TIBCO Hawk® JMX Plug-in

User's Guide

Software Release 2.0
February 2006

Important Information

Preface

TIBCO Hawk JMX Plug-in extends the managing and monitoring capabilities of TIBCO Hawk into applications instrumented using Java™ Management Extensions (JMX) standard interfaces. Via the TIBCO Hawk JMX Plug-in, the TIBCO Hawk agent can connect to JMX Agents, introspect them, and expose the JMX MBeans to the TIBCO Hawk infrastructure as microagents. From there, they can be utilized with TIBCO Hawk rulebases or accessed by any number of TIBCO Hawk console applications.

Topics

Related Documentation

This section lists documentation resources you may find useful.

TIBCO Product Documentation

The following documents form the Product documentation set:

- *TIBCO Hawk JMX Plug-in User's Guide* : Read this manual to gain an understanding of installing and using TIBCO Hawk JMX Plug-in.
- *TIBCO Hawk JMX Plug-in Release Notes*: Read this manual for information on the current release of TIBCO Hawk JMX Plug-in.

Other TIBCO Product Documentation

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

- *TIBCO Hawk® Installation and Configuration Guide*: This manual contains step-by-step instructions for installing TIBCO Hawk software on various operating system platforms.
- *TIBCO Hawk® Admistrator's Guide*: This manual includes basic descriptions of TIBCO Hawk concepts, instructions for using TIBCO Hawk Display, monitoring strategies with examples, a comprehensive FAQ, and a glossary.
- TIBCO Rendezvous™ documentation:This is TIBCO's real-time transport layer that is used by Product software.
- TIBCO Enterprise Message Service ™ documentation: TIBCO Enterprise Message Service software allows you to send messages from your applications in a format that conforms to the Java Messaging Service (JMS) protocol.

Third Party Documentation

You may find the following third-party documentation useful.

- *JMX in Action* by Sullins and Whipple.
- *MX4J Users Guide - http://mx4j.sourceforge.net/docs/index.html*

How to Contact TIBCO Customer Support

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

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

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

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

  http://support.tibco.com

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

Contents

Chapter 1 Introduction

TIBCO Hawk JMX Plug-in extends the functionality of TIBCO Hawk by exposing JMX MBeans to the TIBCO Hawk infrastructure as microagents.

This chapter provides a brief overview of JMX and TIBCO Hawk JMX Plug-in software.

Topics

JMX Overview

As software applications grow, managing them becomes a complex task. So far Java applications had to write their own applications to monitor and manage their applications. JMX (Java Management eXtenstions) is a developing standard that allows Java application developers to integrate their applications with existing management solutions. JMX defines an architecture and API for writing JMX objects, also called *MBeans*, which live inside a container that is defined by the standard. The JMX standard additionally defines a standard method for dynamically augmenting Java objects with management attributes and operations allowing to become dynamically manageable. Functionally, JMX provides similar instrumentation capabilities to TIBCO Hawk's Application Management Interface (AMI) and TIBCO Hawk Console API.

In addition to providing interfaces for application developers, JMX has also been adopted by many of the application server and web server vendors to provide visibility and control into the processes.

About JMX architecture

JMX is designed around a three-level architecture. This design promotes an optimized development framework by enabling different segments of the developer community to focus on the level that best fits their strengths.

The three levels of JMX are defined as:

- Instrumentation

  Resources, such as applications, devices, or services, are instrumented using MBeans. MBeans expose their management interfaces, composed of attributes and operations, through a JMX agent for remote management and monitoring.

- Agent

  The main component of a JMX agent is the *MBean server*. This is a core managed object server in which MBeans are registered. A JMX agent also includes a set of services for handling MBeans. JMX agents directly control resources and make them available to remote management applications. All interactions with an MBean must be performed through interfaces on the MBean server.

- Distributed services (Remote Management)

  Protocol adaptors and standard connectors make a JMX agent accessible from remote management applications outside the JMX agent's Java Virtual Machine (JVM).

MBeans

An MBean is a Java class that meets certain naming and inheritance standards of the JMX specification. The Java class of an MBean exposes the resource to be managed directly through its attributes, operations, and notifications.

The management interface of an MBean consists of:

- Named and typed attributes that can be either read, write, or both.
- Named and typed operations that can be invoked
- Notifications that can be emitted by the MBean

MBeans resides with the MBean server, which is a special MBean that manages a group of MBeans.

JMX Agent

A JMX agent is a Java process that provides a set of services for managing a set of MBeans. It is the container for an MBean server. JMX agents provide services for creating MBean relationships, loading classes (M-Let service), simple monitoring services and timers.

JMX Remote API and JMX Connectors

The JMX Remote API allows remote access to a JMX API MBean server by means of a JMX connector. The protocol used for the standard connector defined by the JMX Remote API is based on Remote Method Invocation (RMI). The JMX Remote API allows remote access to a JMX API MBean server by means of a JMX connector. The JMX Remote API specification describes how you can advertise and find JMX API agents using existing discovery and lookup infrastructures.

The JMX specification defines a connector as an object that makes a Java Management Extensions (JMX) API MBean server accessible to remote Java clients. The client end of a connector exports essentially the same interface as the MBean server. A connector consists of a connector client and a connector server. A connector server is attached to an MBean server and listens for connection requests from clients. A connector client is responsible for establishing a connection with the connector server. A connector client will usually be in a different Java Virtual Machine (JVM) from the connector server, and will often be running on a different machine. A connector server can establish many concurrent connections with different clients. A connector client is connected to only one connector server. A client application can contain many connector clients connected to different connector servers. There can be more than one connection between a given client application and a given server.

Generic Connector

An optional part of the JMX Remote API is a generic connector. The JMX Messaging Protocol (JMXMP) connector is a configuration of the generic connector where the transport protocol is based on TCP.

RMI Connector

This standard RMI protocol must be supported by every implementation that conforms to the JMX Remote API standard. The RMI connector provides a simple mechanism for securing and authenticating the connection between a client and a server. This mechanism provides a basic level of security for environments using the RMI connector. Note that the generic JMXMP connector provides a more advanced level of security.

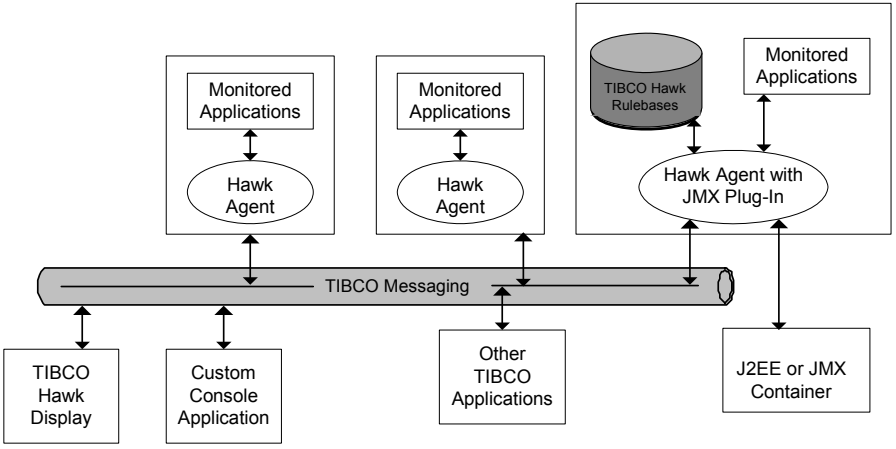
This section describes the JMX 1.2 specification along with JSR 160 (JMX Remoting API). MX4J 3.0.1 implements this specification, but BEA WebLogic Application Server 7.x and JBoss Application Server 3.2.x continue to implement the older JMX 1.0 specification. Applications running in a different JVM connect to BEA WebLogic Application Server 7.x and JBoss Application Server 3.2.x using native RMI implementation.

TIBCO Hawk JMX Plug-in

TIBCO Hawk software is a tool for monitoring distributed systems and applications. In a TIBCO Hawk environment, agents on each local computer perform the monitoring work. TIBCO Hawk agents use microagents to represent and interact with managed objects, such as an application or operating system component. The TIBCO Hawk JMX Plug-in (plug-in), which is deployed as a Microagent Plug-in in a TIBCO Hawk Agent, is a RMI client which is responsible for establishing connection with the connector server component of the MBean server.

The MBeans belonging to the accessed MBean server are exposed to the TIBCO Hawk environment as microagents. TIBCO Hawk Agents use microagents to represent and interact with managed object, which in the case of the plug-in are MBeans. The plug-in maps the MBean's interfaces to the corresponding microagent's methods. Microagent methods can be accessed and invoked from TIBCO Hawk Display or Console API applications. They are also used with the agent's rules engine to provide autonomous monitoring.

The following diagram shows the interaction between TIBCO Hawk, the JMX container, and other TIBCO applications.



The TIBCO Hawk Display application is the primary means of interacting with managed objects on your network. The main window of the TIBCO Hawk Display graphically represents system behavior with each monitored computer represented by a container icon.

Each MBean is represented by a TIBCO Hawk microagent on the local TIBCO Hawk agent.

Supported Application Servers

TIBCO Hawk JMX Plug-in supports the following application servers:

- BEA WebLogic Application Server 7.0, 8.0, and 9.0
- JBoss Application Server 3.2.x (implementing JMX 1.0)

  Uses *RMI over jnp* to connect to an MBean server.
- Tomcat 4.1.x (JMX 1.2.1 Final Release using RMI connection server) and Tomcat 5.x (JMX 1.2.1 Final Release using RMI connection server) via the MX4J classes. In order to use Tomcat with TIBCO Hawk JMX Plug-in, you need to use MX4J 3.0.1. See Using MX4J 3.0 with Tomcat Application Servers on page 11 for details.
- IBM WebSphere 6.0
- JVM 5

Chapter 2 Installation and Configuration

This chapter explains how to install and configure TIBCO Hawk JMX Plug-in.

This software is available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. Please see `readme.txt` file for the availability of this software version on a specific operating system platform. The `readme.txt` for TIBCO Hawk JMX Plug-in is located in the directory `<TIBCO_HOME>\hawk\plugin\jmx`.

Topics

Installation Prerequisites

Before starting the installation procedure, review the topics in this section to determine that your system meets the basic requirements and that you have the prerequisite software installed.

Supported Platforms

The following table lists the platforms supported by TIBCO Hawk JMX Plug-in.

