSL *yes no*	**WAS Requires Client Authentication from Policy Manager** *yes no*
	HTTP or HTTPS Port	
	SOAP Connector Port	
	Policy Manager Host Name	
	Connect to Policy Manager Using SSL *yes no*	**Policy Manager Requires Client Authentication from Management Agent** *yes no*
	Policy Manager HTTP or HTTPS Port	

Table 5 Management Agent for WebSphere Instances Plan (Sheet 3 of 4)

#	Record Information	
5	**WebSphere Application Server Home**	
	Application Server	
	WebSphere Host Name	
	Policy Manager Connects Using SSL *yes no*	**WAS Requires Client Authentication from Policy Manager** *yes no*
	HTTP or HTTPS Port	
	SOAP Connector Port	
	Policy Manager Host Name	
	Connect to Policy Manager Using SSL *yes no*	**Policy Manager Requires Client Authentication from Management Agent** *yes no*
	Policy Manager HTTP or HTTPS Port	
6	**WebSphere Application Server Home**	
	Application Server	
	WebSphere Host Name	
	Policy Manager Connects Using SSL *yes no*	**WAS Requires Client Authentication from Policy Manager** *yes no*
	HTTP or HTTPS Port	
	SOAP Connector Port	
	Policy Manager Host Name	
	Connect to Policy Manager Using SSL *yes no*	**Policy Manager Requires Client Authentication from Management Agent** *yes no*
	Policy Manager HTTP or HTTPS Port	

Table 5 Management Agent for WebSphere Instances Plan (Sheet 4 of 4)

#	Record Information	
7	**WebSphere Application Server Home**	
	Application Server	
	WebSphere Host Name	
	Policy Manager Connects Using SSL *yes no*	**WAS Requires Client Authentication from Policy Manager** *yes no*
	HTTP or HTTPS Port	
	SOAP Connector Port	
	Policy Manager Host Name	
	Connect to Policy Manager Using SSL *yes no*	**Policy Manager Requires Client Authentication from Management Agent** *yes no*
	Policy Manager HTTP or HTTPS Port	
8	**WebSphere Application Server Home**	
	Application Server	
	WebSphere Host Name	
	Policy Manager Connects Using SSL *yes no*	**WAS Requires Client Authentication from Policy Manager** *yes no*
	HTTP or HTTPS Port	
	SOAP Connector Port	
	Policy Manager Host Name	
	Connect to Policy Manager Using SSL *yes no*	**Policy Manager Requires Client Authentication from Management Agent** *yes no*
	Policy Manager HTTP or HTTPS Port	

Configuring the Management Agent

The configuration utility, `ManagementAgentConfig`, explicitly configures a management agent instance.

Before using `ManagementAgentConfig`, you must first start the instance of Policy Manager central services to which the management agent will connect. The configuration utility validates its arguments by contacting central services; if central services are unavailable, the configuration utility reports an error.

Repeat this task to configure each management agent on the respective WebSphere host computer.

To configure a management agent instance, do these steps:

1. Ensure that Policy Manager central services are running.

Before associating a Federated profile, it is important to ensure that the Federated profile is completely synchronized with the Deployment Manager.

2. To configure a federated profile, make sure to start the Deployment Manager.
   - In a new command window, change directory to the Deployment Manager bin location:

     `cd WEBSPHERE_HOME/profiles/<deployment_manager_name>/bin`
   - For Windows:

     `Execute startManager.bat`
   - For Unix:

     `Execute ./startManager.sh`
3. In a new command window, change directory to the utility location:

   `cd TIBCO_HOME/amma-was/<version_num>/bin`
4. Run the `ManagementAgentConfig` utility.

   Supply the required information at each prompt. For a complete description each parameter, see ManagementAgentConfig on page 45.

Make sure the `wasserviceagent.properties` file has correct entries as per your environment settings. The `log4j.properties` file allows you to set customized logging.

For example, the WAS Service Probe log gets generated at the location `<IBM WAS home>\profiles\<profile name> with name was_service_probe.log.`

You can change this location by setting appropriate value for `log4j.appender.service_probe_file.File` property.

5. Verify that the new instance exists. The new instance is a directory named *TIBCO_HOME*`/amma-was/instances/`*appsvr_profile_name/cells/cell_name/nodes/node_name/servers/server_name/pluginagent.properties (*where *appsvr_profile_name* is the name of the corresponding WebSphere application server, which you supply in step 4).

Arranging the Shared Secret for the Management Agents

After you have configured all the agents on a particular WebSphere host, you can arrange the shared secret that they use to communicate with Policy Manager. Since the shared secret is shared in common by a Policy Manager instance and all the agents that communicate with it, you can simultaneously arrange the secret for all the agents co-located on a particular WebSphere host. (However, you must *repeat* this task separately for each WebSphere host where you configure a management agent.)

1. Call `SetSharedSecretForManagementAgent`.

   Paste the encrypted secret as an argument. This value must be identical to the shared secret you supplied to Policy Manager.

   Specify the Policy Manager instance that manages the agent.

2. If you change the location of the trust directory for a WebSphere host, then you must stop and restart all the application servers on that host.

   Otherwise (that is, for subsequent changes to the secret, in which the trust directory location does not change), you do not need to stop and restart the application servers; their management agents refresh themselves at 60-second intervals.

See Also SetSharedSecretForManagementAgent on page 55

Configuring the Database

Policy Manager and its agents (including Management Agent for WebSphere) use a database to store information about services and policies, and to log message traffic.

The default database is HSQLDB. To use any other database, do these steps:

1. Stop the Application server.
2. Obtain the appropriate database driver file, and copy it to the following location:

   was_home/plugins/com.tibco.policy.agent.lib.ext_x.y.z.xxx/

   — Repeat this step on each host computer where you have installed Management Agent for WebSphere.
3. Delete OSGI cache by performing these steps:

   — In a new command window, change directory to the profile's configuration location:

   cd
   WEBSPHERE_HOME/profiles/<*appserver_profile_name*>/configuration

   — Delete all contents present in this directory
4. Re-start the Application server. Refer to section , Starting the Management Agent and Service Probe for IBM WebSphere Application Server in Chapter 3, Tasks.
5. Review Appendix A, Other Databases in *TIBCO ActiveMatrix Policy Manager Installation*. In particular, see the section Modifying the Agents. Do the task steps in that section (and in other sections of the appendix, as needed).
6. Re-Start the Application server. Refer to section , Starting the Management Agent and Service Probe for IBM WebSphere Application Server in Chapter 3, Tasks.

Chapter 3 Tasks

This chapter presents tasks related to Management agents, WebSphere and Service Probe for IBM WebSphere Application Server.

Topics

Starting the Management Agent and Service Probe for IBM WebSphere Application Server

When a management agent has been properly configured within a WebSphere application server, then starting that application server automatically starts the management agent.

To start an application server, do these steps:

1. `cd` *was_home*`/profiles/`*profile_name*`/bin`
2. Use the `startServer` command (see IBM WebSphere documentation).

To start a Federated Application Server, do these steps:

1. Start Deployment Manager. If not started, do these step:
   - `cd was_home/profiles/deployment_manager_profile_name/bin`
   - Use `startManager` command (see IBM WebSphere Network Deployment documentation)
2. Start the Node Agent. If not started, do these step:
   - `cd` *was_home*`/profiles/`*profile_name*`/bin`
   - Use the `startNode` command (see IBM WebSphere documentation)
3. Start the Federated Application Server.
   - `cd` *was_home*`/profiles/`*profile_name*`/bin`
   - Use the `startServer` command (see IBM WebSphere documentation)

To start the Management Agent without the Service Probe, disable the Service Probe flag from the `pluginagent.properties` file.

To enable and disable the Service probe, set `SERVICE_PROBE_ENABLE=true or false`.

Stopping the Management Agent and Service Probe for IBM WebSphere Application Server

Stopping a WebSphere application server automatically stops its management agent and Service Probe.

To stop an application server, do these steps:

1. `cd` *was_home*`/profiles/`*profile_name*`/bin`
2. Use the `stopServer` command (see IBM WebSphere documentation).

To stop Federated Application Server, do these steps:

1. `cd` *was_home*`/profiles/`*profile_name*`/bin`
2. Use the `stopServer` command (see IBM WebSphere documentation)
3. `cd` *was_home*`/profiles/`*profile_name*`/bin`
4. Use the `stopNode` command (see IBM WebSphere documentation)

When a WebSphere application server deploys a SOAP/JMS service, then the Policy Manager console always indicates that the service is available (green dot)—even when the service is down.

