TIBCO BusinessEvents® Enterprise Edition

Administration Guide

Version 6.3.1 | September 2024

Contents

Administration Overview

The *TIBCO BusinessEvents Administration* guide explains how to prepare for deployment. It also explains how to deploy, monitor, and manage the runtime application.

Before you begin to use *TIBCO BusinessEvents Administration*, gain a basic familiarity with the product by completing the tutorials in *TIBCO BusinessEvents Getting Started*, and read *TIBCO BusinessEvents Architect Guide*.

Building EAR Files for Deployment

Deployment requires project Enterprise Archive (EAR) files, which are considered as an input for administrative tasks. For more information on EAR files, see Enterprise Archive (EAR) Files.

You can build EAR files as follows:

- Using TIBCO BusinessEvents Studio. See Building an EAR File in TIBCO BusinessEvents Studio.
- At the command line. See Building an EAR File at the Command Line.

Deploy-Time Configuration

System level configuration is generally needed. Edit the engine TRA file to add and set values for settings that are read before the engine starts.

- See JVM-Level TRA File Configuration

Custom Functions and Third-Party Jars at Deploy-time

With all methods of deployment, ensure that certain files are available at run time. If your project has JAR files for custom functions or third-party software, manually copy them to the runtime location. Copy them to a location on the classpath of the deployed application. The recommended location is the *BE_HOME*`/lib/ext/tpcl` directory. If you choose a location that is not in the classpath, then update the classpath in the TRA file to include the location.

At run time the software uses the classpath set in the `be-engine.tra` file to locate the libraries (third-party libraries and custom function libraries) needed to execute the code. Ensure that you have added all the classpaths needed before you deploy. For example, you must update the classpath to specify the locations of libraries for TIBCO Enterprise Message Service, TIBCO Hawk, third-party software, and custom functions.

Business Rules Deployment Directory Property

Before deploying a business rule and starting the engine, set the property `be.cluster.ruletemplateinstances.deploy.dir` in the Cluster Deployment Descriptor (CDD), `be-engine.tra`, or in a `.properties` file. The property specifies the directory from which the engine loads business rules for the specific project. During startup, the engine reads the business rules from the specified directory and loads them into all the rule sessions. Ensure that the directory is local to the machine on which the engine is running. To avoid conflicts, the deployment directory specified should not contain business rules for other projects.

Deployment

The output of a design-time project is one or more EAR files and one or more CDD files.

For details on configuring and building these files, see Enterprise Archive (EAR) Files.

An EAR file deploys as one TIBCO BusinessEvents processing unit (engine). A processing unit can either contain one cache agent, or it can contain one or more agents of other types. Processing units and agents are defined in the CDD file.

When you deploy an EAR, you specify the CDD file to use, and you specify which processing unit class to deploy.

You can deploy in these ways:

- Using TIBCO BusinessEvents Enterprise Administrator Agent. This is the recommended approach. See TIBCO BusinessEvents Enterprise Administrator Agent.
- At the command line. See Building an EAR File at the Command Line.

- To a TIBCO Administrator domain. See Deploying a Project in a TIBCO Administrator Domain.

> **Note:** For details about deploying TIBCO BusinessEvents Decision Manager classes (implemented virtual rule functions) see *TIBCO BusinessEvents Decision Manager User Guide*.

Overriding Global Variables at Deploy Time

All methods of deployment enable you to override global variables at deploy time. For design-time procedures relating to global variables see "Working with Global Variables" in *TIBCO BusinessEvents Developer Guide*.

Hot Deployment

You can configure your TIBCO BusinessEvents engine to allow you to replace the EAR file without shutting down the engine. This is known as Hot Deployment.

Management and Monitoring

Depending on your method of deployment, you can use either TIBCO BusinessEvents Enterprise Administrator Agent or TIBCO Administrator (with TIBCO Hawk) for monitoring and management:

- TIBCO BusinessEvents Enterprise Administrator Agent.
- Certain topics in Project Deployment, and TIBCO Hawk Microagent Methods.

Authentication and Authorization

Certain components use authentication (TIBCO BusinessEvents Decision Manager). Currently, only TIBCO BusinessEvents Decision Manager uses authorization (access control).

- User Authentication
- Access Control Configuration

Cluster Startup and Shutdown

There are only two main points to keep in mind for orderly system startup and shutdown:

Start storage-enabled agents (cache agents) first

When Cache OM is used, you must start a node that has storage enabled first. In production systems that would be a dedicated cache agent engine. In test deployments, this could be another type of agent node with cache storage enabled.

Stop other engines before storage-enabled agents (cache agents)

In unusual situations where all cache agents are stopped but engines running other types of agents are running, restart all engines.

Engine Startup and Shutdown

Certain actions occur in sequence during engine startup and shutdown.

In any particular project only some of these startup or shutdown actions may be required. For example, a project might not have any startup rule functions.

Note: During startup, the TIBCO BusinessEvents engine tries to load all the business rules present in the shared folder. Any failure when loading the business rules prevents the engine from starting.

Except where noted, this section assumes Cache OM and inference agent startup and shutdown. It provides the main milestones only.

Engine Startup Sequence

The following actions comprise the engine startup sequence:

1. System information is displayed in consoles and is recorded in the log file:
    a. The property file and EAR file that were used to start the engine.
    b. The version of the JAR files it is using, and the version of the JAR files that the EAR file was built with.

2. Cache OM with backing store only: Recovery stage. When the minimum number of cache agents is started (as defined by the Cache Agent Quorum CDD setting), the cluster enters the Recovery state. Various caches are preloaded from the backing store, according to preload settings. When the Recovery state ends, the cluster enters the Ready state.
3. All inference agents build their Rete networks by evaluating conditions against all objects. For (Cache OM only) Inactive (Standby) Nodes: if all agents in an engine are inactive, then this ends the startup sequence for that engine.
4. Channels start for outbound traffic. Inbound listeners do not start yet.
5. Scorecards are created.
6. Startup functions run (for example, they initialize the values of scorecards).
7. The first RTC cycle occurs and all rule actions that are eligible to run now. (Scorecards and startup rule functions can cause rules to be eligible to run. Depending on the state of the entities recovered from the backing store, the RTC takes more or less time.) See *TIBCO BusinessEvents Architect Guide* for more details about RTC cycles.
8. The engine startup advisory event is asserted, and its RTC occurs (as needed).
9. Time events (if any) are asserted:
    a. The clock starts for repeating time events and they are created and asserted at the specified intervals.
    b. Rule-based time events (recovered or scheduled in a startup action) are asserted after the specified delay. The delay begins when the rule or rule function action runs, so, at startup, it is possible for time events to have passed their start time, and they are asserted immediately.
10. Inbound channel listeners activate and accept incoming events and the system is now fully started.

> **Note:** The `be.engine.startup.parallel` is used to start inference agents concurrently, that is in parallel. By default, 5.X inference agents start serially.

Engine Shutdown Sequence

The following actions comprise the shutdown sequence:

1. Inbound channels and listeners shut down.
2. Shutdown rule functions run.
3. An RTC occurs (as needed).
4. Outbound channels shut down.

Order of Precedence at Run time

This is the order of precedence that is established at run time, from the highest priority to the lowest:

1. Command-line arguments at engine startup.
2. Properties set in property files specified at the command line.
3. Properties in the deployed TRA file.
4. CDD file, processing unit level (for the current PU): properties and settings.
5. CDD file, agent class level (for agents listed in the current PU settings, prioritized in reverse order of that list): properties and settings.
6. CDD file, cluster level: properties, settings and message encoding.
7. EAR file properties (such as global variable overrides).

> **Note:** Global variables set in the CDD file are ignored if you deploy using TIBCO Administrator.
>
> TRA files should be used only for system-level settings that must be read before the JVM starts. All other properties should be in the CDD.

Values Used to Establish the Engine Name

When establishing the engine name, TIBCO BusinessEvents searches for a value, and accepts the first found value.

- API setting. If TIBCO BusinessEvents is started using the public API, and a non-null instance name is provided when getting the `RuleServiceProvider` with `RuleServiceProviderManager.newProvider(String instanceName, Properties env)` —this takes precedence over all other name settings.
- The engine name set at the command line using the -name option. An engine name set at the command line overrides the engine name property set in the CDD file or `be-engine.tra` or supplementary property file.
- The engine name set by the `be.engine.name` property in the TRA file. For command-line startup it can be set in a supplementary property file.
- The engine name set in the CDD file, in the **Name** field of the **Processing Unit** tab. See Agent and Processing Unit Configuration in *TIBCO BusinessEvents Developer Guide*.
- The name of the TIBCO Hawk microagent instance. This name exists if TIBCO Hawk is enabled at run time. The microagent name can also be set in the `be-engine.tra` file using the property `Hawk.AMI.DisplayName`.
- The host name.
- This string: `engine`.