Table 1 Supported Platforms

OS	Hardware Platform
Microsoft Windows 2000 Service Pack 3 (build 2195) Microsoft Windows XP (Version 2002 SP2) Microsoft Windows Server 2003Enterprise Edition for server-class hardware, Standard Edition for desktops.	x86
Solaris 8 Solaris 9 Solaris 10	SPARC (32- and 64-bit)
Solaris 10	x86 (32- and 64-bit)
HP-UX 11.x	PA-RISC
HP-UX 11.2	IA64 (Itanium)
AIX 5.2 AIX 5.3	IBM POWER
LINUX (kernel 2.4.21, glibc 2.2 and up, kernel 2.4.21, glibc 2.3 and up, and kernel 2.6.5, glibc 2.3 and up)	x86

Required Software

TIBCO Hawk 4.5.0 or higher. is required and must be installed before starting the installation procedure.

Installation Procedure

Table 2 lists the platform-specific installation packages, where *<version_num>* is the TIBCO Hawk software release number.

For example, the installation package file name for TIBCO Hawk JMX Plug-in 2.0 on Solaris 8 (Sun SPARC) is `TIB_hawkjmx-simple_2.0.0_s4_57.bin.`

Table 2 Installation Package Filenames

OS	Package Filename
Microsoft Windows 2000 Microsoft Windows XP Microsoft Windows Server 2003	`TIB_hawkjmx-simple_<version_num>_w32.exe`
Solaris 8 (SPARC) Solaris 9 (SPARC) Solaris 10 (SPARC)	`TIB_hawkjmx-simple_<version_num>_s4_57.bin`
Solaris 10 (x86)	`TIB_hawkjmx-simple_<version_num>_i86_57.bin`
HP-UX 11.x (PA-RISC)	`TIB_hawkjmx-simple_<version_num>_h7_110.bin`
HP-UX 11.2 (IA64)	`TIB_hawkjmx-simple_<version_num>_h7_ia64.bin`
AIX 5.2 AIX 5.3	`TIB_hawkjmx-simple_<version_num>_rs_51.bin`
LINUX	`TIB_hawkjmx-simple_<version_num>_lnx86_24.bin`

Follow these steps to install TIBCO Hawk JMX Plug-in:

1. Refer to Table 2, Installation Package Filenames for the installation package appropriate for your platform.
2. Invoke the installer.

   On Microsoft Windows, double-click the executable located in that directory

   `TIB_hawkjmx-simple_<version_num>_w32.exe`

   On UNIX, type the following command

`./<package name>`, where `<package name>` is the name of the package appropriate for your platform.

For example, for Solaris 8 (Sun SPARC),

```
./TIB_hawkjmx-simple_2.0.0_s4_57.bin
```

3. Review and accept the license agreement.
4. Choose the TIBCO Hawk directory where the plug-in will be installed.

   OnMicrosoft Windows, accept the default values.

   On UNIX, choose the TIBCO Hawk installation directory where you want to install the plug-in.
5. Review the installation summary and click Finish to complete the installation process.

Post-Installation Tasks

Perform the following post-installation tasks as needed.

Update the TIBCO Hawk configuration

If this is not the first plug-in used by TIBCO Hawk, this section should be skipped.

- Update the MicroAgent Plugins option (Microsoft Windows platform only)

  On the Microsoft Windows platform, use the *Update the MicroAgent Plugins* option using TIBCO Hawk Configuration Utility. To use the Configuration Utility:

  — Select **Start>Programs>TIBCO>TIBCO Hawk>Hawk Configuration**.

  — Navigate to the *Agent Configuration* tab.

  — Update the *Plugins* option to specify the directory that will be searched when loading the microagent plugins. For more information, see *TIBCO Hawk Installation and Configuration*.

- Update the *hawkagent.cfg* file (UNIX platform only)

  Within the *hawkagent.cfg* file, update the `-hma_plugin_dir` option. This option specifies the directory that will be searched when loading the microagent plugins. For more information, see *TIBCO Hawk Installation and Configuration*.

Upgrading from a previous release

If upgrading from Hawk JMX Plug-in 1.0 and using Tomcat application server, the `type` specified in the `<MBeanServer>` element of the configuration file should be changed from `MX4jRMI` to `JSR160`.

The configuration file `JMXPluginConfig.xml` used by the previous release is renamed `JMXPluginConfig.xml.bkp`.

Using MX4J 3.0 with Tomcat Application Servers

In order to use Tomcat application servers with TIBCO Hawk JMX Plug-in, you need to upgrade to MX4J 3.0 (implementing JMX 1.2). The MX4J distribution can be downloaded from *http://sourceforge.net.*

If upgrading from Hawk JMX Plug-in 1.0 and using MX4J2.0, it is not required to upgrade to MX4J3.0.

Follow these steps to use MX4J 3.0:

1. Stop the application server.
2. Remove or rename <*TOMCAT_HOME*>/server/lib/mx4-jmx.jar (Old MX4J 1.1 distribution), where <*TOMCAT_HOME*> is the application server installation directory.
3. From MX4J3.0 distribution copy mx4.jar, mx4j-remote.jar and, mx4-tools.jar to <*TOMCAT_HOME*>/common/lib
4. Restart the application server to make sure that MBean Server starts up.

Uninstallation Procedure

This section describes how you can uninstall TIBCO Hawk JMX Plug-in.

Microsoft Windows

Use one of the following options to uninstall TIBCO Hawk JMX Plug-in:

- Click **Start>Programs>TIBCO>TIBCO Hawk <*version_num*>> Plug-in-JMX>Uninstall.**

  Where, <*version_num*> refers to the TIBCO Hawk software version number.

- Navigate to the <*TIBCO_HOME*>`\hawk\plugin\jmx\_uninst` directory and invoke the `Tibuninstall.exe` program.

UNIX

Navigate to the <*TIBCO_HOME*>`/hawk/plugin/jmx/_uninst` directory and invoke the `Tibuninstall.bin` program.

Configuration

The plug-in uses the `<TIBCO_HOME>/hawk/plugin/JMXServiceMA.hma` file to obtain the location of the configuration file that will be used by the plug-in. By default, this configuration file is named `JMXPluginConfig.xml`.

Use the XSD file, `JMXconfiguration.xsd`, located in the `<TIBCO_HOME>/hawk/plugin` directory to validate the configuration file before using it.

Appendix A on page 43 lists sample configuration files.

The following table describes the top level *<JMXConfiguration>* element of the *JMXPluginConfig.xml* configuration file.

JMXConfiguration
MethodTimeout The method timeout value in seconds. Default value is 10 (seconds).
PluginDirectory The location of the JMX plug-in directory. The default location is *<TIBCO_HOME>*`/hawk/plugin`.
TraceDirectory Location of the trace directory. The default location is *<TIBCO_HOME>*`/hawk/plugin/jmx/log`.
TraceFilename The name of the trace file. The default name is `JMXPlugin.log`.
AddMeanServerName If this property is set to `true` the Hawk microagent name for every MBean discovered by the plug-in will include the MBeanServer name. The default value is `false`.
TraceLevel The trace level used by the plug-in. The trace level is expressed as the sum of any of the following valid values: -1 (All), 0 (always), 1 (Info), 2 (Warning), 4 (Error), and 8 (Debug). The default value is 7 (which is Info+Warning+Error).

JMXConfiguration

MBeanServerList

The list of MBean servers to which the plug-in connects to.

MBeanServer

This element provides connection information for a specific MBean server.

- name: The name given to the MBean server
- type: The type of the MBean server.

  Valid values are

  `JSR160`, (for Tomcat, JVM, BEA WebLogic 9.0)

  `JBoss3`, (for JBoss 3.2.x)

  `WebSphere` (for IBM WebSphere 6.0)

  and, `WebLogic` (for BEA WebLogic 7.0 and 8.0).
- JMXClassPath: The class path to the vendor specific MBean server jar files.

When connecting to multiple MBean servers, this element has to be replicated with one <*MBeanServer*> element for each MBean server.

ParameterList

The list of parameters used to connect to the MBean Server.This element consists of name-value pairs.

For information on customising this element for the supported application server, see Customising the MBeanServer Element on page 15. Additional parameters can be specified using name-value pairs.

Customising the MBeanServer Element

The `<MBeanServer>` element of the configuration file contains connection information for each MBean sever. This filter is a list, delimited by ';', of the MBeans that you are interested in exposing as TIBCO Hawk microagents.The following sections describe in details in the information required by each supported servers.

JSR 160 compliant App Servers: Tomcat, WebLogic 9 and JVM 5

The plug-in provides two different JSR 160 implementations. One for Tomcat and WebLogic 9 application servers and another for JVM 5.

After installing the MX4J libraries, the The *<JMXClassPath>* element should be defined as:

```
<MX4J_HOME>/lib/mx4j.jar;<MX4J_HOME>/lib/mx4j-remote.jar
```

where, *<MX4J_HOME>* is the location of the MX4J libraries.

If using JVM 5 and if the TIBCO Hawk agent is running within JVM 5, the *<JMXClassPath>* element can be defined as

```
<JMXClassPath>""</JMXClassPath>
```

This is because the necessary classes are included in the JVM 5 runtime libraries.

To connect to the Application server you need to specify:

- `JMXServiceURL`, the connection URL to connect to the MBean server.
- `login and` password, which are the used to connect to the application server.
- `MBeanFilter`, which is the list of MBean(s) that you are interested in. If the `MBeanFilter` has a null value or if the MBeanFilter element is not used, all MBeans are accessed.

Example

```
<MBeanServer name="Mx4jRMI1234" type="JSR160">
  <JMXClassPath>C:/mx4j301/lib/mx4j.jar;C:/mx4j301/lib/mx4j-remote
.jar</JMXClassPath>
  <ParameterList>
    <param name="JMXServiceURL" value="localhost" />
    <param name="login" value="0" />
    <param name="password" value="" />
    <param name="MBeanFilter"
value="Catalina:type=Connector,service=Tomcat-Standalone,port=6789
,address=null;Catalina:type=Connector,service=Tomcat-Standalone,po
rt=8009,address=null" />
  </ParameterList>
</MBeanServer>

<!--Following is an example for JVM 5 -->

<MBeanServer name="J2SE5JVM" type="JSR160">
  <JMXClassPath>C:/mx4j20/lib/mx4j.jar;C:/mx4j20/lib/mx4j-remote.j
ar</JMXClassPath>
  <ParameterList>
    <param name = "JMXServiceURL" value =
"service:jmx:rmi:///jndi/rmi://localhost:6580/jmxrmi"/>
    <param name = "login" value = ""/>
    <param name = "password" value = ""/>
    <param name = "MBeanFilter" value = "java.lang:*"/>
   </ParameterList>
</MBeanServer>
```

JBoss App Server

The *<JMXClassPath>* element should be defined as:

```
C:/<JBOSS_HOME>/lib/jboss-jmx.jar;C:/<JBOSS_HOME>/client/jbossall-
client.jar;C:/<JBOSS_HOME>/client/jmx-connector-client-factory.jar
```

where, *<JBOSS_HOME>* is your JBOSS installation directory.