Modifying Application Server Credentials

The utility `ManagementAgentConfig` caches the WebSphere application server's administrative username and password. If these credentials subsequently change, you must explicitly update the following property values in two locations:

- `WEBSPHERE_USERNAME`
- `WEBSPHERE_PASSWORD`

Do these steps:

1. Use the utility *TIBCO_HOME*`/policymanager/3.0/bin/encryptSharedSecret` to encrypt the new password. (This utility is included with the Policy Manager product. For details, see *TIBCO ActiveMatrix Policy Manager User's Guide*)
2. Stop the WebSphere application server.

First Location: Agent Instance

3. In a text editor, open this file:

TIBCO_HOME`/amma-was/instances/`*appsvr_profile_name*`/cells/`*cell_name*`/nodes/`*node_name*`/servers/`*server_name*`/pluginagent.properties`

4. Locate the property key `WEBSPHERE_USERNAME`, and replace the old username with the new value.
5. Locate the property key `WEBSPHERE_PASSWORD` and `SPM_SP_WEBSPHERE_PASSWORD` , and replace the old password value with the new encrypted value.

   The `SPM_SP_WEBSPHERE_PASSWORD` is encrypted using the Tibco Password Obfuscater located under the tibcohome/amxadministrator/<amx version>/bin/passwordobfuscator directory.
6. Save the file.

Second Location: Application Server Profile

7. In a text editor, open the file `IBM/WebSphere/AppServer/profiles/`*appsvr_profile_name*`/cells/`*cell_name*`/nodes/`*node_name*`/servers/`*server_name*`/properties/pluginagent.properties`.
8. Locate the property key `WEBSPHERE_USERNAME`, and replace the old username with the new value.
9. Locate the property key `WEBSPHERE_PASSWORD`, and replace the old password value with the new encrypted value.
10. Save the file.

11. Restart the WebSphere application server.

Modifying Service Probe for IBM WebSphere Application Server Credentials

To explicitly update the property values to modify the Service Probe credentials, do the following:

Edit *TIBCO_HOME*/amma-was/serviceprobe/wasserviceagent.properties filre to provide following information:

For TIBCO ActiveMatrix Administrator Server:

```
amx_admin_server_urls
amx_admin_server_auth_username
amx_admin_server_auth_password
#Custom Metrics Enable/Disable
serviceprobe.custommetrics.enable=true
```

Note that only URLs of ActiveMatrix Administrator servers that exist in the same cluster can be specified for `amx_admin_server_urls`.

For the `amx_admin_server_urls` parameter, you can specify multiple URLs separated by delimiter. The default value of delimiter is "; " (semicolon) that can be changed using the `amx_admin_server_url_delimeter` parameter.

Restart the Management Agent for WebSphere to reflect these changes in the `wasserviceagent.properties` file.

Configuring a JMS-Based JAX-RPC Service

TIBCO Enterprise Message Service™ supports JAX-RPC services deployed in a WebSphere application server. To configure an Enterprise Message Service provider to cooperate with WebSphere 6.1, do these tasks:

Task A JNDI: Configure WebSphere to Use Enterprise Message Service as the JNDI Provider

As a result of this task, both the WebSphere application server and the client container can find the TIBCO Enterprise Message Service `URLConnectionFactory` when encountering the `tibjmsnaming` JNDI naming scheme.

1. In the directory *was_home*`/AppServer/lib/ext`, create a text file called `jndi.properties`.
2. Add the following line into that file:

   ```
   java.naming.factory.url.pkgs=com.tibco.tibjms.naming
   ```
3. Save the `jndi.properties` file.
4. Re-start the WebSphere application servers.

Task B JMS: Configure WebSphere to Use Enterprise Message Service as the JMS Provider

Follow the instructions in *TIBCO Enterprise Message Service Application Integration Guide*, in the chapter "Integrating With IBM WebSphere Application Server Version 6.1."

Removing a Management Agent Instance

To remove a management agent from a WebSphere application server profile, do these steps:

1. Stop the WebSphere application server.

   The application server must be stopped before you begin the next step.

2. Run the utility `RevertManagementAgent`.

   This utility automatically starts the application server, undeploys the management agent, and stops the application server again.

3. Restart the application server (as appropriate).

See Also RevertManagementAgent on page 53

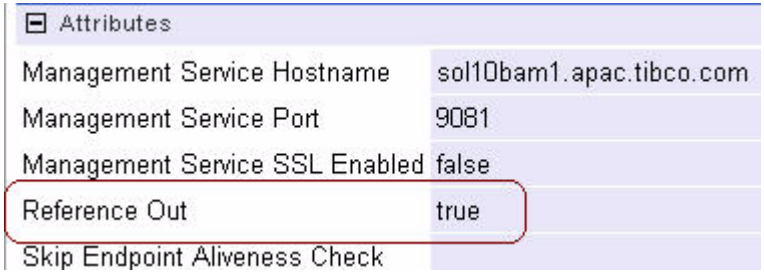
Using Embedded Client Side Proxy Feature

The Embedded Client Side Proxy feature enables security context propagation, credential mapping, and encryption or decryption to external services as well as provides a capability that management agent can be configured as a substitute for proxy agent.

It exists inside the ActiveMatrix Management Agent for WebSphere and exposes external endpoints through “References” and manages them. Therefore, the users can configure the client side policies on them the same way as is done for the proxy agent.

The "Reference Out" attribute is set to "`true`" for Reference endpoints as shown in the following screenshot.

Figure 7 Reference Out Attribute Set to 'True'

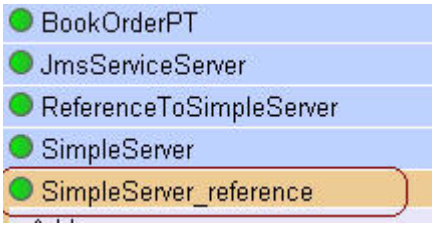


To use this functionality, the users need to modify their web services (that reference other web services) to add a reference binding file `amma-was-config.xml`. For SOAP/HTTP based single/multi hop web services (a web service calling another web service) this file has to be packaged in the web archive under `WEB-INF` folder and for SOAP/JMS this file has to be packaged in the enterprise archive under `META-INF` folder. This binding file contains WSDL URL information of the referenced web service.

The existing single/multi hop web services *must* be redeployed by packaging the reference binding file in order to get the reference endpoints auto managed.

The reference endpoint uses the service name of the referenced web service by adding the suffix “`_reference`”.

Figure 8 Referenced Web Service



Even if multiple web services in one application server references a common web service which is deployed in the same or other application server, only one single Reference endpoint will be exposed in the Policy Manager Console.

If a web service is referencing multiple web services then those many reference endpoints will be created. This solely depends on the number of `WSDL` URLs specified in the reference binding file `amma-was-config.xml`.

Refer to the following sample snippet of the reference binding file:

```
<?xml version="1.0"?>
<amma-was-config>
<!-- WSDL URL for SOAP/HTTP Web Service -->
  <ExternalServicesWsdlUrlList>
<ExternalServiceWsdlUrl>http://hostname:9084/SimpleServer/services
/SimpleHttpPortTypeEndpoint1/wsdl/Simple.wsdl</ExternalServiceWsdl
Url>

<!-- WSDL URL for SOAP/JMS Web Service -->
<ExternalServiceWsdlUrl>http://hostname:9080/tibcoplugin/amberpoin
t?getJmsWsdl&amp;ServiceName=http://xmlns.example.com/121131693344
4:JmsServiceServer&amp;Port=JmsServiceServerPortTypeEndpoint</Exte
rnalServiceWsdlUrl>
  </ExternalServicesWsdlUrlList>

</amma-was-config>
```

Chapter 4 Utilities

This chapter presents command line utilities included with Management Agent for WebSphere software.

Topics

Prerequisites

When running the utilities `ManagementAgentConfig`, `RevertManagementAgent` or `SetSharedSecretForManagementAgent`, you must be logged in as a user that has appropriate privileges to modify file is the *was_home* directory tree.

When running the `ManagementAgentConfig` utility on Websphere profile in a Network Deployment environment, the Deployment Manager for the configured profile must be in a running state along with Node and the Server in a stopped state, otherwise the `ManagementAgentConfig` utility fails.

If a new server is created for a federated profile, ensure that the node agent has been started atleast once.

If a server is running as a Window service, stop and disable it.

To allow `RevertManagementAgent` utility to unmanage the server side WebSphere application server service and agent, when the Referencing web service calls the Referenced web service which are deployed on different WebSphere application server profiles; perform the following steps in this order:

1. Run the `RevertManagementAgent`utility on the profile that hosts the Referenced web service.
2. Then, run the `RevertManagementAgent`utility on the other profile that hosts the Referencing web service.

ManagementAgentConfig

Command Utility

Syntax `ManagementAgentConfig [-console]`

Description Creates and configures a management agent instance in a WebSphere application server.