Primary Key Strategies

To get improved performance of entity lookups in cache and stores, and to make the lookup behavior more compatible with modern stores (databases), you can choose to use the key-based lookup strategy.

Key-Based Lookup Strategy

The key-based lookup strategy is enabled by default. You can set a concept or event property as primary key using the **Present in Key** metadata setting in the CDD in `DomainObject Overrides`. You can also select more than one entity properties as **Present in Key** to have a composite key. You can use the `Cluster.DataGrid.CacheLoadConceptByKeysByUri` or `Instance.getByKeysByUri` catalog functions to load the concept.

If you do not mark an entity property as present-in-key then the `extId` becomes primary key for that entity. If `extId` is not provided for an entity instance, an internal `extId` is automatically generated for that instance.

You can also use the key-based lookup strategy and load a concept using external key or composite key then use the following settings:

- The `Cluster.DataGrid.CacheLoadConceptByKeysByUri()` and `Instance.getByKeysByUri` catalog function to load concepts using composite keys.
- The **Present in Key** column under Properties Metadata section of the entities override (**Cluster > Object Management > Domain Objects > Overrides**) setting in the CDD editor.

When you use the key-based lookup strategy, the `Cluster.DataGrid.`*`*ByExtId`* catalog functions are not available for use. You can use `Cluster.DataGrid.`*`*ByUri`* functions instead. Also, the `Instance.getByIdByUri` catalog function is not available. The object table is not used for store or cache lookups. Thus, in the CDD editor, the **Preload Handles** setting (**Cluster > Object Management > Domain Objects > Overrides**) is also not available.

Legacy Lookup Strategy

The legacy (object table based) lookup strategy is disabled by default. The legacy lookup strategy is the entity lookup strategy where an entity instance can be fetched or loaded

into working memory by specifying its Long ID or extId with or without specifying its URI.

For more details about object tables, see The Role of the Object Table topic in *TIBCO BusinessEvents Architects Guide*.

Difference Between Legacy and Key-Based Lookup Strategies

Difference Between Legacy And Key-Based Lookups

Legacy Lookup Strategy	Key-Based Lookup Strategy
Object Table is required to maintain index for cache and stores.	No object table is required.
The auto generated Long Id is always the primary key.	Concept or event properties can be made the primary key.
Composite keys are not available for data lookups.	You can use composite keys as primary key for data lookups.
Less compatible with modern stores.	More compatible with modern stores.
Use of `Id` is restricted as the primary key.	No such restriction. Default is `extId`, but you can define any property as the primary key.

Migration of TIBCO BusinessEvents 5.x Projects

With the TIBCO BusinessEvents 6.2.2 release, the new key-based lookup behavior is enabled by default. You can optionally continue to use legacy ID mode by enabling the Legacy Lookup Strategy in your project.

For a detailed process, see Enabling the Legacy Lookup Strategy topic in *TIBCO BusinessEvents Migration Guide*.

For details about strategies and processes for migrating projects, see *TIBCO BusinessEvents Migration Guide*.

JVM-Level TRA File Configuration

The engine executable files each have an associated configuration file with the extension `.tra`. These files are updated only for JVM-level property settings.

As needed, configure the TRA file for JVM-level settings that must be set before the TIBCO BusinessEvents engine starts. Other settings go in the CDD file. JVM-level settings in the CDD file are ignored. For non-TIBCO BusinessEvents related JVM settings, see Java documentation as needed. The TRA file also contains some helpful comments for such properties.

At run time, the software uses the classpath set in the `be-engine.tra` file to locate the libraries (third-party libraries and custom function libraries) needed to execute the code. Ensure that you have added all the classpaths needed before you deploy. For example, update the classpath to specify the locations of libraries for TIBCO Enterprise Message Service, third-party software, and custom functions.

In some cases you must also copy the JAR files. If a JAR has dependencies on native libraries, edit *BE_HOME*`/bin/be-engine.tra` and as needed, update `LD_LIBRARY_PATH`, `SHLIB_PATH`, and `LIBPATH` as needed, depending on the operating system. For the design-time equivalent of these tasks, see "Adding and Working with Launch (Debug or Run) Configurations" and "Enabling the Test Connection Feature" in *TIBCO BusinessEvents Developer Guide*.

For additional information about system configuration, see the following:

- You must provide the *TEA_HOME* location in the *BE_HOME*`/teagent/bin/be-teagent.tra file`.
- For TIBCO Enterprise Message Service:

  If the software is installed locally, set the *EMS_HOME* variable or *RV_HOME* variable in the *BE_HOME*`/bin/be-engine.tra` files. The classpath already contains entries for these variables.

Note: For JMS channels that use TIBCO Enterprise Message Service version 5, installed locally, you must change the existing setting in the `be-engine.tra` property `tibco.env.STD_EXT_CP`: Change `%EMS_HOME%/clients/java` to `%EMS_HOME%/lib`.

- If TIBCO Enterprise Message Service is not installed locally, copy the `jms-2.0.jar` and `tibjms.jar` files to *`BE_HOME`*`/lib/ext/tpcl`. This location is specified in the standard classpath in the `be-engine.tra` file as shipped.
- For WebSphere MQ Channels, copy the copy the relevant JAR files and the binding file to the directory *`BE_HOME`*`/lib/ext/tpcl`.

  This location is specified in the standard classpath in the `be-engine.tra` file as shipped.
- For instructions on configuring the system to work with ActiveMatrix BusinessWorks see ActiveMatrix BusinessWorks Integration in *TIBCO BusinessEvents Developer Guide*.

Setting JMX Properties

JMX properties are set for various purposes.

Procedure

1. Set JMX properties for other purposes.

   Using a JMX-compliant monitoring tool such as JConsole can be useful for other purposes.
2. To enable a JMX-compliant monitoring tool to view the exposed MBeans, set these properties in the *`BE_HOME`*`/bin/be-engine.tra` files:

   `java.property.com.sun.management.jmxremote=true`

   ```
   java.property.com.sun.management.jmxremote.ssl=false
   java.property.com.sun.management.jmxremote.port=5558
   ```
3. You can also set the JMX connector port for deployment with TIBCO Administrator using this CDD property:

   `be.engine.jmx.connector.port`

4. For LDAP authentication, set the below JMX properties:

```
<property name="be.engine.jmx.connector.authenticate"
value="true"/>
<property name="be.mm.auth.type" value="ldap"/>
<property name="be.auth.ldap.type" value="openldap"/>
<property name="be.mm.auth.ldap.host" value="localhost"/>
<property name="be.mm.auth.ldap.port" value="11389"/>
<property name="be.mm.auth.ldap.adminDN"
value="uid=admin,ou=system"/>
<property name="be.mm.auth.ldap.adminPassword" value="secret"/>
<property name="be.mm.auth.ldap.baseDN" value="ou=system"/>
<property name="be.mm.auth.ldap.uidattr" value="uid"/>
<property name="be.mm.auth.ldap.roleAttr" value="member"/>
<property name="be.mm.auth.ldap.objectClass" value="*"/>
<property name="be.mm.auth.ldap.useRoleDN" value="false"/>
<property name="be.mm.auth.ldap.ssl" value="<true>"/>
<property name="be.engine.jmx.connector.ssl" value="true"/>
```

Note: `be.mm.admin.role` and `be.mm.user.role` properties should be set only if the group is neither `MM_ADMINISTRATOR` nor `MM_USER`.

Sample JConsole command:

```
jconsole -J-Djavax.net.ssl.trustStore=<locationTrustStoreFile_jks>
-J-Djavax.net.ssl.trustStoreType=JKS -J-
Djavax.net.ssl.trustStorePassword=<locationTrustStoreFile_jks>
```

For more information about the LDAP authentication properties, see LDAP Authentication Properties for the CDD File.

TIBCO BusinessEvents Enterprise Administrator Agent

TIBCO® Enterprise Administrator provides a centralized administrative interface to manage and monitor multiple TIBCO products deployed in an enterprise. A product is exposed to TIBCO Enterprise Administrator with the help of an agent. TIBCO BusinessEvents is included with TIBCO BusinessEvents Enterprise Administrator Agent that can be used to administer, manage, and monitor BusinessEvents applications.