To connect to the JBoss server you need to specify:

- `JNDIUrl`, which is the location of the JNDI name server (JBoss Naming Server).
- `MBeanFilter`. which is the list of MBean(s) that you are interested in.

Example

```
<MBeanServer name="JBOSS1" type="JBoss3">
  <JMXClassPath>C:/JBOSS323/jboss-3.2.3/lib/jboss-jmx.jar;C:/JBOSS
323/jboss-3.2.3/client/jbossall-client.jar;C:/JBOSS323/jboss-3.2.3
/client/jmx-connector-client-factory.jar</JMXClassPath>
  <ParameterList>
    <param name = "JNDIUrl" value = "jnp://localhost:3099"/>
    <param name = "MBeanFilter" value = "MletAgent:*"/>
  </ParameterList>
</MBeanServer>
```

BEA WebLogic 7.0 and 8.0 App Servers

The *<JMXClassPath>* element should be defined as:

`C:/<WEBLOGIC_HOME>/server/lib/weblogic.jar`

where, *<WEBLOGIC_HOME>* is your WebLogic installation directory.

To connect to the BEA WebLogic server you need to specify:

- `JNDIUrl`, which is the location of the JNDI name server.
- `User`, which is the username used to connect to the server.
- `Password`, which is the encrypted password for a valid username used to connect to the server. See Encrypting Password on page 18 for details on encrypting your password.
- `MBeanFilter`, which is the list of MBean(s) that you are interested in.

Example

```
<MBeanServer name="WebLogic71" type="WebLogic">
  <JMXClassPath>C:/bea7/weblogic700/server/lib/weblogic.jar</JMXCl
assPath>
  <ParameterList>
    <param name="JNDIUrl" value="t3://localhost:7001" />
    <param name="User" value="admin" />
    <param name="Password" value="#!S7cxdb4VKZKt9uXpwsCB5mD9MWdB"
/>
    <param name="MBeanFilter"
value="JMImplementation:type=MBeanServerDelegate;mydomain:Name=mys
erver,Server=myserver,Type=JTARecoveryService" />
  </ParameterList>
</MBeanServer>
```

IBM WebSphere App Server

The *<JMXClassPath>* element when using IBM WebSphere contains the location of the required classes and should be defined as:

<WEBSPHERE_HOME>/lib;*<WEBSPHERE_HOME>*/java/jre/lib;*<WEBSPHERE_HOME>*/java/jre/lib/ext

where, *<WEBSPHERE_HOME>* is your WebSphere installation directory.

To connect to the IBM WebSphere server you need to specify:

- `ConnectorType`, the value of this parameter should be set to `RMI` or `SOAP`.
- `ConnectorHost`, which is the host where the server is running.
- ConnectorPort, which is the port used for connecting to the server.
- `MBeanFilter`, which is the list of MBean(s) that you are interested in.

If connecting to IBM WebSphere using `ConnectorType` as `RMI`, the TIBCO Hawk Agent should be started using IBM JVM.

Example:

```
<MBeanServer name="WebSphere1" type="WebSphere">
  <JMXClassPath>I:/lib;I:/java/jre/lib;I:/java/jre/lib/ext</JMXCla
ssPath>
  <ParameterList
    <param name = "ConnectorType" value = "SOAP"/>
    <param name = "ConnectorHost" value = "hk-win07.na.tibco.com"/>
    <param name = "ConnectorPort" value = "8880"/>
    <param name = "MBeanFilter" value = "WebSphere:*"/>
  </ParameterList>
</MBeanServer>
```

Encrypting Password

Follow these steps to encrypt your password:

1. Invoke the script from the command line using the following syntax. The scripts in located in the directory `<TIBCO_HOME>/hawk/bin`

   On Microsoft Windows, `tibhawkpassword.exe -encrypt <string>`

   On UNIX, `tibhawkpassword.sh -encrypt <string>`

   where *<string>* is the password string you want to encrypt.

2. Copy and paste the output of the script into the configuration file.

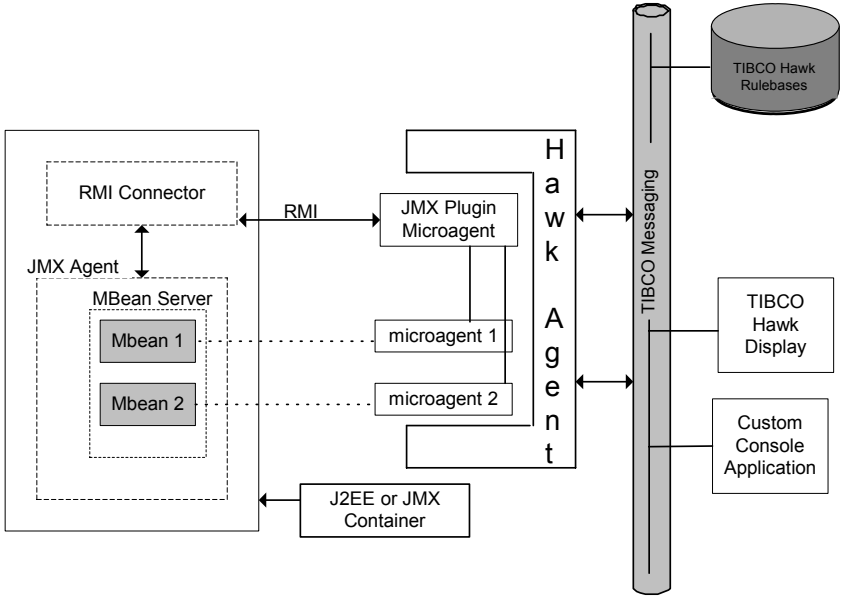
Chapter 3 Using TIBCO Hawk JMX Plug-in

This chapter describes how to use the plug-in software. Additionally, methods exposed by the microagent which represent the plug-in are described. These methods can be used to monitor the plug-in.

Topics

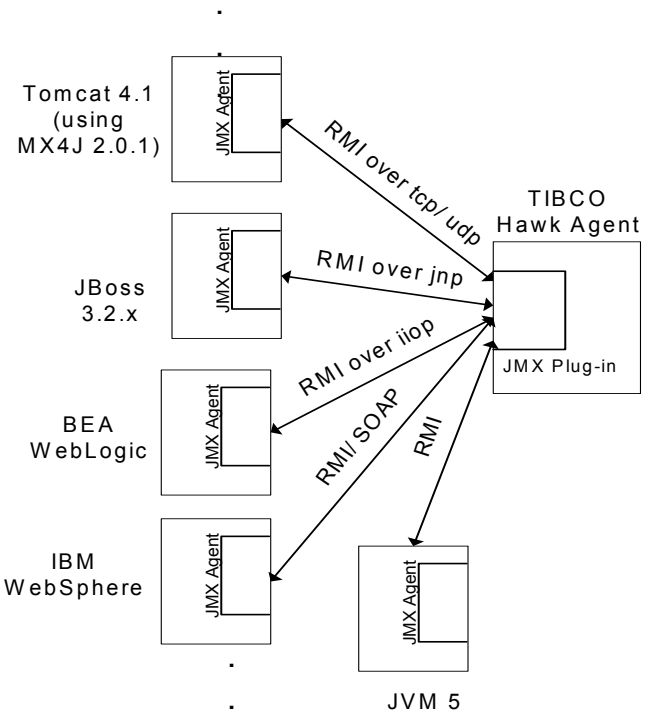
Overview

The TIBCO Hawk JMX Plug-in uses the information contained in the `JMXPluginConfig.xml` configuration file to connect to the MBeanServer(s). If the plug-in is not able to connect to the MBean server, it indefinitely attempts to make the connection. On successfully connecting to the MBean server, the plug-in uses the filters set in the configuration file to access MBeans belonging to the container. Each MBean that is accessed is exposed as a TIBCO Hawk microagent along with the MBean's interfaces. See Mapping MBeans to Microagents, page 22 for more details. The following diagram show in details the interaction between TIBCO Hawk and the JMX environment.



Whenever a new MBean is loaded in the MBean server and meets the filter criteria, the corresponding TIBCO Hawk microagent and methods are created dynamically. Similarly, all TIBCO Hawk microagent and methods created for a MBean are removed when that MBean is unloaded. If the plug-in looses connection to the MBean server, all TIBCO Hawk microagent and methods created for MBeans belonging to the MBean server are removed and are re-created when the connection is re-established. The plug-in continues to keep track of the MBean server after loosing connection to the MBean server.

The plug-in can be configured to access several instances of each supported application server by specifying multiple instances of the `<MBeanServer>` element in the configuration file. See Configuration on page 14 for more information. The following figure shows the protocols used by the plug-in to connect to the supported application servers.



Mapping MBeans to Microagents

The TIBCO Hawk JMX Plug-in uses information contained in the configuration file to connect to the MBeanServer(s). Using the filters set in the configuration file, MBeans within the container are accessed and each MBean is converted into a microagent. The microagent is given the same name as the MBean.

MBean Attributes

MBean attributes can be of two types

- read-only attributes, or
- read-write attributes.

For each read-only attribute, the plug-in creates a corresponding synchronous *get* method.

For each read-write attribute, the plug-in creates 2 corresponding methods, a *get* method and a *set* method.

The names of the methods follow the convention:

`_<operation type>MBean Attribute name.`

where `<operation type>` is either `get` or `set`.

For example,

for a read-write attribute `acceptCount`, the two methods created are

- `_getacceptCount`, and
- `_setacceptCount`.

MBean Operations

Each MBean operation is mapped into a synchronous method which has the same name as that of the operation. Unlike microagent methods which are derived from MBean attributes, microagent methods corresponding to MBean operations do not have a _ (underscore) prefix.

The JmxPluginMicroAgent Microagent

In addition to creating microagents for each of the accessed MBeans, the plug-in creates another microagent, the `JmxPluginMicroAgent`. `JmxPluginMicroAgent` is the JMX service plug-in microagent which acts as a *wrapper* for all the microagents created by the plug-in. This microagent helps keep track of the MBean servers monitored by TIBCO Hawk. The microagent methods provided by the plug-in can be used to monitor the MBean servers. For details about the methods, see Available Microagent Methods on page 25.

MBeanServerDelegate

`MBeanServerDelegate` is a special MBean created by the MBean server and represents the MBean server from the management point of view. The Plug-in registers itself as a listener with the `MBeanServerDelegate`.The MBeanServerDelegate MBean emits MBeanServerNotifications each time an MBean is registered or unregistered. In addition to providing notification information, this MBean provides information pertaining to the implementation of the MBean server.