Location

OS	Utility Location
Windows	*tibco_home*\amma-was\<*version_num*>\bin\ManagementAgentConfig.exe
UNIX	*TIBCO_HOME*/amma-was/<*version_num*>/bin/ManagementAgentConfig

Remarks This utility creates one management agent instance each time it runs to successful completion. It associates that agent with a WebSphere application server, according to the parameters you specify.

For each new management agent instance, you must run this utility to create and configure it. Use values you recorded in Table 5, Management Agent for WebSphere Instances Plan, on page 24.

This utility configures the management agent instance with the Policy Manager host and port. If the location of the central services changes, you must remove the agent instance (you may preserve the `agentcache` directory), and run this utility again to recreate the agent instance.

Log This utility logs its activity to the file *TIBCO_HOME*/amma-was/<*version_num*>/logs/ManagementAgentConfig.log.

This utility deploys the file PolicyAgentManagementService-<*appserver_profile_name*>-<*appserver_cellname*>-<*appserver_nodename*>-<*appserver_servername*>.ear within the WebSphere application server profile. You must *not* modify any configuration parameters of this `.ear` file.

Parameter	Description
`-console`	The utility can operate using a graphical user interface (GUI) or in console mode. When this command line flag is *absent*, the utility executes in the GUI mode, prompting you to enter information in text fields within the pop-up windows. When this command line flag is *present*, the utility executes in Console mode, prompting you to type information on separate lines of the console or alphanumeric interface. Console mode is suitable for configuring agents in situations that do not support GUI mode.

Procedure

1. Refer to Table 5, Management Agent for WebSphere Instances Plan, on page 24, where you recorded the values that you will use in this procedure.
2. On the WebSphere application server host computer, locate the `ManagementAgentConfig` utility, and start it (either double-click its icon, or type its name at a command prompt).

   Enter the parameter values required for configuration. (If you supply the `-console` flag, the prompts appear as text in the command I/O stream, rather than in windows.)
3. Enter the parameter values from Table 5 on page 24. Figure 9, Figure 10 and Figure 11 illustrate the user interface for entering these values.