TIBCO BusinessEvents Enterprise Administrator Agent is the backend server process that provides a management and monitoring functionality for a BusinessEvents application and cluster. The agent communicates with the TIBCO Enterprise Administrator Server for UI interactions and communicates with BusinessEvents instances using JMX. The BusinessEvents Enterprise Administrator Agent communicates with the remote machines for deployments using Secure Shell (SSH).

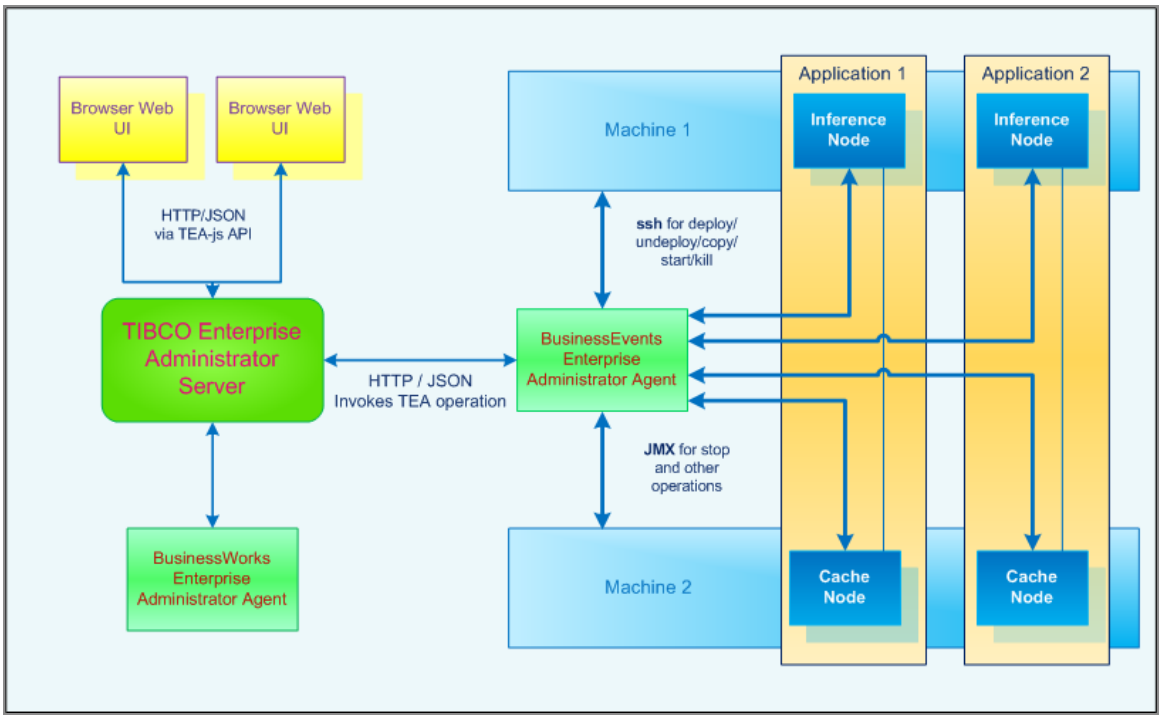
Using BusinessEvents Enterprise Administrator Agent, you can monitor and manage the BusinessEvents application deployment using a single dashboard. Thus BusinessEvents Enterprise Administrator Agent provides you with a better user experience for BusinessEvents application management.

The integration consists of two main components:

- TIBCO Enterprise Administrator server
- TIBCO BusinessEvents Enterprise Administrator Agent

TIBCO Enterprise Administrator server renders the UI and works with the user sessions and product-specific agents. The agents provide product-specific functionality, such as deployment, monitoring, and management of the product. TIBCO Enterprise Administrator server also provides a mechanism to bind users to the agent-specific roles and permissions. TIBCO Enterprise Administrator server can host or connect with multiple product agents, providing a common dashboard across products to end users. The following diagram shows the architecture for BusinessEvents integration with TIBCO Enterprise Administrator.

Figure 1: TIBCO BusinessEvents and TIBCO Enterprise Administrator Integration Architecture



Starting BusinessEvents Enterprise Administrator Agent

Start the TIBCO BusinessEvents Enterprise Administrator Agent to use the monitoring and management capabilities of TIBCO Enterprise Administrator.
In the Windows platform, you can also install the TIBCO BusinessEvents Enterprise Administrator Agent as a service, see Installing TIBCO BusinessEvents Enterprise Administrator Agent as Windows Service.

Before you begin

- In the Windows platform, ensure that Cygwin is installed and configured for using SSH. For details about Cygwin, see https://cygwin.com.
- Ensure that TIBCO Enterprise Administrator server is running. See *TIBCO Enterprise Administrator User Guide* for information on starting the TIBCO Administrator server.

- You must provide the *TEA_HOME* location in the *BE_HOME*`/teagent/bin/be-teagent.tra file`.

Procedure

Based on your interface, you can start TIBCO BusinessEvents Enterprise Administrator Agent by performing either of the following steps:

Interface	Steps to Start TIBCO BusinessEvents Enterprise Administrator Agent
Command-line	a. Navigate to *BE_HOME*`\teagent\bin`. b. Run `be-teagent.exe` to start the BusinessEvents Enterprise Administrator Agent. **Note:** If you are running `be-teagent.exe` in the Windows command-line console, then to display native characters in the command-line console, in the `log4j.properties` file, set the `log4j.appender.stdout.encoding` property value as the encoding of the command-line console. For example, the default encoding of command-line console of Japanese Windows 2012 Server is MS 932, thus, set `log4j.appender.stdout.encoding=MS932` in <*BE_HOME*>`\teagent\config\log4j.properties`.
Windows UI	On Windows, click **Start > All Programs > TIBCO > <*TIBCO_HOME*> > TIBCO BusinessEvents <*version*> > Start Enterprise Administrator Agent**.

Command-line (if TIBCO BusinessEvents Enterprise Administrator Agent is installed as a Windows service)	If you have set the service start type to manual, run the following command to start the service:
	`net start <service_name>`
	For example:
	`C:\tibco\be\6.0\teagent\bin>net start TIBCO-BETEA-Agent`
	`The TIBCO BusinessEvents Enterprise Administrator`

Interface	Steps to Start TIBCO BusinessEvents Enterprise Administrator Agent
	`Agent service is starting.` `The TIBCO BusinessEvents Enterprise Administrator Agent service was started successfully.`

Result

In the command prompt, the TIBCO BusinessEvents Enterprise Administrator Agent started successfully message is displayed.

Installing TIBCO BusinessEvents Enterprise Administrator Agent as Windows Service

You can install TIBCO BusinessEvents Enterprise Administrator Agent as a Windows service and configure it to start automatically or manually.

Procedure

1. Open the `be-teagent.tra` for editing. The `be-teagent.tra` file is located at the `<BE_HOME>\teagent\bin` folder.
2. Provide the *TEA_HOME* location in the `be-teagent.tra` file.

3. Add the following Windows service configuration properties and save the file.

```
ntservice.name=<short name for Windows NT service>
ntservice.displayname=<display name of the service>
ntservice.starttype=<type of start - automatic or manual>
ntservice.binary.path.absolute=<absolute path of be-teagent.exe>
ntservice.interactive=false
```

   For example:

```
ntservice.name=TIBCO-BETEA-Agent
ntservice.displayname=TIBCO BusinessEvents Enterprise Administrator
Agent
ntservice.starttype=automatic
ntservice.binary.path.absolute=C:/tibco/be/6.0/teagent/bin/be-
teagent.exe
ntservice.interactive=false
```

4. Open the command prompt as administrator and go to the location of the `be-teagent.exe` file.

   For example,

```
cd C:\tibco\be\6.0\teagent\bin
```

5. Run the following command to install TIBCO BusinessEvents Enterprise Administrator Agent as Windows service.

```
be-teagent.exe --install
```

Result

After successful installation, the TIBCO BusinessEvents Enterprise Administrator Agent service is listed in the Window services list.

What to do next

If you have set the start type to manual, run the following command to start the service:

```
net start <service_name>
```

For example:

```
C:\tibco\be\6.0\teagent\bin>net start TIBCO-BETEA-Agent
The TIBCO BusinessEvents Enterprise Administrator Agent service is
starting.
The TIBCO BusinessEvents Enterprise Administrator Agent service was
started successfully.
```

Signing in to the TIBCO Enterprise Administrator Server

You can use the Web UI to connect to the TIBCO Enterprise Administrator server.

Before you begin

You must start the TIBCO Enterprise Administrator server before logging in to the Web UI. Open the command prompt and navigate to <*TIBCO_HOME*>. Run `<TIBCO_HOME>\tea\<version>\bin\tea.exe`.