The microagent corresponding to this MBean is not created by default.

Using Notifications

MBeans communicate with registered listeners using notifications. These notifications communicate changes in the state of the MBeans or the occurrence of certain events. In addition to these notifications, the plug-in sends notifications each time an MBean is added to or removed from the MBean server that is being monitored and if there is a change in the status of the connection to the MBean server.

The TIBCO Hawk microagent for any MBean that is capable of broadcasting notifications (by implementing the `javax.management.NotificationBroadcaster` interface) has an additional TIBCO Hawk microagent method, `onMBeanNotification`. See page 32 for details of this microagent method. All notifications sent by the MBean are mapped to this asynchronous method.

The notifications sent when a MBean is added or removed are sent for all MBeans within the container regardless of the filters used. These notifications are useful in monitoring applications for which no microagent is created. For example, you could have an application A which depends on application B, but application B does not have a corresponding TIBCO Hawk microagent. In case application B goes down, the notification sent communicating this condition could be used to by a rulebase to set an alert condition.

Available Microagent Methods

This section describes the methods exposed by the `JMXPluginMicroagent`. These methods can be used as any other TIBCO Hawk method. They can be invoked directly from TIBCO Hawk Display or used within a rulebase. The following methods are available for monitoring and managing the plug-in.

Method	Description
JmxPluginMicroAgent:getMBeanCount	Returns a table listing the MBean Server and it's MBean Count
JmxPluginMicroAgent:onNotification	Sends an asynchronous message when a new MBean is registered or unregistered. This method has been deprecated.
JmxPluginMicroAgent:onMBeanServerNotification	Sends an asynchronous message when a new MBean is registered or unregistered.
JmxPluginMicroAgent:onConnectionNotification	Sends an asynchronous message when there is a change in the status of the connection to the MBeanServer(s).
JmxPluginMicroAgent:getLoadedMBeans	Returns a table listing all loaded MBeans from the MBean Server.
JmxPluginMicroAgent:getReleaseVersion	Returns version information about the current plug-in release.

JmxPluginMicroAgent:getMBeanCount

Method

Returns a table listing the MBean Server and its MBean count.

Type Synchronous, IMPACT_INFO

Arguments

Parameter	Type	Description
MBeanServerName	String	The user specified MBean server name. If the MBeanServerName is not specified, information for all loaded MBeans is returned.

Returns

Parameter	Type	Description
MBeanServer Connection URL	String	The JNDIUrl for the MBean server.
MBeanServerName	String	The user specified MBean server name. This name is specified in the configuration file.
MBeanCount	Integer	The number of MBeans loaded from the specified MBean server.

JmxPluginMicroAgent:onNotification

Method

Sends an asynchronous message when a new MBean is registered or unregistered. Please note that this method has been deprecated.

Type Asynchronous, IMPACT_INFO

Arguments None.

Returns

Parameter	Type	Description
MBeanServer	String	The MBean server name
Name	String	The name of the MBean
Type	String	The type of notification
Message	String	The MBean message
SeqNum	Long	The sequence number
Source	String	The source
TimeStamp	String	The time stamp

JmxPluginMicroAgent:onMBeanServerNotification

Method

Sends an asynchronous message when a new MBean is registered or unregistered.

Type Asynchronous, IMPACT_INFO

Arguments

Parameter	Type	Description
MBeanServerAliases	String	The list of MBean servers separated by ";", from which you want to receive messages when a new MBean is either registered or unregistered. If the MBeanServerAlias is not specified, messages are sent for all MBeans registered and unregistered with all accessed MBean servers.
ObjectNameFilter	String	The list of ObjectName/ObjectName prefixes separated by ";". If the ObjectNameFilter is not specified, send all notifications.

Returns

Parameter	Type	Description
MBeanServerAlias	String	The MBean Server Name
MBeanName	String	The name of the MBean
TimeStamp	String	The time stamp of the notification.
Type	String	The type of notification
UserData	String	The user data
SeqNum	Long	The sequence of the notification
Message	String	The message
Source	String	The source of the notification.

JmxPluginMicroAgent:onConnectionNotification

Method

Sends an asynchronous message when there is a change in the status of the connection to the MBeanServer(s).

Type Asynchronous, IMPACT_INFO

Arguments

Parameter	Type	Description
MBeanServerAlias	String	The list of MBean servers separated by ";", from which you want to receive messages when there is a change in the status of the connection. If the MBeanServerAlias is not specified, status change messages are sent for all MBeans servers.
TypeFilter	String	The list of Types or Type prefixes separated by ";" used to filter notifications. If no filter is supplied, all notifications are sent.

Returns

Parameter	Type	Description
MBeanServerAlias	String	The MBeanServer Alias
MBeanName	String	The name of the MBean
TimeStamp	String	The time stamp of the notification.
Type	String	The type of Notification
UserData	String	The user data
SeqNum	Long	The sequence of the notification
Message	String	The message
ConnectionID	String	The connection id
Source	String	The source of the message

JmxPluginMicroAgent:getLoadedMBeans

Method

Returns a table listing all loaded MBeans from the MBean Server.

Type Synchronous, IMPACT_INFO

Arguments

Parameter	Type	Description
MBeanServerName	String	The user specified MBean server name.

Returns

Parameter	Type	Description
Name	String	The name of the MBean.
Display name	String	The display name of the MBean.
MBeanServer Connection URL	String	The connection URL for the MBean server.
MBeanServerName	String	The user specified MBean server name.

JmxPluginMicroAgent:getReleaseVersion

Method

Returns version information about the current release of the plug-in.

Type Synchronous, IMPACT_INFO

Arguments None

Returns

Parameter	Type	Description
Name	String	The release name.
Version	String	The release version. Same as the concatenation of the major, minor and update fields with periods (.) between.
Date	String	The release date.
Major	String	The version major number.
Minor	String	The version minor number.
Update	String	The version update number.

onMBeanNotification

Method

This method is created for each microagent whose corresponding MBean is capable of emitting notifications.

This method is only supported on JVM 5 and the Tomcat, BEA WebLogic 9.0, JBoss 3.2.x, and IBM WebSphere 6 application servers.

Type Asynchronous, IMPACT_INFO

Arguments

Parameter	Type	Description
TypeFilter	String	Send notifications for given type/type prefixs separated by ";". If the type is not supplied, all notifications are sent.

Returns

Parameter	Type	Description
TimeStamp	String	The time stamp.
Type	String	The type of notification.
UserData	String	The user data.
SeqNum	Long	The sequence of the notification
Message	String	The notification message.
Event	String	The notification class name.
Source	String	The source of the notification.

Chapter 4 Using The Sample Application

This chapter describes the sample application packaged with the plug-in.

Topics

Overview

The sample application contains a servlet for registering and unregistering MBeans. The sample application can be used only with Tomcat application server.

Before invoking the servlets, use TIBCO Hawk Display to subscribe to the `JMXPluginMicroAgent:onMBeanServerNotification` method. Subscribing to this method brings up a *Subscription Results* window. Initially this window is empty, but each time an MBean is either registered or unregistered, the corresponding message is displayed here.

Before using the samples, please make the steps outlined in the section Using MX4J 3.0 with Tomcat Application Servers on page 11 and the following section have been carried out.

1. Stop the Tomcat Server.
2. Copy the `jmxweb.war` file from `<TIBCO_HOME>/hawk/plugin/jmx/examples/TomcatMX4J2.0` to `<TOMCAT_HOME>/webapps` directory. This war file contains the servlet which locates the Tomcat MBean Server, which is MX4J 2.0 compliant and starts the RMI connection server attached to the MBean server.
3. Restart the Tomcat server.

   Once the server starts, it will unjar the war file which includes the configuration file `web.xml` that is used by the sample application. If you modify any of the following configuration parameters you will need to re-start the web server and additionally make appropriate changes to the configuration file used by the plug-in:

   — `RMIConnServerPortNum`: RMI Connection Server Port Number

   — `RMIRegistryPortNum`: RMI Registry Port Number

   — `HttpHostName`: Http Adaptor Host Name

   — `HttpAdaptorPortNum`: Http Adaptor Port Number

   For details on updating the configuration file used by the plug-in, see Customising the MBeanServer Element on page 15.

Registering an MBean

This section describes the sample application which when invoked registers an MBean `MyDomain:name=index<number>:0`, where *<number>* is a whole number generated by the servlet. Make sure that the `<MBeanFilter>` element of the configuration file used by the plug-in includes the pattern for the MBeans created by the sample. For example, the `<MBeanFilter>` element could contain the string `MyDomain:*`.

Follow these steps to use the sample application to register an MBean:

1. Open a web brower and in the *Address* field type the command

   `http://<host-name>:<port-number>/jmxweb/JmxServlet41`

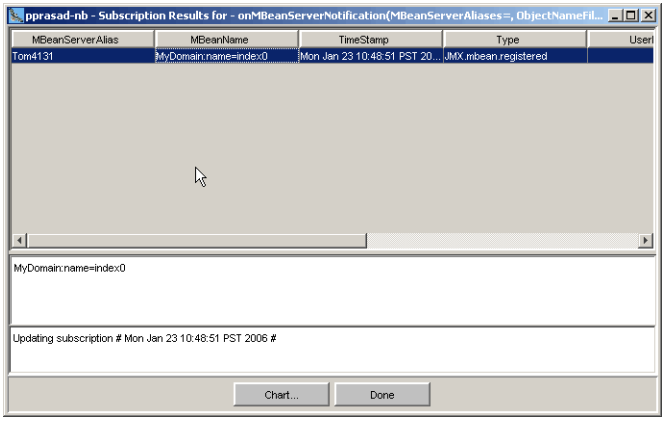
   where *<host-name>* is where the tomcat server is running, and *<port-number>* the port number that the server uses.

   For example,

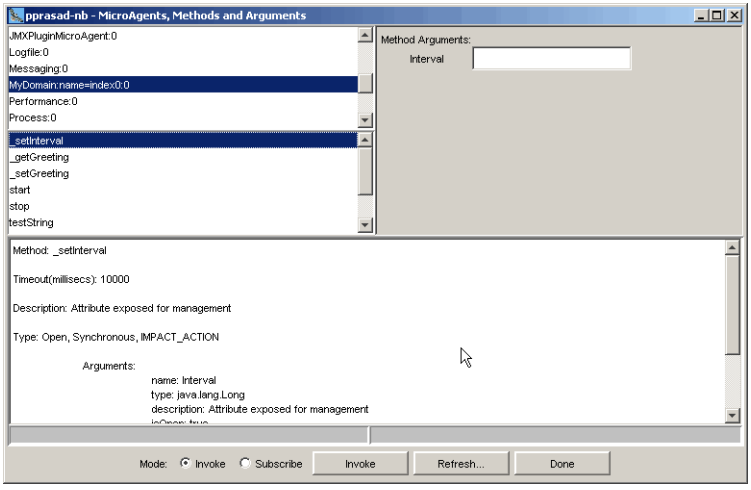
   `http://hk-win05:4567/jmxweb/JmxServlet41`,

2. Click the *Refresh* button on your browser window.

A new MBean is created. The *Subscription Results* window displays the registration message for the new MBean as shown in the following figure.