Figure 9 ManagementAgentConfig: WebSphere Parameters

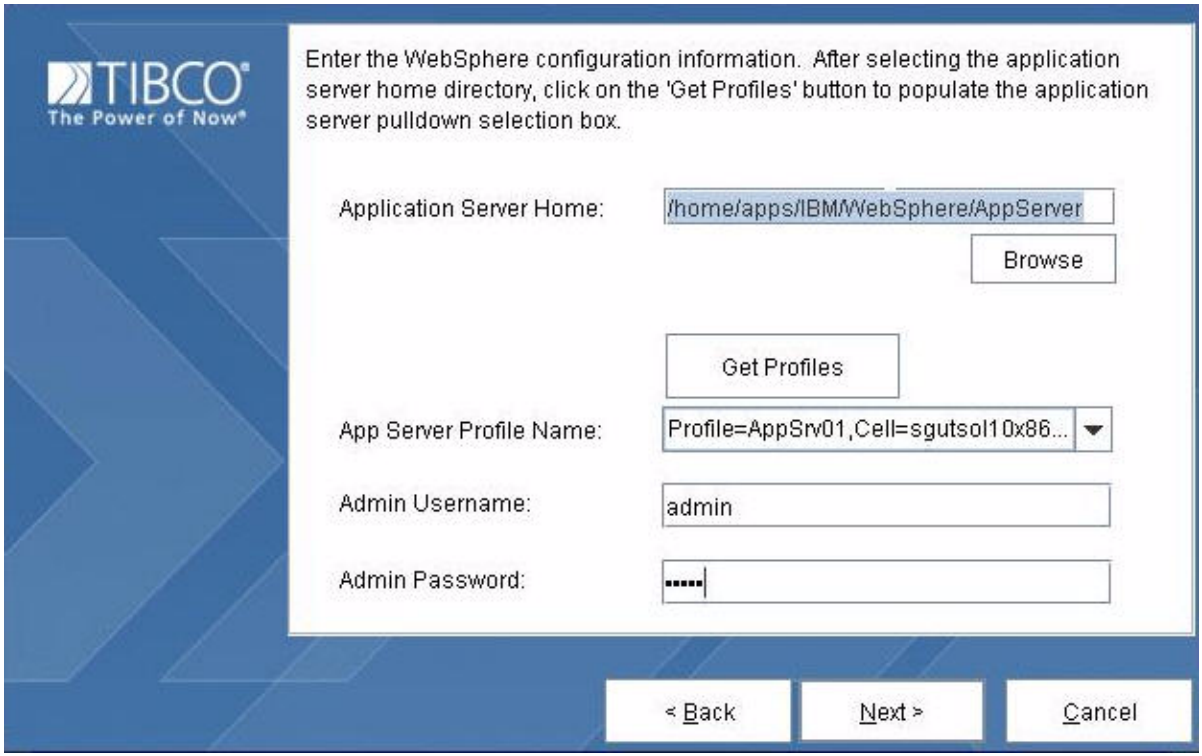


Figure 10 ManagementAgentConfig: Management Agent Parameters

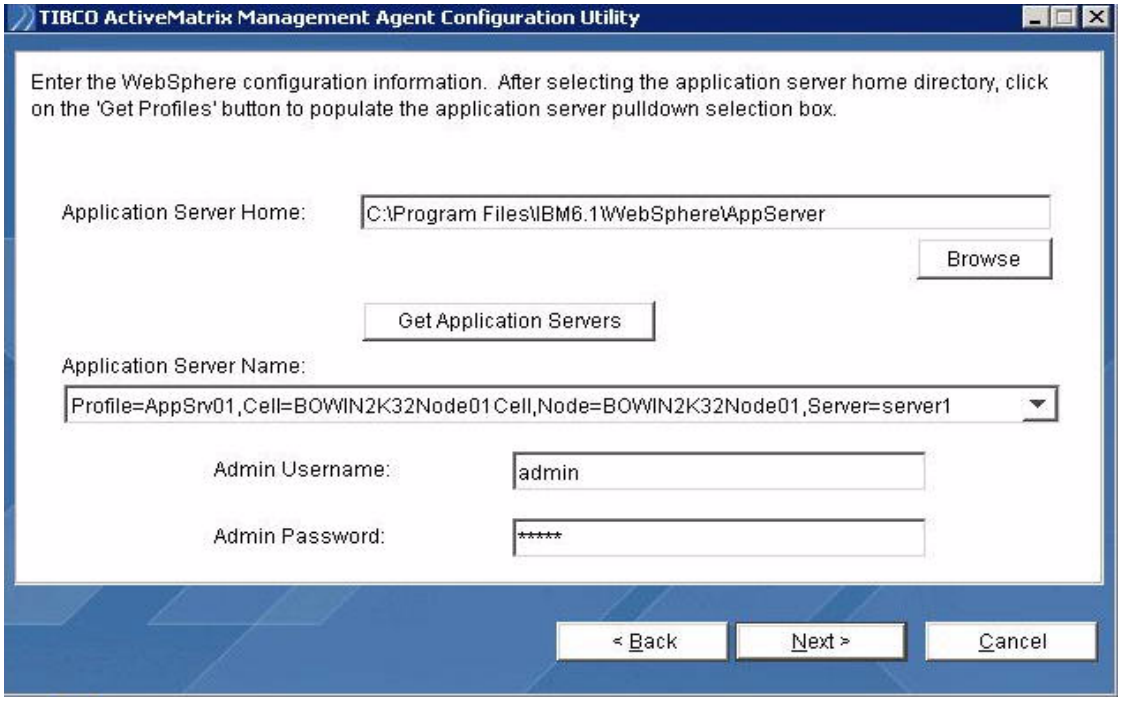


Figure 11 ManagementAgentConfig: Policy Manager Parameters

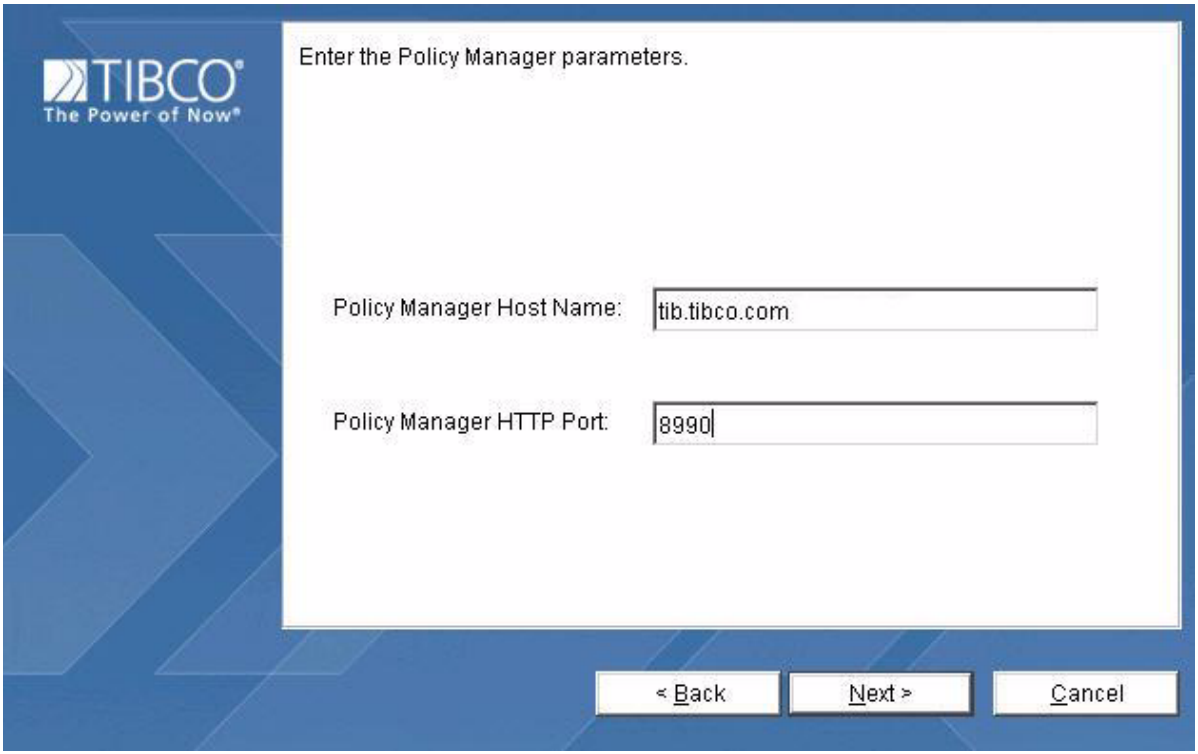


Table 6 ManagementAgentConfig: Agent Instance Parameters (Sheet 1 of 3)

Parameter	Description
WebSphere Parameters	
These parameters pertain to the application server. All parameters are required unless noted. Enter these values in the window shown in Figure 9 on page 47. Use values you recorded in Table 5 on page 24.	
`Application Server Home`	Supply the full directory pathname of installation directory for WebSphere Application Server software; for example, `/IBM/WebSphere/AppServer`.

Table 6 ManagementAgentConfig: Agent Instance Parameters (Sheet 2 of 3)

Parameter	Description
`ApplicationServer Server Name`	The utility configures a management agent for the application server that you select from this drop-down menu. First, collect WebSphere server names into the drop-down menu, by clicking the **Get Servers** button. Then select a server name from the menu; the menu contains only those server names for which agents have not yet been configured.
`Admin Username & Password`	Supply the application server's administrative username and password. ***Note:*** *If the application server is federated, then specify the deployment manager's administrative username and password.* If you did not configure a username and password for the WebSphere application server, you may omit them here as well. (We recommend using administrative passwords to protect production environments.) For more information, see Modifying Application Server Credentials on page 36.
Management Agent Parameters These parameters configure inbound and internal communication for the management agent instance. All parameters are required. Enter these values in the window shown in Figure 10 on page 48. Use values you recorded in Table 5 on page 24.	
`WebSphere Host Name`	Supply the fully-qualified hostname of the WebSphere default host object. (You specified this hostname when you configured the WebSphere application server profile.)
`Policy Manager Connects Using SSL`	When enabled, the agent listens for Policy Manager connection requests using HTTPS (SSL) protocols (and ignores HTTP requests). When disabled, the agent listens for HTTP requests (and ignores HTTPS requests).

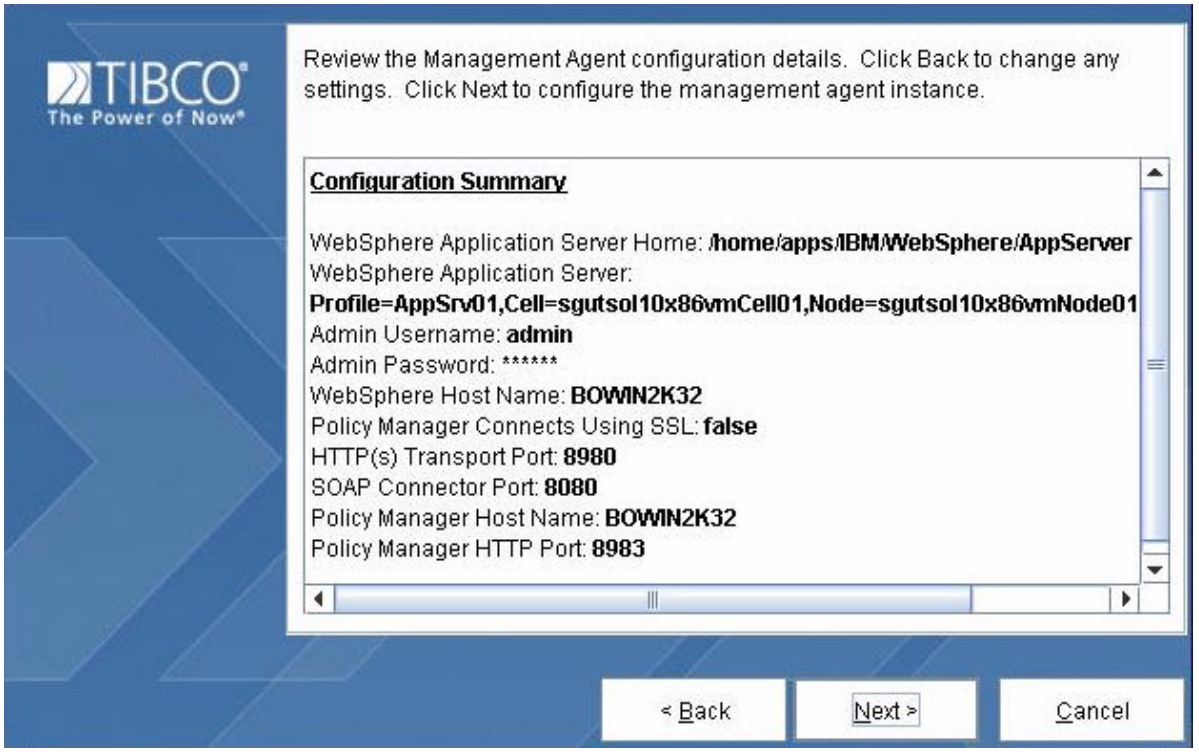
Table 6 ManagementAgentConfig: Agent Instance Parameters (Sheet 3 of 3)

Parameter	Description
HTTP or HTTPS Transport Port	Supply one of two possible port numbers, depending on the value you specified for Policy Manager Connects Using SSL (see above): • When enabled, supply the HTTPS transport port. • When disabled, supply the HTTP transport port. (You specified these ports when you configured the WebSphere application server profile.) Policy Manager connects to the management agent at this port.
SOAP Connector Port	Supply the SOAP connection port. (You specified this port when you configured the WebSphere application server profile.) Note that for a **Federated Application Server**, get this value from the serverindex.xml file. The serverindex.xml file contains an element with a serverType attribute value as "NODE_AGENT". This element consists of a child element with endPointName attribute value as "SOAP_CONNECTOR_ADDRESS". This element has a child element with one of the attribute as port. The value for this attribute must be used as the soap connector port. The management agent connects to a JMX interface at this port to discover the deployed services within the application server.
Policy Manager Parameters These parameters configure outbound communication from the management agent to Policy Manager. All parameters are required. Enter these values in the window shown in Figure 11 on page 49. Use values you recorded in Table 5 on page 24.	
Policy Manager Host Name	Supply the fully qualified hostname of the Policy Manager central services host computer. (You specified this hostname when you configured Policy Manager.)
Connect to Policy Manager Using SSL	When enabled, this management agent connects to Policy Manager using HTTPS (SSL) protocols. When disabled, it connects using HTTP.
Policy Manager HTTP or HTTPS Port	Supply the central services port that you specified when you configured Policy Manager.

4. Review the configuration values, as in Figure 12. Compare them against the plan you recorded in Table 5 on page 24. Make sure they are correct. If you

have made any last-minute changes, update your plan accordingly for future reference.

Figure 12 ManagementAgentConfig: Summary



5. Finish the configuration by clicking the **Next** button until the utility finishes.

See Also Record the Management Agent for WebSphere Configuration Plan on page 20
Record the Policy Manager Installation Plan in *TIBCO ActiveMatrix Policy Manager Installation*

If the `ManagementAgentConfig` fails due to wrong or invalid inputs, re-run it with valid inputs to reconfigure the agent correctly.