You must also start the BusinessEvents Enterprise Administrator Agent (see Starting BusinessEvents Enterprise Administrator Agent).

Procedure

1. Open a browser and navigate to the URL `http://localhost:8777/tea/`, where localhost is the default hostname and 8777 is the default port number.

   > **Note:** The default port number and other settings can be changed by modifying the settings in the `tea.conf` file that is available under `<TIBCO_CONFIG_HOME>\tibco\cfgmgmt\tea\conf`.

2. Enter your login credentials and click **Sign In**.

   The default username is admin and the default password is admin. The default timeout for a session is 30 minutes.

TIBCO BusinessEvents Enterprise Administrator Agent Monitoring

BusinessEvents Enterprise Administrator Agent provides charts to monitor performance of the agent itself.

The performance of the BusinessEvents Enterprise Administrator Agent is aggregated over 5 minutes and an hour based on various performance parameters. The two group of available charts are:

- Five Minute Statistics
- Hourly Statistics

Under both the groups, the following charts are displayed.

Average Used Memory

The chart displays the averages of the used memory percentage over five minutes or an hour.

Average CPU

The chart displays the averages of the CPU consumption over five minutes or an hour .

Threadcount

The chart displays the number of running threads associated with the BusinessEvents Enterprise Administrator Agent.

Note: Ensure the property **be.tea.agent.jmx.port** is set to a unique value in the `be-teagent.props` file.

BusinessEvents Application Management

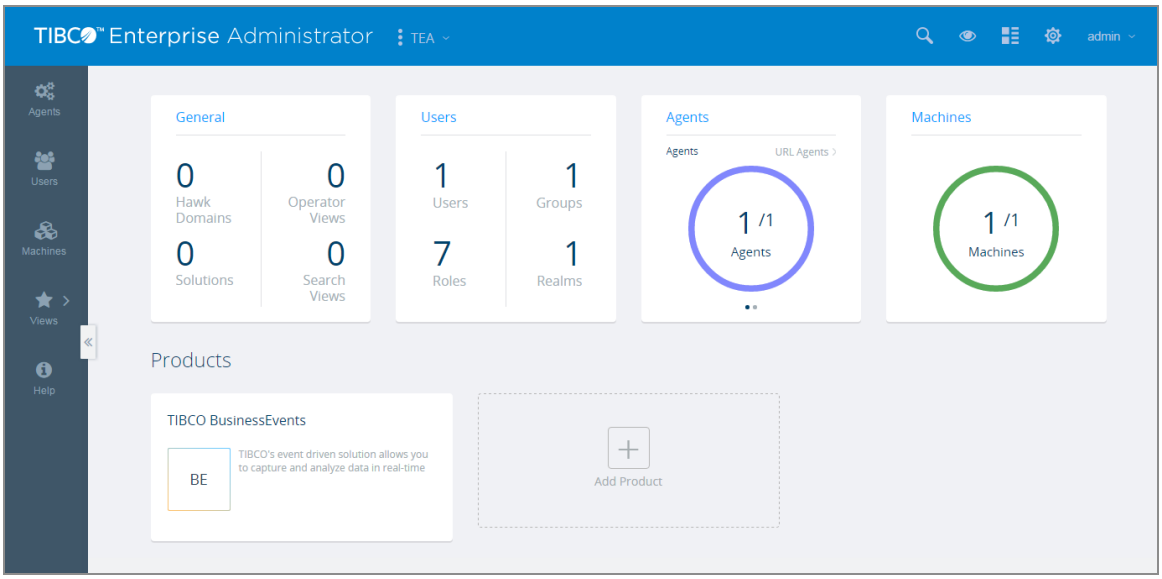
TIBCO Enterprise Administrator provides the functionality to define machines associated with the BusinessEvents application deployment. Also, you can associate instances of BusinessEvents processing units to these machines. In addition, you can also change the configuration of these processing unit instances and invoke MBeans defined by the application.

TIBCO Enterprise Administrator Landing Page

On successful authentication, the landing page is displayed. The username with which you have logged in is shown as a menu option in the title pane. The landing page displays cards with information on the general details, users, agents, machines, and products exposed to the TIBCO Enterprise Administrator server. Each of the details appearing on the card can be clicked to see more details. All the products (for example, TIBCO BusinessEvents) exposed to the server are displayed as cards. You can click on a product card to see product details.

Note: Commonly used options available on the menu are also visible on the navigation bar. To get more help on any of the features, click admin . Select **Help** and click **Go to Documentation**. This takes you to the TIBCO Enterprise Administrator Documentation.

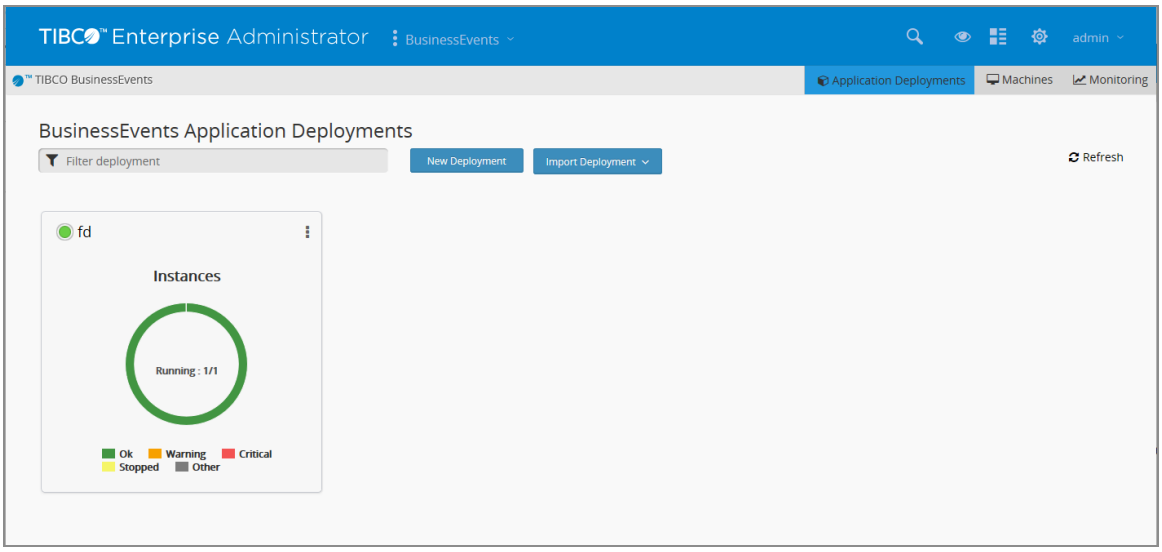
Figure 2: TIBCO Enterprise Administrator Landing Page



BusinessEvents Enterprise Administrator Agent Start Page

After clicking the TIBCO BusinessEvents product card on the TIBCO Enterprise Administrator landing page, the BusinessEvents start page is displayed. The default start page is the Application Deployments page. In the Application Deployments page you can view the deployed applications as well as perform new application deployment.

Figure 3: BusinessEvents Enterprise Administrator Agent Start Page



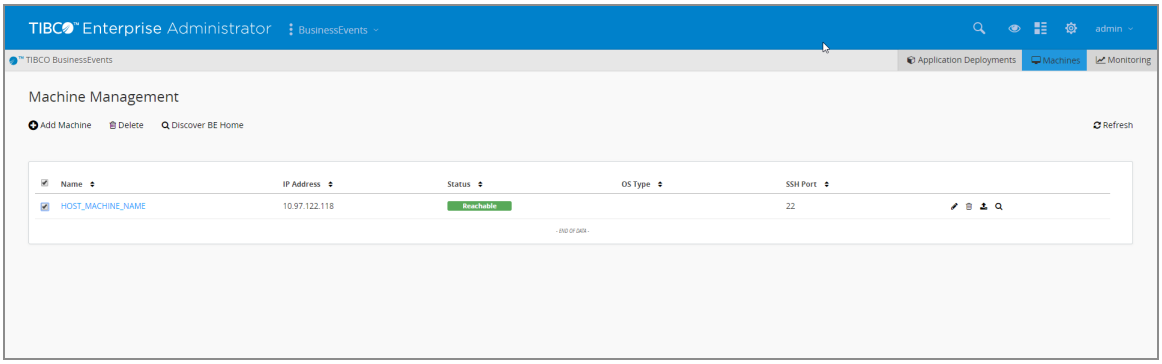
Machine Management

You can add a machine, edit a machine, or delete the machine from the machine repository of the BusinessEvents Enterprise Administrator Agent using the Machine Management page.