A microagent for this MBean is dynamically created. If the filters set in the configuration file includes the above named MBean, it can be accessed using TIBCO Hawk Display. The following figure shows the microagent corresponding to the MBean created by the servlet.



Unregistering an MBean

Follow these steps to use the sample application to unregister an MBean:

1. Open a brower and in the *Address* field type the command:

   `http://<host-name>:<port-number>/jmxweb/JmxServlet41?Remove=MyDomain:name=index<number>`

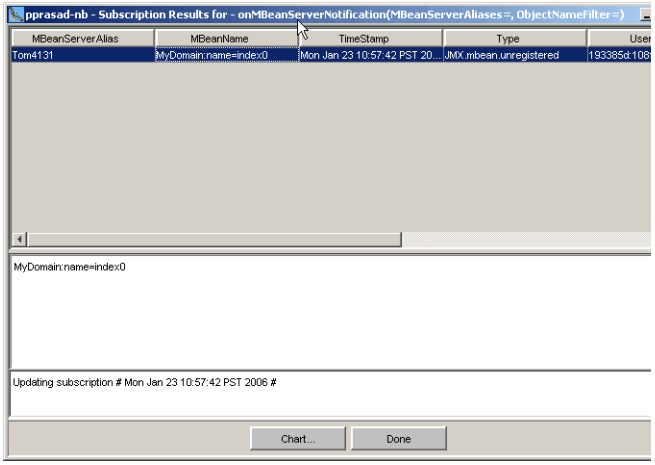
   where *<host-name>* is where the tomcat server is running and *<port-number>* the port number that the server uses and *<number>* corresponds to the MBean you want to unregister.

   For example,

   `http://hk-win05:4567/jmxweb/JmxServlet41?Remove=MyDomain:name=index1`

2. Click the *Refresh* button on your browser window.

The message corresponding to the MBean just unregistered is displayed in the *Subscription Results* windows, as shown in the following figure.



The microagent that was created for MBean is dynamically removed and is no longer displayed in the TIBCO Hawk Display screen.

Building a Rulebase

This section describes the steps involved in creating a rulebase which subscribes to MBean registration notifications and generates an TIBCO Hawk alert when when a MBean is registered.

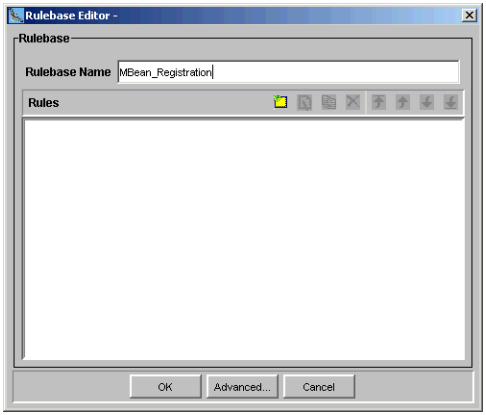
To create a new rulebase in TIBCO Hawk Display:

1. Right-click on the agent icon where the plug-in is running and select **Get Configuration>Rulebases**.

   The Rulebase List for Agent dialog displays. This dialog lists rulebases loaded by the current agent and provides options for performing rulebase operations.

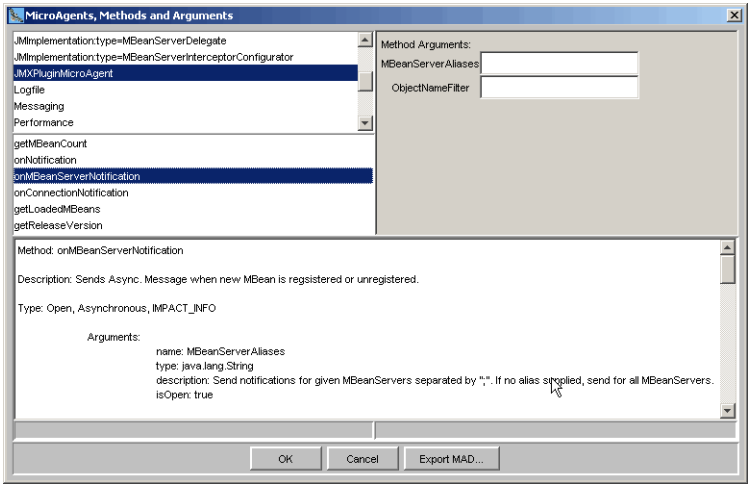
2. Click the **Create** button.
3. Type **MBean_Registration** in the **Rulebase Name** field.

   The Rule Editor dialog displays:



4. Click the **Create** icon on the toolbar.
5. Click the **Edit...** button next to the Data Source field.

The following figure shows the Microagents, Methods and Arguments dialog with the `JMXPluginMicroAgent` selected.



6. Select the **JMXPluginMicroAgent** microagent.

   The methods for this microagent are displayed in the lower panel of the screen.

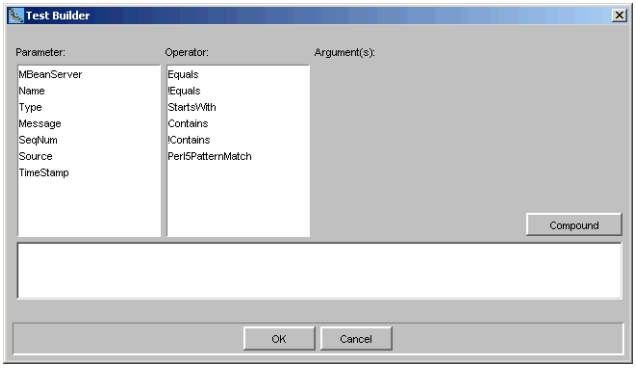
7. Select the **onMBeanServerNotification** method and click **OK**.

   This method send an asynchronous message when a new MBean is registered or unregistered.

8. Click the **Create** icon on the toolbar to create a new test.

9. Click the **Edit...** button next to the **If** field.

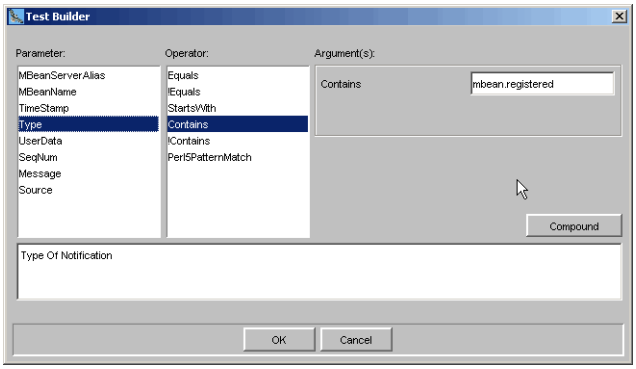
   The Test Builder dialog displays:



10. Select **Type** as the test parameter.

This is one of the result fields returned by the `onMBeanServerNotification` method, which is the data source for this rule. The text area below the parameter list displays a short description of each parameter.

11. Select the Contains operator and type `register` in the Contains field as shown in the following figure:



12. Click OK to save the test.
13. In the Test Editor dialog, click the Create button on the toolbar.
14. Click in the Messages field and type the following:

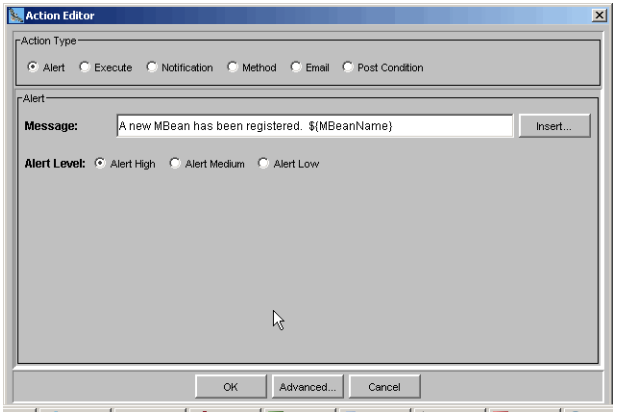
    ```
    A new MBean has been registered.
    ```

15. Click the Insert button and select Name.

    This appends the MBean name to the alert message.

16. For alert level, click **Alert High**.

    The dialog should look like the following:



17. Click **OK** four times (to exit Action, Test, Rule and Rulebase Editors).
18. The **MBean_Registration** rulebase is included in the Rulebase List for Agent dialog on the adapter machine.
19. Click **Apply Changes** to save the rulebase on the agent where the plug-in is running.

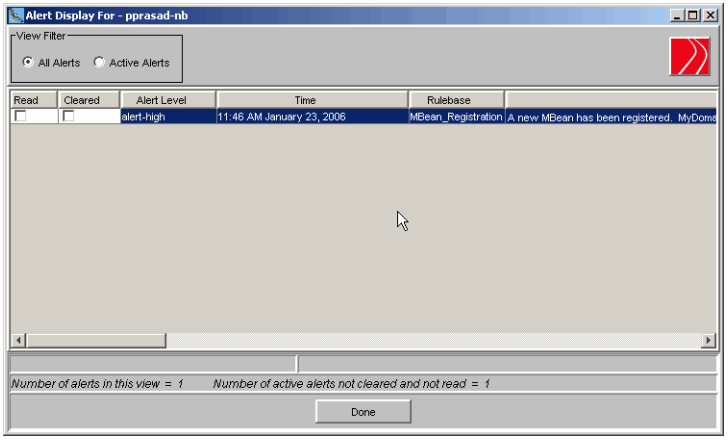
Viewing the Results

This section uses the sample application (for Tomcat application server) bundled with the plug-in to create a new MBean. But the rulebase created in the previous section will be triggered and will generate an alert message on receiving the registration message each time an MBean is registered with the MBean server when using the other supported application servers as well.