RevertManagementAgent

Command Utility

Syntax

```
RevertManagementAgent
    -webSphereHome
     [-profileName] [-cellName] [-nodeName] [-serverName]
     [-allProfiles] | [-deleteAgentCache]
     [-cleanup] | [-deleteAgentCache]
```

Description

Remove the management agent instance from a WebSphere application server (or from all application servers).

Location

OS	Utility Location
Windows	*tibco_home*\amma-was\<*version_num*>\bin\RevertManagementAgent.exe
UNIX	*TIBCO_HOME*/amma-was/<*version_num*>/bin/RevertManagementAgent

Remarks

When argument strings contain internal space characters, surround the strings with double quote characters (").

Log This utility logs its activity to the file *TIBCO_HOME*/amma-was/<*version_num*>/logs/RevertManagementAgent.log.

- To **revert a specific Server,** use the following argument:

```
-webSphereHome   "webSphereHome" -profileName "profileName"
-cellName "cellName" -nodeName "nodeName"
-serverName"serverName" [-deleteAgentCache]
```

- To **revert all Servers for a given Node instance**, use the following argument:

```
-webSphereHome   "webSphereHome" -profileName "profileName"
-cellName "cellName" -nodeName "nodeName" [-deleteAgentCache]
```

- To **revert all Managed Servers from all Profiles** , use the following argument:

```
-webSphereHome   "webSphereHome" -allProfiles
[-deleteAgentCache]
```

- To **delete all Management Agent related files** from Websphere home directory and a mangementAgent instance directory completely, use the following argument:

```
-webSphereHome   "webSphereHome" -cleanup [-deleteAgentCache]
```

The parameter `-deleteAgentCache` is optional in either case . If specified, the utility removes the management agent's data cache directory (in addition to removing the management agent software).

SetSharedSecretForManagementAgent

Command Utility

Syntax

```
SetSharedSecretForManagementAgent
      [-secret] encryptedSecret]
      [-trustDir pathname]
      [-PolicyMgrId instance]
```

Description Arrange the shared secret for Management Agent for WebSphere instances.

Remarks The shared secret enables a set of policy components to communicate with one another, while not interfering with other sets of policy components. This utility arranges the shared secret for management agents within WebSphere application manager instances.

This utility performs a subset of these actions, in this order:

1. The utility copies a template properties files to the trust directory.
2. If the `secret` parameter is present, then the utility stores the encrypted shared secret in the properties file (in the trust directory).
3. The utility adds the location of the trust directory to two WebSphere commands—`setupCmdline` and `startServer`.

Restart After using this utility to set the shared secret, the new secret takes effect when you restart the corresponding WebSphere application server.

Location

OS	Utility Location
Windows	*tibco_home*/amma-was/<*version_num*>/bin/SetSharedSecretForManagementAgent.exe
UNIX	*TIBCO_HOME*/amma-was/<*version_num*>/bin/SetSharedSecretForManagementAgent

(Sheet 1 of 2)

Parameter	Description
`-secret` *encryptedSecret*	Supply the shared secret password (as an encrypted string). When absent, the utility uses the default shared secret (not recommended).

(Sheet 2 of 2)

Parameter	Description
`-PolicyMgrId` *instance*	Specifies a policy manager instance, using the form *host*`:`*port*. When present, the utility updates *only* the local management agents communicating with this policy manager instance. When absent, the utility updates all management agents on the host computer (without regard to policy manager instances).
`-trustDir` *pathname*	The utility copies a template properties file to this location. When absent, this utility copies to the default location (recommended). The default location is *TIBCO_HOME*`/amma-was/.trust/pm-`*host*`-`*port* (where *host* and *port* specify the location of the Policy Manager instance to which the management agent connects). Do not place the trust directory under*TIBCO_HOME*`/amma-was/instances`.

See Also `encryptSharedSecret` in documentation for TIBCO ActiveMatrix Policy Manager

Chapter 5 SSL

Management agents can use SSL to protect communications from Policy Manager. This chapter presents configuration tasks related to SSL.

Topics

Overview

You can configure two levels of SSL communication:

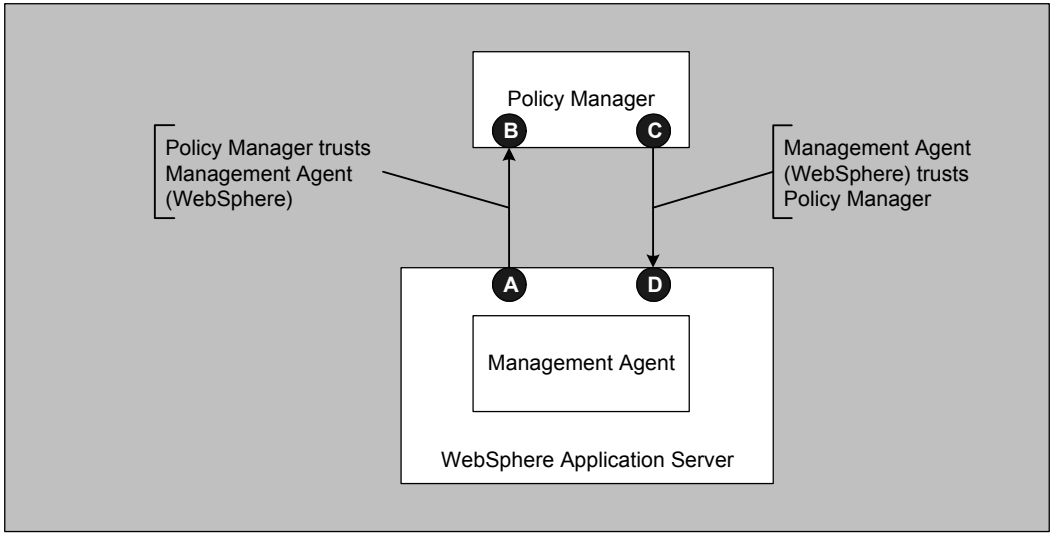
- Policy Manager can use SSL for requests to the management agent in WebSphere. (That is, Policy Manager requires server authentication from WebSphere.)
- WebSphere application server can require client authentication from Policy Manager.
- The management agent in WebSphere can use SSL for requests to Policy Manager. (That is, WebSphere requires server authentication from Policy Manager.)
- Policy Manager can require client authentication from the management agent in WebSphere.

In the following discussion, truststore files may contain either public certificates of the relevant entity, or of a certificate authority.

SSL for Policy Manager Requests to Management Agent

Letter labels (A–E) in Figure 13 correspond to SSL keystore and truststore files.

Figure 13 SSL for Policy Manager Requests to Management Agent



You can prepare these JKS or JCEKS files using any certificate utility (for example, `keytool`).

Policy Manager connects to management agents to deliver policy instructions and to query for log data.

Client Authentication

In addition, if the WebSphere application server requires client authentication, you must prepare two more items:

- Keystore file for Policy Manager containing its identity (C in Figure 13)
- Truststore file for the WebSphere application server, containing Policy Manager's public certificate (D in Figure 13)

WebSphere Documentation

Instruction steps in this chapter require you to configure keystore and truststore information within WebSphere. For details, see IBM WebSphere documentation; such as the following URL:

- http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.iseries.doc/info/iseriesnd/ae/usec_sslkeystore.html

Configuring SSL for Policy Manager Requests

To configure the management agent to enable secure connection from Policy Manager, do these tasks:

Task A Configure management agent to accept SSL requests

When this parameter is enabled, the management agent accepts only HTTPS connections from Policy Manager.

Task B Arrange WebSphere identity keystore

Arrange the WebSphere application server's identity (certificate with private key) in the application server's keystore file. For instructions, see WebSphere Application Server documentation.

The management agent inherits this identity from the application server.

Task C Arrange Policy Manager truststore

1. Arrange the WebSphere application server's public certificate in Policy Manager's truststore.
2. In a text editor, open the file `startPolicyMgr.tra`.
3. Determine whether that file already specifies a truststore using properties like these:

   ```
   java.property.javax.net.ssl.trustStore=truststoreLocation
   java.property.javax.net.ssl.trustStorePassword=truststorePW
   java.property.javax.net.ssl.trustStoreType=truststoreType
   ```

   — If so, modify the truststore, adding the public certificate of the WebSphere application server.

   — If not, do these steps:

   a. Create a truststore containing the public certificate of the WebSphere application server. The truststore file must be accessible from the Policy Manager host.

   b. Modify the file `startPolicyMgr.tra` by adding the three properties above, to specify the new truststore.

Task D Arrange Policy Manager keystore for client authentication

If the WebSphere application server requires client authentication from SSL clients, then you must also ensure proper configuration of Policy Manager's identity (certificate with private key) in Policy Manager's keystore.