Machine Repository

The BusinessEvents Enterprise Administrator Agent maintains a common machine repository that can be used for multiple application deployments. The repository also contains the machines which are auto-detected by the TIBCO Enterprise Administrator server.

Figure 4: BusinessEvents Enterprise Administrator Agent Machine Management Page



In the Machine Management page, you can upload an external artifact file to a machine if

required for your project, see Uploading External Artifacts . If there are multiple TIBCO BusinessEvents installation on host machines, then BusinessEvents Enterprise Administrator Agent discovers all of them, see Auto-detecting BusinessEvents Installations on Host Machines. Each host machine have a link, associated with their name, to the page which displays instances of all application deployments on that machine and all TIBCO BusinessEvents installations on the host machine, see Viewing All Instances And BusinessEvents Installations On The Host Machine.

Viewing Machines List

You can view a list of all the machines in the BusinessEvents Enterprise Administrator Agent machine repository in the Machine Management page.

Procedure

1. Select the **Machines** tab.

   The Machine Management page is displayed.

Result

The Machine Management page lists all the machines in the machine repository. You can sort the list based on value of any of the columns. Following are the default columns for the machine list:

- Name
- IP Address
- Status
- Operating System
- SSH Port

Adding a Machine

Register a machine, where you want to deploy the application, into the BusinessEvents Enterprise Administrator Agent machine repository.

Procedure

1. Select the **Machines** tab.

   The Machine Management page is displayed.

2. Click **Add Machine**.

   The Create Machine page is displayed.

3. Enter the machine details and click **Save**.

Machine Properties

Field	Description
Machine Name	The name of the machine
IP Address	The IP address of the machine. This IP address is used for connecting to the machine.
OS Type	The operating system of the machine is auto-detected. If the operating system is not detected then the default operating system is blank.
Username	The operating system username to be used for all deployment related and start commands using SSH.
Password	The password for the user specified in the **Username** field. The password is stored in the encrypted form. Optional, if certificate based authentication is enabled
SSH Port	The SSH port used by the SSH server on this machine
Default Deployment Path	The path where deployment related artifacts are stored
Add BE Home	Click **Add BE Home** to add a new entry for BusinessEvents installation location. Specify the details of the BusinessEvents installation in the new table that is displayed: • **BE HOME:** The path to the BusinessEvents installation (for example, `C:\tibco\be\<version>`) • **BE TRA:** The path of the master TRA file to use. This file is used as a template while creating instance specific TRA files. The default value is based on the **BE HOME** value: `<BE HOME>\bin\be-engine.tra`. For example, `C:\tibco\be\<version>\bin\be-engine.tra`

Field	Description
	• **BE Version:** The version of the TIBCO BusinessEvents software installed at the specified **BE HOME**. You can also remove a BusinessEvents installation using **Remove** icon, if that **BE HOME** is not used in any instance.
Discover BE Installations	Click **Discover BE Installations** to auto-list all the BusinessEvents installations (version 5.3 or later) in the specified machine. The details of the **BE HOME**, **BE TRA**, and **BE Version** are populated automatically. If required you can add more BusinessEvents installations using **Add BE Home**.

A new machine is added to the BusinessEvents Enterprise Administrator Agent and is listed on the Machine Management page.

Deleting a Machine

Delete a machine from the BusinessEvents Enterprise Administrator Agent machine repository.

Procedure

1. Select the **Machines** tab.

   The Machine Management page is displayed.

2. Click the **Delete** icon  for the machine that you want to delete.

   The delete confirmation prompt is displayed.

3. Click **Delete** to confirm the delete action.

   The confirmation message is displayed: Host is deleted successfully. The machine is now also removed from the machine list on the Machine Management page.

Editing Machine Details

You can edit the machine details on the Machine Management page.

Procedure

1. Select the **Machines** tab.

   The Machine Management page is displayed.

2. Click the **Edit** icon  for the machine that you want to edit.

   The Edit Machine page is displayed.

3. Update the machine details and click **Save**.

   If the instance is deployed, you cannot update the BusinessEvents installation details and **Default Deployment Path**. See Adding a Machine for details on the machine properties.

   The updated machine details are displayed in the machine lists on the Machine Management page.

Uploading External Artifacts

You can upload external artifacts to manage your application through teagent. For example, your application might require some external JAR files to run successfully or you might want to use a custom log4j logging configuration for applications deployed through teagent.

In BusinessEvents Enterprise Administrator agent, you can upload the external artifacts to the selected *BE_HOME*.

Procedure

1. Select the **Machines** tab.

   The Machine Management page is displayed.

2. Open the Upload External Classes window in either of the following ways:

   - Click the **Upload** icon  for the machine which you want to upload the external files.
   - Click the machine name to open the Machine Details page. In the Machine Details page, click **Upload External Artifacts**.

3. Select the **BE Home** to which you want to upload the external files.

4. In the **External File/s** field, click **Browse** to browse the required external file and click **Open**.

   > **Note:** For uploading the multiple external files, compress those files into an archive and then upload that archive.

5. Click **Finish**.

Result

The files are uploaded to the *BE_HOME*/lib/ext/tpcl/beTeagentUpload folder and are included in the classpath of an instance that is started using this **BE Home**.

Auto-detecting BusinessEvents Installations on Host Machines

TIBCO BusinessEvents Enterprise Administrator Agent can auto detect all the TIBCO BusinessEvents installations (version 5.3 or later) on a host machine.

Procedure

1. Select the **Machines** tab.

   The Machine Management page is displayed.

2. In the Machine Management page, you can auto detect BusinessEvents installations (version 5.3 or later) on host machines in any of the following ways:
   - Select the checkboxes of all the machines for which you want to auto detect BusinessEvents installations and click **Discover BE Home**.
   - Click the Edit icon  of the machine for which you want to auto detect BusinessEvents installations, in the Edit Machine page, click **Discover BE Installations** and click **Save**.
   - Click the Discover icon  of the machine for which you want to auto detect BusinessEvents installations.
   - Click the machine name to open the Machine Details page. On the Machine Details page, click **Discover BE Home**.

What to do next

To view all detected TIBCO BusinessEvents installations in a machine, see Viewing All Instances And BusinessEvents Installations On The Host Machine.

Viewing All Instances And BusinessEvents Installations On The Host Machine

In the Machine Management page, if required, you can view all the instances of the application deployment on a machine. You can also view the location of all the TIBCO BusinessEvents installations on that machine.

Procedure

1. Select the **Machines** tab.

   The Machine Management page is displayed.

2. Click the Machine name for which you want to view instances or BusinessEvents installations.

   The host machine details page displays the following tabs:

   - **Instances** - The **Instances** tab displays a list of instances grouped by the application deployments on the machine. You can expand an application deployment to view all its instances. Click an instance name to see details of that instance. You can also select checkboxes of multiple instances and **Start**, **Stop**, or **Kill** those instances.
   - **BE Homes** - The **BE Homes** tab displays a list of all TIBCO BusinessEvents installations (version 5.3 or later) on the machine. It also displays the respective TRA file and TIBCO BusinessEvents version for the BusinessEvents installation.

Application Deployment Management

You can add, edit, import, and delete an application deployment using the BusinessEvents Application Deployment page.

Creating a New Application Deployment

A new application deployment is created to manage and monitor the BusinessEvents application.

Procedure

1. Select the **Application Deployments** tab.

   The **Application Deployment** tab is the default start page for the BusinessEvents Enterprise Administrator Agent UI.

   The BusinessEvents Application Deployment page is displayed.

2. Click **New Deployment**.

   The New Deployment page is displayed to enter the deployment details.

3. Enter the details in the New Deployment page and click **Save**.

Field	Description
Cluster Deployment Descriptor	Click **Browse** and select the CDD file for the BusinessEvents application. For example, `C:\tibco\be\5.3\examples\standard\FraudDetectionCache\FraudDetectionCache\fdcache.cdd` for the FraudDetectionCache application.
Enterprise Archive	Click **Browse** and select the EAR file for the BusinessEvents application. For example, `C:\tibco\be\5.3\examples\standard\FraudDetectionCache\fdache.ear` for the FraudDetectionCache application.
Deployment Name	Enter the deployment name for the BusinessEvents application. By default, the agent takes the EAR file name as the default deployment name.

A new card for the BusinessEvents application is displayed on the BusinessEvents Application Deployments page. The creation confirmation message is displayed: <*Application deployment*> application deployment created successfully.

Exporting Application from TIBCO Administrator

For migration from TIBCO Administrator to TIBCO Enterprise Administrator, the application needs to be exported from TIBCO Administrator in an archive file.