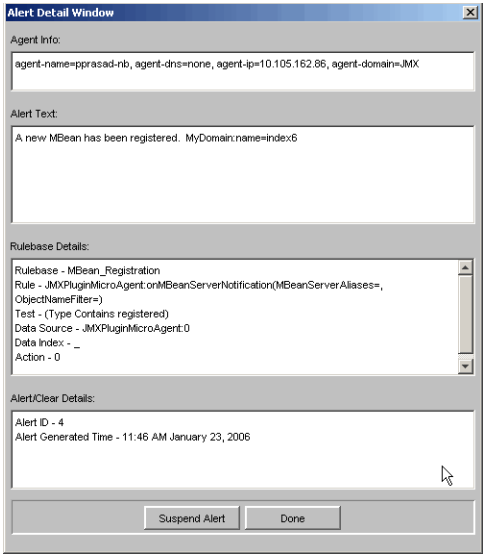
Follow the steps outlined in Registering an MBean on page 35 to register a new MBean. When the new MBean is registered an alert message is generated by the **MBean_Registration** rulebase. To view this alert:

1. Right-click on the agent icon where the adapter is running.
2. Select Show Alerts from the menu.

   The Alert Display window shows the alert generated by the **MBean_Registration** rulebase:



3. Double-click on the alert to display detailed properties.



4. Click **Done** to close the Alert Detail window.

Appendix A Sample Configuration Files

This appendix lists sample `JMXService.hma` and `JMXPluginConfig.xml` configuration files.

Topics

JMXServiceMA.hma

```
<!--
   Copyright 2004-2006 TIBCO Software Inc. All Rights Reserved.
  This software is the confidential and proprietary information of
  TIBCO Software Inc.
-->
<microagent>
<!-- The classname of the microagent -->
  <classname>COM.TIBCO.hawk.jmx.JMXServiceMA</classname>
<!-- The arguments to be passed to the
MicroAgent.initializeMicroAgent() method -->
  <arguments>
    <arg>C:/tibco/hawk/plugin/JMXPluginConfig.xml</arg>
  </arguments>

<!-- The classpath (a list of jar files) containing
the classes required for implementation. If a simple
jar file name is used (i.e., not a fully qualified name)
then the jar file must reside in the same directory as this
.hma file -->

  <classpath>
    <path>JMXServiceMA.jar</path>
    <path>TIBCrypt.jar</path>
  </classpath>
</microagent>
```

JMXPluginConfig.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<JMXConfiguration xmlns:xsi =
"http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation = "JMXconfiguration.xsd">
  <!-- Method Timeout Value in Seconds -->
  <MethodTimeout>10</MethodTimeout>
  <!-- JMX Plugin Dir Location -->
  <PluginDirectory>c:/tibco/hawk/plugin</PluginDirectory>
  <!-- Trace Dir Location -->
<TraceDirectory>c:/tibco/hawk/plugin/jmx/log</TraceDirectory>
  <!-- Trace File Name  -->
  <TraceFilename>JMXPlugin.log</TraceFilename>
  <!-- If this property is set to "true" the Hawk microAgent name
for every MBean discovered by the plugin will include the
MBeanServer name.
For e.g., if the MBean name is "MBean1" its Hawk microagent name
would be "MBean1-MBeanServerName" -->
  <AddMBeanServerName>false</AddMBeanServerName>
  <!-- Trace Level  -->
  <!-- ALL = -1, ALWAYS=0, INFO=1, WARNING=2, ERROR=4 DEBUG=8 -->
  <!-- Default is INFO + WARNING + ERROR -->
  <TraceLevel>7</TraceLevel>

  <!-- First MBean Server's Properties -->
  <!-- Following is sample MX4J JMX properties -->
  <MBeanServerList>
   <MBeanServer name="Mx4jRMI1234" type="JSR160">
   <JMXClassPath>C:/mx4j20/lib/mx4j.jar;C:/mx4j20/lib/mx4j-remote.
jar</JMXClassPath>

    <ParameterList>
    <!-- the address of the connector server to connect to. -->
     <param name = "JMXServiceURL" value =
"service:jmx:rmi://localhost/jndi/jrmp"/>
     <!-- login/password to connect to the connector server. -->
     <param name = "login" value = ""/>
     <param name = "password" value = ""/>
     <!-- MBean filter value -->
     <param name = "MBeanFilter" value =
"JMImplementation:type=MBeanServerDelegate;MyDomain:*"/>
    <!-- a set of attributes to determine how the connection is
made. for example:
    <param name = "java.naming.factory.initial" value =
"com.sun.jndi.rmi.registry.RegistryContextFactory"/>
    <param name = "java.naming.provider.url" value =
"rmi://localhost:6431"/>
     -->
    </ParameterList>
   </MBeanServer>

   <MBeanServer name="J2SE5JVM" type="JSR160">
```

```
   <JMXClassPath>C:/mx4j20/lib/mx4j.jar;C:/mx4j20/lib/mx4j-remote.
jar</JMXClassPath>
    <ParameterList>
    <!-- the address of the connector server to connect to. -->
    <param name = "JMXServiceURL" value =
"service:jmx:rmi:///jndi/rmi://localhost:6580/jmxrmi"/>
    <!-- login/password to connect to the connector server. -->
    <param name = "login" value = ""/>
    <param name = "password" value = ""/>
    <!-- MBean filter value -->
    <param name = "MBeanFilter" value = "java.lang:*"/>
    </ParameterList>
   </MBeanServer>

   <MBeanServer name="JBOSS1" type="JBoss3">
   <JMXClassPath>C:/jboss-3.2.3/lib/jboss-jmx.jar;C:/jboss-3.2.3/c
lient/jbossall-client.jar;C:/jboss-3.2.3/client/jmx-connector-clie
nt-factory.jar</JMXClassPath>
    <ParameterList>
    <!--  JBOSS Naming Server Location  -->
    <param name = "JNDIUrl" value = "jnp://localhost:1099"/>
    <--  MBean Filter Value  -->
    <param name = "MBeanFilter" value =
"JMImplementation:*;MletAgent:*"/>
    </ParameterList>
   </MBeanServer>

<MBeanServer name="WebLogic71" type="WebLogic">
<JMXClassPath>C:/bea/weblogic700/server/lib/weblogic.jar</JMXClass
Path>
    <ParameterList>
    <!--  WebLogic Server URL  -->
    <param name = "JNDIUrl" value = "t3://localhost:7001"/>
    <!-- User Name -->
    <param name = "User" value = "admin"/>
    <!-- Password, Use jmxpassword.bat utility to ecrypt the
password for Windows platform and tibjmxpassword script for UNIX
platform -->
    <param name = "Password" value = "EncryptMe"/>
    <!--  MBean Filter Value  -->
    <param name = "MBeanFilter" value =
"JMImplementation:type=MBeanServerDelegate;mydomain:Name=myserver,
Server=myserver,Type=JTARecoveryService"/>
    </ParameterList>
   </MBeanServer>

   <MBeanServer name="WebSphere1" type="WebSphere">
   <JMXClassPath>C:/IBM/WebSphere/AppServer/lib;C:/IBM/WebSphere/A
ppServer/java/jre/lib;C:/IBM/WebSphere/AppServer/java/jre/lib/ext<
/JMXClassPath>
    <ParameterList>
    <param name = "ConnectorType" value = "SOAP"/>
    <param name = "ConnectorHost" value = "localhost"/>
    <param name = "ConnectorPort" value = "8880"/>
    <param name = "MBeanFilter" value = "JMImplementation:*"/>
    </ParameterList>
   </MBeanServer>
```