1. In a text editor, open the file `startPolicyMgr.tra`.
2. Determine whether that file already specifies an identity keystore, using properties like these:

   ```
   java.property.javax.net.ssl.keyStore=keystoreLocation
   java.property.javax.net.ssl.keyStorePassword=keystorePW
   java.property.javax.net.ssl.keyStoreType=keystoreType
   ```

   — If the file `startPolicyMgr.tra` *does* specify an identity keystore, verify that the keystore meets the requirements listed in the table (below), and then stop (do not complete the remaining steps).

   — If the file `startPolicyMgr.tra` does *not* specify an identity keystore, then do the remaining steps.
3. Create an identity keystore containing the Policy Manager's (central services) identity (certificate with private key). Ensure that the keystore meets the requirements in Table 7 (below).
4. Modify the file `startPolicyMgr.tra` by adding the three properties above, to specify the new keystore.

Table 7 Policy Manager Identity Keystore Requirements

Identity Keystore Requirements
The keystore file must be accessible from the Policy Manager host.
The keystore file must contain exactly one entry.
Tomcat requires that the keystore password is identical to the private key password.

Task E Arrange WebSphere truststore for client authentication

If the WebSphere application server requires client authentication from SSL clients, then you must also arrange Policy Manager's public certificate in the WebSphere application server's truststore. For instructions see WebSphere Application Server documentation.

Chapter 6 Security Context

This chapter presents a compact API for accessing and forwarding security context objects, which are associated with message exchanges.

Topics

Overview

Background

Some security policies attach a security context object to each message exchange. Security context objects can contain user role and attribute information. For example, an authentication policy might obtain user roles from an identity management system (IMS). The provider service might access and use this information. Alternatively, another policy at another policy agent might need this same information, and rather than retrieving it from the IMS a second time, it would be more efficient to re-use the information.

Capabilities

This chapter presents a Java API that allows WebSphere services to access the security context information, and to forward the security context information to another provider for re-use.

Three methods implement this functionality for JAX-RPC services, and three separate methods do the same for JAX-WS services.

Forwarding is always the result of an explicit method call; it is never automatic.

Implementation

These methods arrange to forward security context information in a message header named `com.tibco.security.userinformation`.

Programmer's Checklist for Security Context

Developers of web service programs can use this checklist during the four phases of the development cycle: installing Management Agent for WebSphere software, coding your Java program, compiling your Java program, and deploying your program as a WebSphere service.

Install

Install the Management Agent for WebSphere software release, which automatically includes the Java archive file `com.tibco.policyagent.was.api_`*version*`.jar`, which contains the class `com.tibco.amma.was.security.SecurityContext`.

Code

- Import `com.tibco.amma.was.security`.
- For JAX-RPC services, your endpoint implementation class must do these actions:
  - Define an instance variable of type `ServletEndpointContext`.
  - Implement the `ServiceLifecycle` interface.
  - Its `init` method must initialize the instance variable (of type `ServletEndpointContext`).
- For JAX-WS services, your endpoint implementation class must define a variable of type `WebServiceContext`, using the `@Resource` annotation.

Compile

- The `CLASSPATH` variable must include the archive file directory *TIBCO_HOME*`/components/eclipse/plugins`, to access the archive file `com.tibco.policyagent.was.api_`*version*`.jar`.
- The `CLASSPATH` variable must include the directory *TIBCO_HOME*`/components/eclipse/plugins/com.tibco.policy.agent.lib_version`, to access the archive file `ap-xmlaccess.jar`. <>

TIBCO_HOME`/components/eclipse/plugins/com.tibco.policy.agent.lib_version`

- For JAX-RPC services, the `CLASSPATH` variable must include the directory *was_home*`/runtimes`, to access the archive file `com.ibm.ws.webservices.thinclient_6.1.0.jar`.
- For JAX-WS services, the `CLASSPATH` variable must include the directory *was_home*`/plugins`, to access the archive file `org.apache.axis2_6.1.0.jar`.

Deploy

When you deploy your Java services in a WebSphere application server, you must not disable the WebSphere installation option **Create MBeans for resources** (this option is enabled by default).

Enabling the Security Context API: Administrative Task

`SecurityContext` methods require that the management agent's `forwardUserInfoContextDoc` parameter is set to `true`. You must explicitly set this parameter in each management agent instance that manages a service that uses any `SecurityContext` methods.

For complete instructions, see the task Forwarding User Information in *TIBCO ActiveMatrix Policy Manager User's Guide*.

SecurityContext

Class

Declaration

```
class com.tibco.amma.was.security.SecurityContext
  extends java.lang.Object
```

Purpose Interface methods to access and forward security-related information associated with message exchanges.

Remarks Programs do not create instances of SecurityContext. Instead, web server programs use its static methods to get information from existing message exchange objects, and transfer (forward) security context information to new outbound request messages.

Method	Description	Page
Methods for JAX-RPC Services		
SecurityContext.getAttributesFromJaxRPCContext	Get user attributes from the security context in a servlet endpoint context object.	69
SecurityContext.getRolesFromJaxRPCContext	Get user roles from the security context in a servlet endpoint context object.	70
SecurityContext.setSecurityContextForJaxRPC	Forward the security context object to another web service provider.	71
Methods for JAX-WS Services		
SecurityContext.getAttributesFromJaxWSContext	Get user attributes from the security context in a web service context object.	72
SecurityContext.getRolesFromJaxWSContext	Get user roles from the security context in a web service context object.	73
SecurityContext.setSecurityContextForJaxWS	Forward the security context object to another web service provider.	74

SecurityContext.getAttributesFromJaxRPCContext

Method

Declaration

```
static  Map<String,String> getAttributesFromJaxRPCContext(
    ServletEndpointContext servletEndptContext)
  throws IOException,SAXException
```

Purpose

Get user attributes from the security context in a servlet endpoint context object.

Parameter	Description
`servletEndptContext`	Copy the user attributes from the security context in this servlet endpoint context object.

See Also

For information about `ServletEndpointContext`, see java.sun.com/webservices/docs/1.5/api/javax/xml/rpc/server/ServletEndpointContext.html

SecurityContext.getRolesFromJaxRPCContext

Method

Declaration

```
static  List<String> getRolesFromJaxRPCContext(
    ServletEndpointContext servletEndptContext)
  throws IOException,SAXException
```

Purpose

Get user roles from the security context in a servlet endpoint context object.

Parameter	Description
`servletEndptContext`	Copy the user roles from the security context in this servlet endpoint context object.

See Also

For information about `ServletEndpointContext`, see java.sun.com/webservices/docs/1.5/api/javax/xml/rpc/server/ServletEndpointContext.html

SecurityContext.setSecurityContextForJaxRPC

Method

Declaration

```
static void setSecurityContextForJaxRPC(
    Call call,
    ServletEndpointContext servletEndptContext)
  throws SOAPException
```

Purpose

Forward the security context object to another web service provider.

Remarks

This method copies the security context information from a servlet endpoint context object onto an outbound request message.

Parameter	Description
`call`	Copy the security context to this call object (setting a property of the call).
`servletEndptContext`	Copy the security context from this endpoint context object.

See Also

For information about `ServletEndpointContext`, see java.sun.com/webservices/docs/1.5/api/javax/xml/rpc/server/ServletEndpointContext.html
For information about JAX-RPC Call objects, see http://java.sun.com/j2ee/1.4/docs/api/javax/xml/rpc/Call.html

SecurityContext.getAttributesFromJaxWSContext

Method

Declaration

```
static  Map<String,String> getAttributesFromJaxWSContext(
    WebServiceContext wsContext)
  throws IOException,SAXException
```

Purpose

Get user attributes from the security context in a web service context object.

Parameter	Description
`wsContext`	Copy the user attributes from the security context in this web service context object.

See Also

For information about `WebServiceContext`, see java.sun.com/javase/6/docs/api/javax/xml/ws/WebServiceContext.html

SecurityContext.getRolesFromJaxWSContext

Method

Declaration

```
static  List<String> getRolesFromJaxWSContext(
    WebServiceContext wsContext)
  throws IOException,SAXException
```

Purpose

Get user roles from the security context in a web service context object.

Parameter	Description
`wsContext`	Copy the user roles from the security context in this web service context object.

See Also

For information about `WebServiceContext`, see java.sun.com/javase/6/docs/api/javax/xml/ws/WebServiceContext.html

SecurityContext.setSecurityContextForJaxWS

Method

Declaration

```
static void setSecurityContextForJaxWS(
    SOAPMessage message,
    WebServiceContext wsContext)
  throws SOAPException
```

Purpose Forward the security context object to another web service provider.