Procedure

1. Run the following command to export the application from TIBCO Administrator.

   ```
   ./AppManage -batchExport -domain domain_name -user username -pw password
   -exportDeployed -dir path_to_store_exported_data
   ```

   The `AppManage.batch` file is created, which contains the name of the application, application EAR file name, and configuration file (XML) name. In the same folder, the application EAR file and the configuration (XML) file are also generated.

2. In the same folder, place a copy of the application CDD file.

   The CDD file name should be the same as the application name. If multiple applications are exported, then all respective CDD files should be placed.

3. Create a `config.csv` file, which contains the machine details, in the same folder.

   The format of the `config.csv` file is:

   ```
   APPLICATIONNAME1,INSTANCENAME1,IP1,JMXPORT
   APPLICATIONNAME1,INSTANCENAME2,IP1,JMXPORT
   APPLICATIONNAME3,INSTANCENAME1,IP2,JMXPORT
   ...
   IP1,OS,DEPLOYMENTPATH,SYSUSER,SYSPASS,SSHPORT
   IP2,OS,DEPLOYMENTPATH,SYSUSER,SYSPASS,SSHPORT
   ...
   ```

   Where:

   - `APPLICATIONNAME`: Name of the application.
   - `INSTANCENAME`: The name of the instance related to the application.
   - `IP`: IP address
   - `JMXPORT`: JMX port of the instance.
   - `DEPLOYMENTPATH`: Path where the application is deployed.
   - `SYSUSER`: Machine username
   - `SYSPASS`: Machine password
   - `SSHPORT`: SSH port of the machine. The default value is 22.

   A sample `config.csv` file is present in the *`BE_HOME`*`\teagent\cli\python` folder.

   > **Note:** For wizard-based import into TIBCO BusinessEvents Enterprise Administrator Agent, the `config.csv` is not mandatory to be present in the archive file of TIBCO Administrator exported application.

4. Compress all the files in an archive file.

Importing TIBCO Administrator Applications

Using TIBCO BusinessEvents Enterprise Administrator agent, you can import an existing TIBCO Administrator application to TIBCO Enterprise Administrator.

Before you begin

Ensure that the deployment that is exported from TIBCO Administrator is in an archive file.

See Exporting Application from TIBCO Administrator for more details.

> **Note:** For wizard-based import, the `config.csv` is not mandatory to be present in the archive file of TIBCO Administrator exported application.

Procedure

1. Select the **Application Deployments** tab.

   The **Application Deployment** tab is the default start page for the BusinessEvents Enterprise Administrator Agent UI.

   The BusinessEvents Application Deployment page is displayed.

2. Click **Import Deployment > TIBCO Administrator Deployment.**

   The Import TRA Deployments page is displayed.

3. In the Import TRA Deployments page, click **Browse** and select the archive that was created after exporting TIBCO Administrator application.

4. Click **Import**.

   The Create TRA Deployments page is displayed with the list of application present in the archive.

5. Select the checkbox for the application that you want to import and click **Proceed**.

6. Verify the machine details and instance details and click **Create**.

   This information is loaded from the `config.csv` file that is included in the archive. Update the information in this page if required.

   The error message is shown after clicking **Create**, if there are any issues with the details.

Result

A new card for the imported application (for example *FraudDetection* is displayed in the BusinessEvents Enterprise Administrator agent.

Importing the BusinessEvents Enterprise Administrator Agent Application

You can import an application that was previously exported from the TIBCO BusinessEvents Enterprise Administrator agent.

Before you begin

The application archive with the application CDD file, EAR file, and configuration XML file, is exported from the BusinessEvents Enterprise Administrator agent. See Exporting Application from TIBCO BusinessEvents Enterprise Administrator Agent for more details.

Procedure

1. Select the **Application Deployments** tab.

   The **Application Deployment** tab is the default start page for the BusinessEvents Enterprise Administrator Agent UI.

   The BusinessEvents Application Deployment page is displayed.

2. Click **Import Deployment > BE TEA-Agent Deployment**.

   The Import Exported Deployment window is displayed.

3. In the Import Exported Deployment window, click **Browse** and select the application archive that was exported from BusinessEvents Enterprise Administrator agent. Click **Open**.

4. Click **Save**.

   If a deployment with the same name is already present, then an error is displayed. When no deployment with the same name exists, a new card for the BusinessEvents application (for example, `FraudDetection`) is displayed on the BusinessEvents Application Deployments page.

Applying the Project Specific Master TRA File

By default, BusinessEvents Enterprise Administrator agent uses the TRA file present in the BusinessEvents installation on the machine for creating instance TRA files. If you want to use a project specific TRA file to configure the instance TRA file, you can specify a master TRA file for that deployment.

After applying the master TRA file, status of all the existing instances changes to **Needs Deployment**, and any newly created instance uses this master TRA file. The master TRA file is machine specific, thus, all the deployment in this machine uses this master TRA file.

Procedure

1. Select the **Application Deployments** tab.

   The **Application Deployment** tab is the default start page for the BusinessEvents Enterprise Administrator Agent UI.

   The BusinessEvents Application Deployment page is displayed.

2. Click the **More Options** icon  for the application deployment for which you want to apply a project specific master TRA file and select **Master Application TRA file**.

   The Master Application TRA file page is displayed.

3. Click **Add New**.

   The fields are displayed for the **Machine** and **TRA file** columns.

4. Select the **Machine** from the dropdown list.

5. Select **Upload**, if you want to upload a project specific TRA file to the host machine. Specify the **Upload Location** (destination path) of the host machine file. Click **Upload** icon to browse and select the TRA file that you want to upload.

6. Alternatively, select **Path** and specify location of the TRA file in the host machine.

   Use this option if the master TRA file is already present in the host machine.

7. Click **Save**.

Editing an Application Deployment

Using the edit option, you can update the existing deployment with a new CDD or a new EAR file.

Procedure

1. Select the **Application Deployments** tab.

   The **Application Deployment** tab is the default start page for the BusinessEvents Enterprise Administrator Agent UI.

   The BusinessEvents Application Deployment page is displayed.

2. Click the **More Options** icon for the application deployment you want to edit and select **Edit Deployment**.

   The Edit Deployment page is displayed.

3. Update the details in the Edit Deployment page and click **Save**.

Field	Description
Cluster Deployment Descriptor	Click **Browse** and select the new CDD file for the BusinessEvents application. For example, `C:\tibco\be\5.3\examples\standard\FraudDetectionCache\FraudDetectionCache\fdcache.cdd` for the FraudDetectionCache application.
Enterprise Archive	Click **Browse** and select the new EAR file for the BusinessEvents application. For example, `C:\tibco\be\5.3\examples\standard\FraudDetectionCache\fdache.ear` for the FraudDetectionCache application.
Deployment Name	Disabled. You cannot update the deployment name for the BusinessEvents application.

   The update confirmation message is displayed: <*Application deployment*> application is edited successfully.

Deleting an Application Deployment

You can delete a deployment defined earlier.
Deleting an application deployment does not delete the associated machine entries from the machine repositories.

Before you begin

All processing unit instances associated for the application should be undeployed before deleting the BusinessEvents application deployment.

Procedure

1. Select the **Application Deployments** tab.

   The **Application Deployment** tab is the default start page for the BusinessEvents Enterprise Administrator Agent UI.

   The BusinessEvents Application Deployment page is displayed.

2. Click the **More Options** icon  for application deployment you want to delete and select **Delete Deployment**.

   The Delete Deployment confirmation page is displayed.

3. Click **Delete Deployment** to confirm the delete command; otherwise, click **Cancel** to cancel the delete command.

   The application deployment card is now removed from the BusinessEvents Application Deployment page. The deletion confirmation message is displayed: <*Application deployment name*> is successfully deleted.

Exporting Application from TIBCO BusinessEvents Enterprise Administrator Agent

You can export the application and all its components such as, CDD file, EAR file, and configuration XML file in an archive file.

Procedure

1. Select the **Application Deployments** tab.

   The **Application Deployment** tab is the default start page for the BusinessEvents Enterprise Administrator Agent UI.

   The BusinessEvents Application Deployment page is displayed.

2. Click the **More Options** icon  and select **Export Deployment**.

   The Save As window is displayed.

3. Browse the folder to which you want to save the archive of the application, provide the archive name, and click **Save**.

   The archive with the application CDD file, EAR file, and configuration XML file with all the configuration details is saved.

Hot-Deploying an BusinessEvents Application

Using the BusinessEvents Enterprise Administrator Agent, you can hot-deploy the EAR file for a running application.