```
  </MBeanServerList>
</JMXConfiguration>
```

Appendix B Error Messages

This appendix list the error messages used by TIBCO Hawk JMX Plug-in.

Topics

TIBCO Hawk JMX Plug-in Error Messages

Please check argument for Server type for MBeanServerElement %1 %2 is not a Valid Server Type. WebLogic, JBoss3, WebSphere and JSR160 is a valid MBeanServer type. Ignoring this MBean Server

Role: Error

Resolution: Please specify a valid MBean server type in the configuration file and re-start TIBCO Hawk Agent.

'AddMBeanServerName' should be set to 'true' to avoid naming conflict.

Role: Warning

Resolution: Please make sure the specified file exists.

File: %1 does not exists. Please check location.

Role: Error

Resolution: Please make sure the specified file exists.

Exception caught in IntializeMicroAgent : %1 %2

Role: Error

Resolution: Please contact TIBCO Support.

Exception in setMicroAgentContainer: %1

Role: Error

Resolution: Please contact TIBCO Support.

Error setting AsyncDataHandler for Worker Thread for start of subscription.

Role: Error

Resolution: Please contact TIBCO Support.

Error setting AsyncDataHandler for Worker Thread for stop of subscription.

Role: Error

Resolution: Please contact TIBCO Support.

Attribute not found Exception: %1

Role: Error

Resolution: Make sure the MBean to which this attribute belongs to exists. If the problem persists, contact TIBCO Support.

Invalid Attribute Value Exception: %1

Role: Error

Resolution: Please contact TIBCO Support.

Instance not found Exception: %1

Role: Error

Resolution: Make sure the MBean to which this attribute belongs to exists. If the problem persists, contact TIBCO Support.

Reflection Exception: %1

Role: Error

Resolution: The signature of the MBean may have changed. If this is the case, re-start TIBCO Hawk Agent.

Runtime Error Exception: %1

Role: Error

Resolution: Please contact TIBCO Support.

Remote RMI Exception : %1

Role: Error

Resolution: Check the connection to the application server.

IO Exception : %1

Role: Error

Resolution: Check the connection of the plug-in with the MBean sever. If the problem persists, please contact TIBCO Support.

Error removing MicroAgents from Agent attached on Thread:%1

Role: Error

Resolution: Re-start TIBCO Hawk Agent. If the problem persists, contact TIBCO Support.

Exception caught in Filtering MBean: %1

Role: Error

Resolution: Check the syntax of the MBean filter specified in the configuration file and re-start TIBCO Hawk Agent.

Security Exception Decrypting password %1 Please check the password entered and restart the Hawk Agent.

Role: Error

Resolution: Please check the password and re-start TIBCO Hawk Agent.

Ignoring MBean: %1 because it has No Attribute and No Operation.

Role: Info

Ignoring MBean %1 due to the following error: %2.

Role: Error

Resolution: Depends on the content of the exception.

Waiting for 2 Seconds before contacting %1 connection server at: %2

Role: Info

TIBCO Software Inc. End User License Agreement

READ THIS END USER LICENSE AGREEMENT CAREFULLY. BY DOWNLOADING OR INSTALLING THE SOFTWARE, YOU AGREE TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT AGREE TO THESE TERMS, DO NOT DOWNLOAD OR INSTALL THE SOFTWARE AND RETURN IT TO THE VENDOR FROM WHICH IT WAS PURCHASED.

Upon your acceptance as indicated above, the following shall govern your use of the Software except to the extent all or any portion of the Software (a) is subject to a separate written agreement, or (b) is provided by a third party under the terms set forth in an Addenda at the end of this Agreement, in which case the terms of such addenda shall control over inconsistent terms with regard to such portion(s).

License Grant. The Software is the property of TIBCO or its licensors and is protected by copyright and other laws. While TIBCO continues to own the Software, TIBCO hereby grants to Customer a limited, non-transferable, non-exclusive, license to use the number of Permitted Instances set forth in the Ordering Document, in machine-readable, object code form and solely for Customer's internal business use.

Restrictions. Customer agrees not to (a) make more copies than the number of Permitted Instances plus a reasonable number of backups; (b) provide access to the Software to anyone other than employees, contractors, or consultants of Customer; (c) sublicense, transfer, assign, distribute to any third party, pledge, lease, rent, or commercially share the Software or any of Customer's rights under this Agreement (for the purposes of the foregoing a change in control of Licensee is deemed to be an assignment); (d) use the Software for purposes of providing a service bureau, including, without limitation, providing third-party hosting, or third-party application integration or application service provider-type services, or any similar services; (e) use the Software in connection with ultrahazardous activities, or any activity for which failure of the Software might result in death or serious bodily injury to Customer or a third party; or (f) directly or indirectly, in whole or in part, modify, translate, reverse engineer, decrypt, decompile, disassemble, make error corrections to, create derivative works based on, or otherwise attempt to discover the source code or underlying ideas or algorithms of the Software.

Beta and Evaluation Licenses. Notwithstanding the foregoing, if the Software is being provided for demonstration, beta testing, or evaluation purposes, then Customer agrees (a) to use the Software solely for such purposes, (b) that the Software will not be used or deployed in a production environment, and (c) that such use shall automatically terminate upon the earlier of thirty days from the date Customer receives the right to install the Software, or Customer's receipt of notice of termination from TIBCO.

Technical Support. Provided Customer has paid applicable support fees (not included with Software fees unless separately listed), TIBCO shall provide support for generally available TIBCO Software on an annual basis commencing on the Purchase Date, as follows ("Support"): Customer shall designate at TIBCO's support website https://support.tibco.com/eSupport/newuser.html, the number of technical support contacts permitted under the level of Support purchased (contacts are changeable upon 48-hours prior written notice to TIBCO). Each contact may contact TIBCO for problem resolution during TIBCO's published support hours corresponding to the level of Support fees paid.

Upon notice from a contact of a Software problem which can be reproduced at a TIBCO support facility or via remote access to Customer's facility, TIBCO shall use reasonable efforts to correct or circumvent the problem according to its published support objectives. TIBCO reserves the right to make changes only to the most currently available version. TIBCO will use reasonable efforts to support the previously released version of the Software for a maximum of six months.

TIBCO shall have no obligation to support the Software (i) for use on any computer system running other than the operating system software for which the Software is approved (as set forth in the Software documentation) and licensed hereunder, or (ii) if Customer has modified or authorized a third party to modify the Software. TIBCO shall have no obligation to modify any version of the Software to run with any new versions of any operating system, or any other third party software or hardware. If Customer purchases Support for any Software, Customer must purchase the same level of Support for all copies of the Software for which it is licensed.

Support may be extended for one-year periods on the anniversary of each Purchase Date at the standard amounts set forth in its price list, for as long as TIBCO offers Support. Customer may reinstate lapsed support for any then currently supported Software by paying all Support fees in arrears and any applicable reinstatement fee. Upgrades, patches, enhancements, bug fixes, new versions and/or new releases of the Software provided from time to time under Support shall be used only as replacements to existing Permitted Instances, and shall not be deemed to increase that number, and use thereof shall be governed by the terms of this Agreement, except for the first paragraph of the Limited Warranty and any right of return or refund.

Consulting Services. Customer may request additional services ("Services") either in an Ordering Document, or by a separate mutually executed work order, statement of work or other work-request document incorporating this Agreement (each, a "Work Order"). Unless otherwise expressly agreed to in a Work Order, all Services and any work product therefrom shall be (a) performed on a time and materials basis, plus meals, lodging, travel, and other expenses reasonably incurred in connection therewith, (b) deemed accepted upon delivery, and (c) exclusively owned by TIBCO (except for confidential information of Customer identified to TIBCO in the Ordering Document), including all right, title and intellectual property or other right or interest therein. Each Work Order is intended to constitute an independent and distinct agreement of the parties, notwithstanding that each shall be construed to incorporate all applicable provisions of this Agreement. Specific to TIBCO training services, additional information regarding courses, registration, restrictions or limitation can be found at TIBCO's website at http://www.tibco.com/services/education under Education Programs. Fees for Services shall be due and payable in United States dollars net 30 from the date of TIBCO's invoice.

Limited Warranty. If Customer obtained the Software directly from TIBCO, then TIBCO warrants that for a period of thirty (30) days from the Purchase Date: (i) the media on which the Software is furnished will be free of defects in materials and workmanship under normal use; and (ii) the Software will substantially conform to its published specifications. This limited warranty extends only to the original Customer hereunder. Customer's sole and exclusive remedy and the entire liability of TIBCO and its licensors under this limited warranty will be, at TIBCO's option, repair, replacement, or refund of the Software and applicable Support fees, in which event this Agreement shall terminate upon payment thereof.

This warranty does not apply to any Software which (a) is licensed for beta, evaluation, testing or demonstration purposes for which TIBCO does not receive a license fee, (b) has been altered or modified, except by TIBCO, (c) has not been installed, operated, repaired, or maintained in accordance with instructions supplied by TIBCO, (d) has been subjected to abnormal physical or electrical stress, misuse, negligence, or accident, or (e) is used in violation of any other term of this Agreement. Customer agrees to pay TIBCO for any Support or Services provided by TIBCO related to a breach of the foregoing on a time, materials, travel, lodging and other reasonable expenses basis. If Customer obtained the Software from a TIBCO reseller or distributor, the terms of any warranty shall be as provided by such reseller or distributor, and TIBCO provides Customer no warranty with respect to such Software.

EXCEPT AS SPECIFIED IN THIS LIMITED WARRANTY, THE SOFTWARE, SUPPORT AND SERVICES ARE PROVIDED "AS IS", ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS, AND WARRANTIES INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OR CONDITION OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, SATISFACTORY QUALITY OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE, ARE HEREBY EXCLUDED TO THE EXTENT ALLOWED BY APPLICABLE LAW. NO WARRANTY IS MADE REGARDING THE RESULTS OF ANY SOFTWARE, SUPPORT OR SERVICES OR THAT THE SOFTWARE WILL OPERATE WITHOUT ERRORS, PROBLEMS OR INTERRUPTIONS, OR THAT ERRORS OR BUGS IN THE SOFTWARE WILL BE CORRECTED, OR THAT THE SOFTWARE'S FUNCTIONALITY OR SERVICES WILL MEET CUSTOMER'S REQUIREMENTS. NO TIBCO DEALER, DISTRIBUTOR, AGENT OR EMPLOYEE IS AUTHORIZED TO MAKE ANY MODIFICATIONS, EXTENSIONS OR ADDITIONS TO THIS WARRANTY.

Indemnity. If Customer obtained the Software from TIBCO directly, then TIBCO shall indemnify Licensee from and against any final judgment by a court of competent jurisdiction, including reasonable attorneys' fees, that the unmodified TIBCO Software infringes any patent issued by the United States, Canada, Australia, Japan, or any member of the European Union, or any copyright, or any trade secret of a third party; provided that TIBCO is promptly notified in writing of such claim, TIBCO has the exclusive right to control such defense and/or settlement, and Licensee shall provide reasonable assistance (at TIBCO's expense) in the defense thereof. In no event shall Licensee settle any claim, action or proceeding without TIBCO's prior written approval. In the event of any such claim, litigation or threat thereof, TIBCO, at its sole option and expense, shall (a) procure for Licensee the right to continue to use the TIBCO Software or (b) replace or modify the TIBCO Software with functionally equivalent software. If such settlement or modification is not commercially reasonable (in the reasonable opinion of TIBCO), TIBCO may cancel this Agreement upon sixty days prior written notice to Licensee, and refund to Licensee the unamortized portion of the license fees paid to TIBCO by Licensee based on a five-year straight-line depreciation. This Section states the entire liability of TIBCO with respect to the infringement of any Intellectual Property rights, and Licensee hereby expressly waives any other liabilities or obligations of TIBCO with respect thereto. The foregoing indemnity shall not apply to the extent any infringement could have been avoided by use of the then-current release.

Limitation of Liability. EXCEPT AS PROVIDED UNDER INDEMNITY OR RESULTING FROM A BREACH OF CONFIDENTIALITY (THE "EXCLUDED MATTERS"), IN NO EVENT WILL EITHER PARTY OR TIBCO'S LICENSORS BE LIABLE FOR ANY LOST DATA, LOST REVENUE, LOST PROFITS, DAMAGE TO REPUTATION, BUSINESS INTERRUPTION, OR ANY OTHER INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL, PUNITIVE, EXEMPLARY OR ANY SIMILAR TYPE DAMAGES ARISING OUT OF THIS AGREEMENT, THE USE OR THE INABILITY TO USE THE SOFTWARE, OR THE PROVISION OF ANY SUPPORT OR SERVICES, EVEN IF A PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. EXCEPT FOR THE EXCLUDED MATTERS, IN NO EVENT SHALL A PARTY BE LIABLE TO THE OTHER, WHETHER IN CONTRACT, TORT (INCLUDING ACTIVE OR PASSIVE NEGLIGENCE), BREACH OF WARRANTY, CLAIMS BY THIRD PARTIES OR OTHERWISE, EXCEED THE PRICE PAID BY CUSTOMER UNDER THE APPLICABLE ORDERING DOCUMENT.