Remarks This method copies the security context information from a web service context object onto an outbound request message.

Parameter	Description
`message`	Copy the security context to this *outbound* request message.
`wsContext`	Copy the security context from this web service context object.

See Also For information about `WebServiceContext`, see java.sun.com/javase/6/docs/api/javax/xml/ws/WebServiceContext.html

Custom Metrics on Security Context propagated by TIBCO ActiveMatrix Policy Manager

If you use TIBCO ActiveMatrix Policy Manager for Authentication policies and enable Security Context Propagation (see *ActiveMatrix Policy Manager documentation* for details), you can register custom metrics using ActiveMatrix Service Performance Manager to extract user information forwarded by the Policy.

Authentication Policy

If an authentication policy was applied in the TIBCO ActiveMatrix Policy Manager, refer to the user information is available in the Amberpoint User Information document.

Here is a snippet of the Amberpoint user information document:

```
<?xml version="1.0" encoding="UTF-8"?>
<ap:userInformation
xmlns:ap="http://namespace.amberpoint.com/amf">
<ap:userIdentity>pmuser</ap:userIdentity>
<ap:userRoles />
<ap:clientAddress>10.97.98.163</ap:clientAddress>
<ap:claimedIdentity
xmlns:ap="http://namespace.amberpoint.com/amf"
authenticationMechanism="urn:oasis:names:tc:SAML:2.0:ac:classes:Password"
authenticatorRef="AuthnProvider_794A62E4_9F3A_11DD_9CF5_76AFA2FFAA77"
authenticationStatus="success"
type="com.amberpoint.security.authn.identity.BaseClaimedIdentity"
authenticationProviderHash="244139661">
<ap:userIdentity>pmuser</ap:userIdentity>
<ap:userRoles>
<ap:role name="Accounting Managers"
attributeInfoProviderRef="AttrInfoProvider_794A62E4_9F3A_11DD_9CF5_76AFA2FFAA77" />
<ap:role name="SeanPMGroup2"
attributeInfoProviderRef="AttrInfoProvider_794A62E4_9F3A_11DD_9CF5_76AFA2FFAA77" />
```

```
</ap:userRoles>
<ap:userAttributes />
</ap:claimedIdentity>
</ap:userInformation>
```

You can write the following custom metric expressions to extract classifiers for the user name and role:

```
{{flow=input,document=userInformation}}//ap:userInformation/ap:use
rIdentity
{{flow=input,document=userInformation}}//ap:userInformation/ap:cla
imedIdentity/ap:userRoles/ap:role/@name
```

where the prefix 'ap' is associated with the namespace

```
'http://namespace.amberpoint.com/amf'
```

Custom Metrics Script Example

```
<CustomMetricBundle name="WASSPBookOrderCustomMetricBundle">
<MonitoredObjectRef>
<WASServiceOperation cellName="Machine1Cell01"
applicationName="BookOrderServiceEAR"
serviceName="BookOrderPT"
moduleName= "BookOrderService.war"
serviceInterfaceName="BookOrderPT"
operationName="orderBook">
</WASServiceOperation>
</MonitoredObjectRef>
<NamespacePrefixMap>
<MapEntry prefix="ns0"
namespace="http://www.tibco.com/BookOrderService"/>
<MapEntry namespace="http://namespace.amberpoint.com/amf"
prefix="ap"/>
</NamespacePrefixMap>
<ClassifierCustomMetric name="User" displayName="User"
xpath="{{flow=input,document=userInformation}}//ap:userInformation
/ap:userIdentity">
<NamespacePrefixOverrideMap/>
</ClassifierCustomMetric>
<ClassifierCustomMetric name="BookTitle"
xpath="{{flow=input,document=input}}//ns0:orderBookRequest/bookNam
e">
<NamespacePrefixOverrideMap></NamespacePrefixOverrideMap>
</ClassifierCustomMetric>
```

```
<InstrumentCustomMetric name="QtyOrdered"
xpath="{{flow=input,document=input}}//ns0:orderBookRequest/quantit
y"
keepHistory="true" unit="USD" unitDisplayName="$">
<NamespacePrefixOverrideMap></NamespacePrefixOverrideMap>
<MetricFunction>sum</MetricFunction>
</InstrumentCustomMetric>
</CustomMetricBundle>
```

Appendix A Samples

This appendix presents the details of sample supported by TIBCO ActiveMatrix Management Agent for WebSphere and Service Probe for IBM WebSphere Application Server. It describes the example and presents preliminary tasks that you must complete before running this sample.

A user can use this sample as a reference to run a similar exercise in their own environment or run this sample exercise as is.

Topics

Sample Contents

The sample contains the following items:

- Management Agent for WebSphere service sample "BookOrder"
- BusinessWorks Project as a client
- Security Context API sample

Tasks

Demonstrating the sample involves the following tasks:

- Installing and Configuring the Resource
  - — Configuring Management Agent for WebSphere
- Working with sample
  - — Deploy the Sample Service

    Deploy the sample service in the IBM Admin Console
  - — Apply the Policies to the Sample Service
  - — Send request to service.
  - — Confirm Results

Prerequisites

Before proceeding with the sample execution, ensure the following prerequisites:

- Install Policy Manager
- Install IBM WebShere Application Server
- Install Management Agent for WebSphere software
- Ensure that Management Agent is configured for WebShere Application Server profile on which the sample service is to be deployed

For details, refer to Programmer's Checklist for Security Context, page 65 in Chapter 6, Security Contexts.

Installing the Sample for Management Agent for WebSphere

To install the sample, perform the following steps:

1. TIBCO ActiveMatrix Management Agent for WebSphere installation packages the samples `amma-was_sample.zip`. Extract the `amma-was_sample.zip` file into samples folder. The following screenshot shows the directory structure after extraction.

Figure 14 Directory Structure of Samples

TIBCO ActiveMatrix Management Agent for WebSphere Samples folder structure description is as follows:

- **BookOrderBWClient** - BusinessWorks **Consumer Project** for the BookOrder Service (Refer to the section *Preparing the BusinessWorks Consumers* in the *TIBCO ActiveMatrix Policy Manager Samples*)
- **BookOrderService** - BookOrder JAX-RPC **Web Service project** created using IBM WebSphere Application Server Toolkit version 6.1.0 with latest fix pack.
- **BookOrderServiceEAR**- BookOrder Web Service **EAR project** created using IBM WebSphere Application Server Toolkit version 6.1.0 with latest fix pack.

The description of *BookOrderBWClient*, *BookOrderService* and *BookOrderServiceEAR* is just for reference for the user to understand.

The JAX-RPC based `BookOrderService.ear` file present under the *Deployment-Archive* folder can be directly deployed by the user.

- **Deployment-Archive** - Contains the BookOrder JAX-RPC **Web Service enterprise archive file** that can be deployed in IBM WebSphere Application Server.

  — Importing Policies - Refer to section *Importing Policies* in the *TIBCO ActiveMatrix Policy Manager Samples*)

Import policies definition file `PoliciesDefinition.xml`from the extracted *Samples* folder.

2. Login to IBM Console.
3. Click Install. This will install the sample application.
4. Click Browse and select the `BookOrderService.ear` file located in the Deployment Archive. With all default configurations, click Next.
5. Select the checkbox next to the `BookOrderService` and click Next.
6. Click Finish on the Summary screen that shows the summary of installation options.
7. The next screen shows the status of installation. Make sure that the application is installed successfully and click the Save link.
8. Select the checkbox next to the `BookOrderServiceEar` and click Start button on the Enterprise Applications screen.
9. The message on the next screen shows that application started successfully.
10. Log in to Policy Manager console. You can see that `BookOrderPT` Service is created.

Figure 15 Service Created in Policy Manager Console

Working With Security Context Propagation Sample

This section presents the details about working with installed sample.

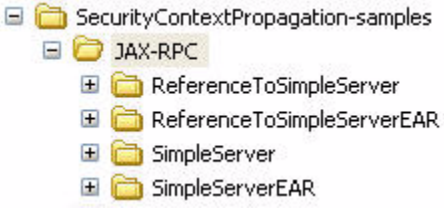
You must set the `forwardUserInfoDoc` flag to `true` on the Management Agent for WebSphere agent to enable Security Context Propagation.

To set the `forwardUserInfoDoc flag` to `true` perform the following steps:

1. From Policy Manager Admin Console, select the "System" tab.
2. Select the agent instance on which the flag needs to be set.
3. Select Edit setup for the agent and edit the xml.
4. Change the `<pfx6:forwardUserInfoContextDoc>false</pfx6:forwardUserInfoContextDoc>` tag to `true`.