You can hot-deploy an application if hot-deployment is enabled in the CDD file for at least one processing unit. See Enabling Hot Deployment for more details on how to enable hot-deployment for a processing unit.

Procedure

1. Select the **Application Deployments** tab.

   The **Application Deployment** tab is the default start page for the BusinessEvents Enterprise Administrator Agent UI.

   The BusinessEvents Application Deployment page is displayed.

2. Click the **More Options** icon  for the application deployment that you want to hot-deploy and select **Hot Deploy**.

   The EAR Hot Deploy page is displayed.

3. Click **Browse** to select the EAR file for the BusinessEvents application and click **OK**.

   The new EAR file is now deployed at the deployment location.

Deployment Views

The BusinessEvents Enterprise Administrator Agent provides you options to view instances of a deployment grouped on the basis of different parameters.

The following views are defined in the BusinessEvents Enterprise Administrator Agent to group the instances:

- All Instances
- Machines

- Processing Units
- Agent Classes

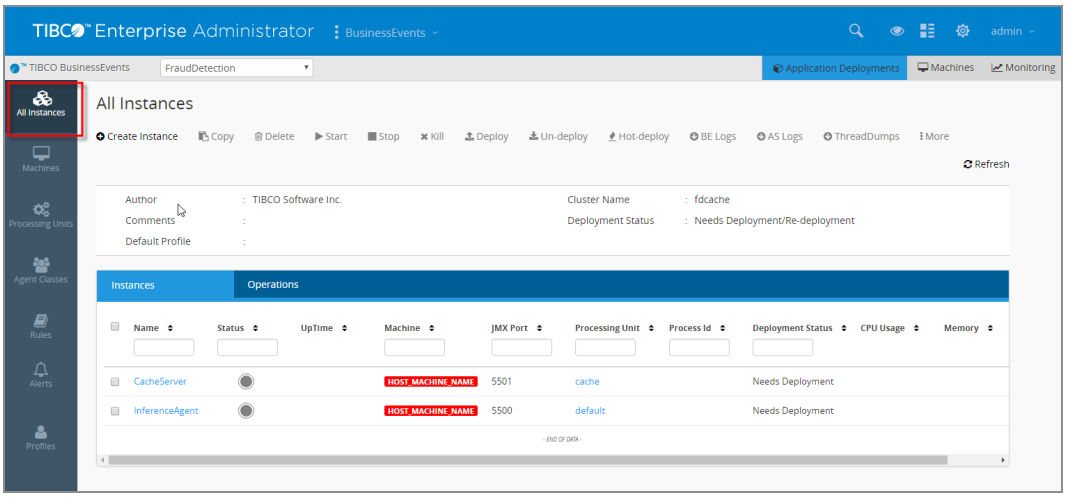
Each view has links to jump to or navigate to related entities. You can track navigation path using the bread crumbs in each view.

In addition to the different instances view, the deployment details page also provides the Rules view and Alerts view.

All Instances View

The All Instances view displays all the instance definitions of the deployment across processing units and machines.

Figure 5: All Instances View



Machines View

The Machines view displays all the machines in the deployment and summary of instances grouped by machines.

Processing Units View

The Processing Units view displays the processing units configured in the CDD and summary of instances grouped by processing units.

Agent Classes View

The Agent Classes view displays the different agent types configured in the CDD, and summary of instances grouped by the Agent Classes defined in the CDD.

Processing Unit Instances Management

For every application deployment you can create instances of the processing unit and deploy them on the associated machines.

Viewing the Instance Management Page

Using the Instance Management page you can manage instances for different processing units of the application. You can also utilize different views to better manage application instances.

Procedure

1. Select the **Application Deployments** tab.

   The **Application Deployment** tab is the default start page for the BusinessEvents Enterprise Administrator Agent UI.

   The BusinessEvents Application Deployment page is displayed.

2. Click the application name for which you want to manage instances.

   The All Instances page is displayed where all the instances of the processing unit for the application are listed.

   You can also select any of the deployment views from the left panel to filter out the instances that you want to manage. See Deployment Views for more details.

Creating an Instance

Define application instances for every application deployment, after they are created. The processing units defined in the CDD file are bound to a machine (from the machine repository), where the instance runs.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. In the All Instances page, click **Create Instance** to create an application instance.

   The PU Instance Creation page is displayed where you can enter details for the instances.

2. Enter the instance details and click **Save**.

Processing Unit Instance Properties

Field	Description
Instance Name	The name of the instance
Processing Unit	Select the processing unit for which you want to create the instance. The dropdown is populated with the processing units defined in the application CDD file.
Machine Name	Select the machine on which you want to deploy. The dropdown lists all machines registered in the machine repository. See Machine Management.
BE Home	Select the *BE_HOME* from list of all the *BE_HOME* for the selected **Machine Name**. The list contains all *BE_HOME* of the BusinessEvents version that matches with the application BusinessEvents version. The BusinessEvents engine is started from the selected *BE_HOME*.
JMX Port	The JMX port that is used to communicate with the instance. The JMX port value should not clash with any other port on the machine specified in the **Machine Name**. The field is auto-filled with an unused port of the machine or existing highest port plus one. For example, if 5501, 5504, and 5506 ports are used then the new port used is 5507. You can change the port number if you do not want to use the suggested port.
JMX User Name	Optional. Specify the JMX username for user authentication for the JMX connections from the BusinessEvents Enterprise Administrator Agent to the instance. Additional configuration is required to activate the JMX authentication. See Authentication and SSL Configurations for more details.
JMX Password	Optional. Specify the password for the JMX username specified in the **JMX User Name**.

Field	Description
	Additional configuration is required to activate the JMX authentication. See Authentication and SSL Configurations for more details.
Deployment Path	The file path in the machine where deployed artifacts are stored. The deployment artifacts include the EAR file, the CDD file, the instance specific TRA file, and the batch files to start the instance. The default path is the **Default Deployment Path** from the associated machine.

The instance listed under the **Instances** tab. Also, the instance creation confirmation message is displayed: <*Instance Name*> instance is created successfully.

Updating an Instance

For undeployed instances you can update its details to associate a different machine or different processing unit.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. Click the **Edit** icon  for the instance you want to edit.

   Tip: In smaller screens you have to scroll the instance list horizontally to see the **Edit** icon.

   The Edit PU Instance page is displayed.

2. Update the details for the instance and click **Save**.

   Update to the **Instance Name** field is disabled; however, you can update all other details if the instance is not deployed. See Processing Unit Instance Properties for more details on the properties.

   The instance update confirmation message is displayed: *<Instance Name>* instance is edited successfully.

Copying an Instance

You can create similar instances for the application using the Copy option.

The copy operation copies all the system properties, JVM properties, global variables, and log settings from the source instance. You can also update the JVM properties, system properties, and global variables after the copy process is complete.

Note: The copy operation does not copy the deployment artifacts, but copies only the application definition and properties. The new instance needs to be deployed to generate deployment artifacts.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. Select the checkbox for the instance you want to copy and click **Copy**.

   The Copy Instance page is displayed.

2. Update the details for the instance and click **Save**.

   See Processing Unit Instance Properties for more details on the properties.

   The instance copy confirmation message is displayed: *<Instance Name>* instance is cloned successfully.

Deleting an Instance

You can delete an instance if it is not deployed.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. Click the **Delete** icon  for the instance you want to delete.

   Tip: In smaller screens you have to scroll the instance list horizontally to see the **Delete** icon.

   The Delete Confirmation page is displayed.

2. Click **Delete**.

   The instance is deleted from the instance list. The instance deletion confirmation message is displayed: *<Instance Name>* instance is deleted successfully.

Deploying Processing Unit Instances

Using the BusinessEvents Enterprise Administrator Agent, you can deploy the processing unit instances to the associated machine.
Deployment includes copying deployment artifacts, such as the CDD file, the EAR file, an instance-specific TRA file, and shell scripts (or Windows batch file) to the deployment path of the associated machine. You can deploy one or more instances at the same time to their associated machines.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. Select the checkboxes for all the instances that you want to deploy.
2. Click **Deploy**.

   The instances are now deployed and their deployment status is now changed to Deployed.

Undeploying Processing Unit Instances

You can undeploy processing unit instances that were previously deployed. You can undeploy only stopped instances.
Undeploying processing unit instances includes deleting deployment artifacts, such as the CDD file, the EAR file, an instance-specific TRA file, and shell scripts (or Windows batch file) from the deployment path of the associated machine. You can undeploy one or more instances at the same time from their associated machines.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. Select the checkboxes for all the instances that you want to undeploy.
2. Click **Undeploy**.