THE FOREGOING LIMITATIONS SHALL APPLY EVEN IF THE ABOVE-STATED REMEDY OR LIMITED WARRANTY FAILS OF ITS ESSENTIAL PURPOSE. BECAUSE SOME STATES OR JURISDICTIONS DO NOT ALLOW LIMITATION OR EXCLUSION OF CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO CUSTOMER.

Confidentiality. "Confidential Information" means the terms of this Agreement; all information marked by the disclosing party as proprietary or confidential; any provided software, related documentation or related performance test results derived by Licensee; and any methods, concepts or processes utilized in provided software or related documentation. Confidential Information shall remain the sole property of the disclosing party and shall not be disclosed to any non-Authorized User without the prior written consent of the disclosing party. If Confidential Information is communicated orally, such communication shall be confirmed as "Confidential" in writing within thirty days of such disclosure. The parties agree to protect the Confidential Information of the other in the same manner it protects the confidentiality of similar information and data of its own (and at all times exercising at least a reasonable degree of care). Except with respect to the Software, items will not be deemed Confidential Information if (i) available to the public other than by a breach of an agreement with TIBCO, (ii) rightfully received from a third party not in breach of any obligation of confidentiality, (iii) independently developed by one party without use of the Confidential Information of the other; (iv) known to the recipient at the time of disclosure (other than under a separate confidentiality obligation); or (v) produced in compliance with applicable law or court order, provided the other party is given reasonable notice of the same. Both parties agree to indemnify the other for any damages the other may sustain resulting from their unauthorized use and/or disclosure of the other's Confidential Information. Such damages shall include reasonable expenses incurred in seeking both legal and equitable remedies. To the extent required by law, at Customer's request, TIBCO shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of TIBCO's applicable fee. Customer agrees to observe obligations of confidentiality with respect to such information.

Export. Software, including technical data, is subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Customer agrees to comply strictly with all such regulations and agrees to obtain all necessary licenses to export, re-export, or import Software.

Government Use. If the Customer is an agency, department, or other entity of the United States Government ("Government"), the use, duplication, reproduction, release, modification, disclosure or transfer of the Software, or any related documentation of any kind, including technical data or manuals, is restricted in accordance with Federal Acquisition Regulation ("FAR") 12.212 for civilian agencies and

Defense Federal Acquisition Regulation Supplement ("DFARS") 227.7202 for military agencies. The Software is commercial computer software and commercial computer software documentation. Use of the Software and related documentation by the Government is further restricted in accordance with the terms of this Agreement, and any modification thereto.

Orders. An Ordering Document shall be deemed accepted only by issuance of a TIBCO invoice and solely for purposes of administrative convenience. None of the terms of the Ordering Document (other than the Software product name, number of Permitted Instances, level of Support, description of Services, and fees due in connection therewith) shall apply for any reason or purpose whatsoever, regardless of any statement on any Ordering Document to the contrary, unless countersigned by an officer of TIBCO. This Agreement constitutes the entire agreement between the parties with respect to the use of the Software, Support and Services, and supersedes all proposals, oral or written, and all other representations, statements, negotiations and undertakings relating to the subject matter hereof. All orders of Software, Support or Services by Customer from TIBCO shall be deemed to occur under the terms of this Agreement (with or without reference to this Agreement), unless expressly superseded by a signed written Agreement between the parties. Software shall be delivered electronically, and such delivery shall occur when the TIBCO Software is made available for download by Customer. Physical deliveries (as applicable) of Software and documentation which typically accompanies the Software on delivery shall be on CD-ROM, FOB Palo Alto, and delivery shall occur by depositing the CD-ROM with TIBCO's overnight carrier (at no charge to Customer).

Term and Termination. Support or Services may be terminated: (a) by either party upon a default of the other, such default remaining uncured for fifteen days from written notice from the non-defaulting party; (b) upon the filing for bankruptcy or insolvency of the other party, (c) by either party upon prior written notice at least sixty days prior to the end of any annual Maintenance period; or (d) by Licensee (for Services), upon ten days prior written notice. Termination of Support or Services shall not terminate this Agreement. Customer may terminate this Agreement in its entirety at any time by destroying all copies of the Software. Upon termination of this Agreement in its entirety, for any reason, Customer must cease using and return or destroy all copies of the Software. Customer's obligation to pay accrued charges and any fees due as of the date of termination, as well as the sections entitled "Confidentiality", "Limited Warranty" and "Limitation of Liability" shall survive any such termination.

Authority. You hereby represent and warrant that you have full power and authority to accept the terms of this Agreement on behalf of Customer, and that Customer agrees to be bound by this Agreement.

General. Fees on the Ordering Document (all to be paid on the latter of thirty days from Invoice by TIBCO or the date set forth in the Ordering Document) do not include sales, use, withholding, value-added or similar taxes, and Customer agrees to pay the same, excluding therefrom taxes related to TIBCO's income and corporate franchise tax. Customer agree to pay all reasonable costs incurred (including reasonable attorneys' fees) in collecting past due amounts under this Agreement. Except as set forth in the Section entitled Limited "Warranty" all fees paid under or in connection with this Agreement are non-refundable and no right of set-off exists. All payments of fees due shall be made in U.S. dollars, net 30 from Purchase Date, or, for any other amounts coming due hereafter, net 30 from TIBCO's invoice. A service charge of one and one-half percent per month will be applied to all invoices that are not paid on time. Licensee agrees to pay all sales, use, value-added, withholding, excise and any other similar taxes or government charges, exclusive of TIBCO's income taxes. No delay in the performance of any obligation by either party, excepting all obligations to make payment, shall constitute a breach of this Agreement to the extent caused by force majeure. Customer hereby grants TIBCO and its independent auditors the right to audit Customer's compliance with this Agreement. If any portion of this Agreement is found to be void or unenforceable, the remaining provisions shall remain in full force and effect. This Agreement shall be governed by and construed in accordance with the laws of the State of California, United States of America, as if performed wholly within the state and without giving effect to the principles of conflict of law. The state and/or federal courts in San Francisco, California, shall have exclusive jurisdiction of any action arising out of or relating to this Agreement. The United Nations Convention on Contracts for the International Sale of Goods is excluded from application hereto. If any portion hereof is found to be void or unenforceable, the remaining provisions of this Agreement shall remain in full force and effect.

Definitions. In connection with this Agreement, the following capitalized terms shall have the following meaning: "Agreement" means this End User License Agreement; "Case Start" means the initiation of a single instance of a defined business process; "Connection" for the following TIBCO Software products shall mean: for TIBCO Enterprise Message Service, a TIBCO Enterprise Message Service client connection to the TIBCO Enterprise Message Service server for the purpose of sending or receiving messages, for TIBCO SmartSockets and TIBCO SmartMQ, any network protocol link established with such TIBCO Software (directly or indirectly) to any other entity, including but not limited to software, firmware or hardware, for TIBCO Enterprise RTView - Standard Monitor System, the number of monitored server instances to TIBCO Rendezvous daemons or TIBCO Hawk agents; for TIBCO Enterprise RTView- EMS Monitor System, a monitored TIBCO Enterprise Message Service Connection (as defined above for that product); for TIBCO General Interface, an electronic data interface to a CPU on a server (which excludes CPUs on devices such as routers, switches, proxies, or HTTP or application servers configured to substantially pass-through information or messages to TIBCO General Interface) that produces information or messages consumed by TIBCO General Interface; "Customer" means the original purchaser or licensee of the Software and any permitted successors and assigns; "Developer" means one user/developer of a TIBCO Software product for use in Development; "Development" means used for software development purposes only; "Enterprise" means an unlimited number of Permitted Instances for a period of one year from the Purchase Date (unless otherwise set forth in the Ordering Document), at which time existing licenses convert to perpetual and Customer may not thereafter deploy additional Permitted Instances, and in any event, shall (during the one-year unlimited deployment period) exclude any entity which acquires, is acquired by, merged into, or otherwise combined with Customer. Customer hereby agrees to provide TIBCO with notice of the number of Permitted Instances deployed at the end of such one-year period within thirty days thereafter; "Fab" means unlimited use for shop-floor manufacturing applications at a Site; "Workstation" shall mean a single end-user computer that is generally intended to be accessed by one person at a time; "Ordering Document" means any purchase order or similar document or agreement requesting Software, Support or Services; "Permitted Instance(s)" means the number of copies of Software running on a Server Instance, Workstation, User, or Development basis, on a designated Platform, as set forth in an Ordering Document, including, without limitation, Enterprise, Site and Fab licensing; "Platform" means the operating system set forth in an Ordering Document; "Purchase Date" means the date of the Ordering Document is accepted by TIBCO; "Server Instance" means a computer with 1 CPU (unless otherwise set forth in the Ordering Document) performing common services for multiple machines; "Site" means an unlimited number of Permitted Instances at a specific

physical address set forth in the Ordering Document (or, in the absence of any address, at Customer's corporate headquarters); "Software" means the software products listed in an Ordering Document (except as provided in the second paragraph hereof), in whole and in part, along with their associated documentation; "TIBCO" means TIBCO Software Inc.; and "Named User" means the number of named users with access to the Software.

Special Product Provisions. TIBCO BusinessPartner: Customer may sublicense to third parties ("Partners") up to the total Number of Copies of TIBCO BusinessPartner, provided that for every such sublicense, the Number of Copies Customer is licensed to use shall be reduced by the same number, and provided further that prior to delivery of TIBCO BusinessPartner to a Partner, such Partner agrees in writing (a) to be bound by terms and conditions at least as protective of TIBCO as the terms of this Agreement, (b) that TIBCO BusinessPartner be used solely to communicate with Customer's implementation of TIBCO BusinessConnect, and (c) for such Partner to direct all technical support and Maintenance questions directly to Customer. Customer agrees to keep records of the Partners to which it distributes TIBCO BusinessPartner, and to provide TIBCO the names thereof (with an address and contact name) within sixty days of the end of each quarter. Third Party Software: Use of any other third-party software identified by its company and/or product name or otherwise designated in Licensee's Ordering Document (collectively "Third Party Software") is subject solely to the terms and conditions of the click-wrap or shrink-wrap license agreement included with the Third Party Software products, and for which TIBCO shall be an intended third-party beneficiary of same. TIBCO shall have no obligation whatsoever in connection with the Third Party Software (including, without limitation, any obligation to provide maintenance or support) and the provision of Third Party Software is accomplished solely as an accommodation and in lieu of Customer purchasing a license to Third Party Software directly from the third party vendor. Embedded/Bundled Products. Some TIBCO Software embeds or bundles other TIBCO Software (e.g., TIBCO InConcert bundles TIBCO Rendezvous). Use of such embedded or bundled TIBCO Software is solely to enable the functionality of the TIBCO Software licensed on the Cover Page, and may not be used or accessed by any other TIBCO Software, or for any other purpose. Open Source Software: If Licensee uses Open Source software in conjunction with the TIBCO Software, Licensee must ensure that its use does not: (i) create, or purport to create, obligations of use with respect to the TIBCO Software; or (ii) grant, or purport to grant, to any third party any rights to or immunities under TIBCO's intellectual property or proprietary rights in the TIBCO Software. You also may not combine the TIBCO Software with programs licensed under the GNU General Public License ("GPL") in any manner that could cause, or could be interpreted or asserted to cause, the TIBCO Software or any modifications thereto to become subject to the terms of the GPL.

Version 5.2, 3/05

Index

U