- The following screenshot shows the `SecurityContextPropagation-samples/JAX-RPC`:

Figure 16 SecurityContextPropagation - Samples Directory



— **SimpleServer** - `SimpleServer JAX-RPC Web SOAP/HTTP Service` project created using IBM WebSphere Application Server Toolkit version 6.1.0 with latest fix pack.

This service does addition of two numbers.

— **ReferenceToSimpleServer** - `ReferenceToSimpleServer JAX-RPC SOAP/HTTP Web Service` project created using IBM WebSphere Application Server Toolkit version 6.1.0 with latest fix pack.

This service accepts two numbers and calls the `SimpleServer` service which returns the total results of this addition back to the `ReferenceToSimpleServer` service.

To work with the sample, perform the following steps.

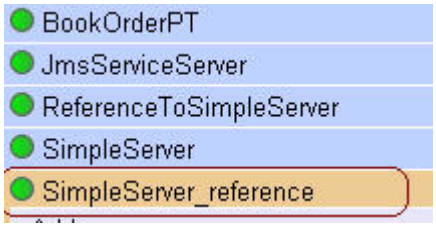
1. Login to the IBM console and Deploy `SimpleServer` Web Service.

It is important to first deploy the target service.

2. `ReferenceToSimpleServer` service calls `SimpleServer` Web service.

   It uses the service name of the referenced web service and appends the suffix "`_reference`" to it in order to distinguish itself from the referenced web service. You can apply Embedded Client Side Proxy to this service.

Figure 17 Referenced Web Service



Using Security Context API

With reference to `ReferenceToSimpleServer` web service project, the following section describes how to develop a web service that uses the Security Context API.

Web Service Implementation

For the Security Context API `setSecurityContextForJaxRPC`(Call call, `ServletEndpointContext servletEndptContext`) to extract the user information document from the `MessageContext` the Web Service needs to implement the *Service Lifecycle* Interface.

Refer the JAX-RPC code snippet below for the correct usage of the `setSecurityContextForJaxRPC`(Call call, `ServletEndpointContext servletEndptContext`) API.

```
/**
 * SimpleHttpPortTypeEndpoint1BindingImpl.java
*/
package com.example.xmlns;
import javax.xml.rpc.ServiceException;
import javax.xml.rpc.Stub;
import javax.xml.rpc.Call;
```

```
import javax.xml.rpc.Service;
import javax.xml.rpc.JAXRPCException;
import javax.xml.namespace.QName;
import javax.xml.rpc.ServiceFactory;
import javax.xml.rpc.ParameterMode;
import javax.xml.rpc.server.ServiceLifecycle;
import javax.xml.rpc.server.ServletEndpointContext;
import com.tibco.amma.was.security.SecurityContext;
public class SimpleHttpPortTypeEndpoint1BindingImpl implements
com.example.xmlns.SimpleHttpPortType, ServiceLifecycle{
private static String endpointAddress =
"http://hostname:9087/SimpleServer/services/SimpleHttpPortTypeEndp
oint1";
private ServletEndpointContext servletEndptContext;
private static String qnameService = "SimpleServer";
private static String qnamePort = "SimpleHttpPortType";
            private static String ENCODING_STYLE_PROPERTY =
"javax.xml.rpc.encodingstyle.namespace.uri";
            private static String NS_XSD =
"http://www.w3.org/2001/XMLSchema";
            private static String URI_ENCODING =
"http://schemas.xmlsoap.org/soap/encoding/";
            public java.math.BigInteger Add(long number1, long
number2) throws java.rmi.RemoteException {
java.math.BigInteger result;
try {
ServiceFactory factory = ServiceFactory.newInstance();
Service service = factory.createService(new QName(qnameService));
QName port = new QName(qnamePort);
Call call = service.createCall(port);
call.setTargetEndpointAddress(endpointAddress);
call.setProperty(Call.SOAPACTION_USE_PROPERTY,    new
Boolean(true)); call.setProperty(Call.SOAPACTION_URI_PROPERTY,
"");
call.setProperty(ENCODING_STYLE_PROPERTY, URI_ENCODING);
...

SecurityContext.setSecurityContextForJaxRPC(call,
servletEndptContext);
System.out.println("ReferenceToSimpleServer invoking SimpleServer
with values (" + number1 + ", " + number2 + ")");
result = (java.math.BigInteger)call.invoke(params);
return result
}
catch (Exception ex) {
ex.printStackTrace();
```

```
return null;
}
}
//  The JAX-RPC runtime  passes the ServletEndpointContext in this
method .
public void init(Object context) throws ServiceException {
this.servletEndptContext = (ServletEndpointContext) context;
}
public void destroy() {
}
}
```

- Using the `SimpleService` project, create the `SimpleService EAR` using IBM WebSphere Application Server Toolkit. Then deploy the `SimpleService EAR`.
- The `endpointAddress` variable highlighted in the snippet must be replaced with the actual `SimpleService` service endpoint url in the `ReferenceToSimpleServer` project code shipped as a sample.
- To use this functionality, the users need to modify their web services (that reference other web services) to add a reference binding file `amma-was-config.xml`.

Snippet of `amma-was-config.xml` for reference:

```
?xml version="1.0"?>
<amma-was-config>
<ExternalServicesWsdlUrlList>
<ExternalServiceWsdlUrl>http://hostname:9087/SimpleServer/services
/SimpleHttpPortTypeEndpoint1/wsdl/Simple.wsdl</ExternalServiceWsdl
Url>
  </ExternalServicesWsdlUrlList>
</amma-was-config>
```

- Create the `ReferenceToSimpleService EAR` using IBM WebSphere Application Server Toolkit.
- The `ReferenceToSimpleServer` project contains `amma-was-config.xml` file under the `WEB-INF` folder. This file needs to be updated with the WSDL URL of the deployed `SimpleService` web service.
- Deploy the `ReferenceToSimpleService EAR`.

  You are now ready to test a `ReferenceToSimpleService` web service calling `SimpleService` web service scenario with security context propagation enabled.

Installing the Sample for Service Probe for IBM WebSphere Application Server

Prerequisites

Before proceeding with the sample execution, ensure the following prerequisites:

- Apache Ant installed
- Ensure that Application Server is in a running state on which the sample service is to be deployed

Deploying the Sample

Perform the following steps to deploy and execute the Service Probe for IBM WebSphere Application Server sample:

1. Edit `book_order_tutorial_data.properties` located at `<TIBCO_Home>/amma-was/<version_num>/samples/` and provide details of Websphere Application Server on which you want to deploy the sample service.
2. From the command locate the samples folder:
   `cd <TIBCO_Home>/amma-was/<version_num>/samples`
3. Run `<Ant_Home>/bin/ant -f book_order_tutorial.xml deploy.`
4. Deploy the `ReBookOrder.ear` located at `<TIBCO_Home>/amma-was/<version_num>/samples` manually from IBM WebSphere Application Server console.

Loading Tutorial Rules, Custom Metrics and Actions in TIBCO ActiveMatrix Service Performance Manager

1. Set the `WAS_CELL_NAME, WAS_MACHINE_NAME, and WAS_SERVER_PORT` in the `TIBCO_SPM_HOME\tutorial\scripts\tutorialsetup_data.properties` file.
2. Replace the following lines in the `TIBCO_SPM_HOME\tutorial\scripts\tutorial_was_spmdata.xml-template`:

```
<ClassifierCustomMetric name="BookTitlesample"
xpath="{{flow=output,document=output}}//ns0:orderBookResponse/ ns0:Book/
ns0:title">
```

```
<InstrumentCustomMetric name="QtyOrderedsample"
xpath="{{flow=input,document=input}}//ns0:orderBookRequest/ ns0:quantity"

<InstrumentCustomMetric name="OrderTotalsample"
xpath="{{flow=output,document=output}}//ns0:orderBookResponse/ ns0:orderTotal"
```

by

```
<ClassifierCustomMetric name="BookTitlesample"
xpath="{{flow=output,document=output}}//ns0:orderBookResponse/Book/title">

<InstrumentCustomMetric name="QtyOrderedsample"
xpath="{{flow=input,document=input}}//ns0:orderBookRequest/quantity"

<InstrumentCustomMetric name="OrderTotalsample"
xpath="{{flow=output,document=output}}//ns0:orderBookResponse/orderTotal"
```

3. To load rules, custom metrics and actions, run the following command from the `TIBCO_SPM_HOME\tutorial\scripts` folder:

   ```
   TIBCO_SPM_HOME\bin\spmcmdline.bat tutorial_build.xml was
   ```

Undeploying the Sample

To undeploy the sample:

1. From command prompt locate the samples folder.

   ```
   cd <TIBCO_Home>/amma-was/<version_num>/samples
   ```

2. Run `<Ant_Home>/bin/ant -f book_order_tutorial.xml undeploy.`

Index

M

P

R

S

T

U

W