   The instances are now undeployed and their deployment status is now changed to Needs Deployment.

Starting Processing Unit Instances

Using the BusinessEvents Enterprise Administrator Agent, you can start processing unit instances that were previously deployed.

An `ssh` command is used to start the instance-specific shell script at the deployment location.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. Select the checkboxes for all the deployed instances that you want to start.
2. Click **Start**.

   The instances are now started and their status is now changed to Running.

Stopping Processing Unit Instances

Using the BusinessEvents Enterprise Administrator Agent, you can gracefully stop processing unit instances that were previously running.

A JMX MBean is used to stop the instance.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. Select the checkboxes for all the running instances that you want to stop.
2. Click **Stop**.

   The instances are now stopped and their status is now changed to Stopped.

Killing an Instance

Using the BusinessEvents Enterprise Administrator Agent, you can forcefully stop processing unit instances that were previously running (uses kill -9 to stop instances on Unix).

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. Select the checkboxes for all the running instances that you want to stop forcefully.
2. Click **Kill**.

   The instances are now stopped and their status is now changed to Stopped.

Invoking Instance Operations

Using the BusinessEvents Enterprise Administrator Agent, you can invoke several MBeans methods for an instance for management and monitoring.

The BusinessEvents Enterprise Administrator Agent lists the MBeans methods in different categories. You can select the instance for which you want to invoke the method and perform the operation.

You can configure the list of methods displayed and update the list with the methods you require. Restart the agent after configuring the list of MBeans methods to display in the BusinessEvents Enterprise Administrator Agent interface.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. Select the **Operations** tab.

   The Operations tab is displayed which lists different categories of the MBeans methods on the left panel.

2. Select the required category to expand it.

   The MBeans methods for that category are listed.

3. Select the method you want to invoke.
4. (Optional) Enter the values in the fields for the method to filter the result.
5. Click **Invoke**.

   The output for the method is displayed in the same page.

Downloading Thread Dumps for Instances

Using the BusinessEvents Enterprise Administrator Agent, you can download thread dumps of multiple instances as a ZIP file.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. Select the checkboxes for the instances for which you want to download thread dumps.
2. Click **ThreadDumps**.

   A `.zip` file is downloaded to your machine, where the browser is running, containing thread dumps for all the selected instances.

Downloading Log Files for Instances

By using the BusinessEvents Enterprise Administrator Agent, you can download log files for multiple instances as a compressed file.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. Select the checkboxes for instances for which you want to download log files.
2. Click **BE Logs** or **AS Logs** to download BusinessEvents logs and Legacy ActiveSpaces logs respectively, for selected instances.

   A `.zip` file is downloaded to your machine, where the browser is running, containing log files for all the selected instances.

Viewing Log File for Instance

By using the BusinessEvents Enterprise Administrator Agent, you can view log files for a instance.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. In the All Instance page, click the instance name for which you want to view the logs.

   The Instance Configuration page is displayed with all the properties.

2. Click **BE Log** or **AS Log** to view the BusinessEvents log or Legacy ActiveSpaces log for the instance respectively.

Hot-Deploying an Instance

Using the BusinessEvents Enterprise Administrator Agent, you can hot-deploy the enterprise archive file for the running processing unit instance.
You can hot-deploy a processing unit instance if the hot-deployment is enabled in the CDD file of the processing unit. See Enabling Hot Deployment for more details on how to enable the hot-deployment for a processing unit.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. Select the checkbox for the instance that you want to hot-deploy and click **Hot-deploy**.

   The EAR Hot Deploy page is displayed.

2. Click **Browse** to select the EAR file for the BusinessEvents application and click **OK**.

   The new enterprise archive file is now deployed at the deployment location.

Hot-Deploying Classes and Rule Template Instances

You can hot-deploy the decision table and rule template classes also using the BusinessEvents Enterprise Administrator Agent in addition to TIBCO BusinessEvents

WebStudio.

Before you begin

Navigate to any of the deployment views, for instance, management. See Viewing the Instance Management Page for more details.

The CDD file should contain the `be.engine.cluster.externalClasses.path` and `be.cluster.ruletemplateinstances.deploy.dir` properties for decision table and rule template instance hot-deployment, respectively, at the cluster level. The value of the property specifies the location where the `.zip` file or JAR file for hot-deployment of decision table classes and rule template instances are uploaded.

Procedure

1. On the All Instance page, click the instance name for which you want to configure the properties.

   The Instance configuration page is displayed with all the properties.

2. Click **Hot Deploy Operations** and select either **Decision Table Classes** for hot-deploying decision table classes or **Rule Template** for rule template instances, respectively.

   > **Note:** These options are enabled only if the `be.engine.cluster.externalClasses.path` property (for the decision table) and the `be.cluster.ruletemplateinstances.deploy.dir` property (for the rule template instance) are present in the CDD file at the cluster level.

3. Click **Browse** to select the `.zip` file for the decision table or rule template, and click **OK**.

   The decision table classes and rule template instances files are now deployed at the location specified in the `be.engine.cluster.externalClasses.path` and `be.cluster.ruletemplateinstances.deploy.dir` properties, respectively.

Processing Unit Instance Configuration

You can configure different properties of the instances including the log levels and global variables using the BusinessEvents Enterprise Administrator Agent UI.

Click any instance name to open the page listing the properties instance categorized in different tabs. Each tab represents a specific set of properties. After editing the configuration properties, redeploy the instances for the changes to take effect.

Global Variables

The **Global Variables** tab lists global variables for the instance with their default values, which are taken from the CDD if present, else as defined in the project. Ensure that the "Service Settable" flag in the project is enabled for editing global variables using the BusinessEvents Enterprise Administrator Agent.

You can also override the global variable value and then redeploy the instance to make the change permanent. If needed, after deployment, you can also delete the override value using the BusinessEvents Enterprise Administrator Agent.

System Properties

The **System Properties** tab lists system properties for the instance. You can add, update, or delete the system properties. The instance needs redeployment for the changes to take effect.

BusinessEvents Properties

Using the **BusinessEvents Properties** tab you can add or override the effective CDD value of the BusinessEvents properties.

In CDD the BusinessEvents properties are specified in the following locations:

- In property groups in CDD PU sections
- In property groups in CDD agent sections
- In property groups at the "Cluster" (the top level) in the CDD

For properties with the same name specified in multiple places in the CDD, the PU level value overrides the agent level value which overrides the Cluster level value. This is called the effective CDD value.

You can override the value for the effective CDD value of a BusinessEvents property using the BusinessEvents Enterprise Administrator Agent. After deployment, the override value is placed in the instance TRA file. You can also delete the override value using the BusinessEvents Enterprise Administrator Agent.

In addition to the default CDD properties, you can also add new properties and also delete these newly added properties.

JVM Properties

You can update the following JVM properties for the instance:

- Initial heap size (-Xms)
- Maximum heap size (-Xmx)

Deploy the instance at least once before updating these properties. Thus, the default values are loaded from the target machine `be-engine.tra` file.

Log Levels

You can also change the log level of an instance using the BusinessEvents Enterprise Administrator Agent. You can specify multiple logger patterns and set a level for each of the patterns. The run time will evaluate these patterns and log levels are applied accordingly. Log level changes can be applied directly to running instances or they can be deployed, so that the changes are permanent.

Group Operation Support

You can also update configuration properties for multiple instances as a group operation.

Configuring Global Variables of an Instance

Using BusinessEvents Enterprise Administrator Agent, you can override the value of a global variable for an instance.

> **Note:** If you use a global variable for a *Cluster Name* field in the CDD file, you cannot override it through the BusinessEvents Enterprise Administrator Agent interface or the command-line interface.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. In the All Instance page, click the instance name for which you want to configure the properties.

   The Instance configuration page is displayed with all the properties.

2. In the **Configuration** tab, select **Global Variables**.

   All the global variables for the instance are listed in the page with their default value.

3. Select the **Override** option and update the **New Value** column of the global variable with a new value.

4. Click **Save** to save the changes.

   The **Deployment Status** of the instance is now changed to Needs Deployment.

What to do next

Redeploy the instance to apply the changes to the instance (see Deploying Processing Unit Instances). After redeployment, the updated value of the global variable is listed in the **Deployed Value** column.

Configuring System Properties of an Instance

Using BusinessEvents Enterprise Administrator Agent you can configure system properties for an instance.

You can now perform the following tasks to configure system properties of the instance:

- Override the value of a system property. See Overriding the Value of a System Property.
- Delete a previously deployed override value. See Deleting the Override of a System Property.
- Add a new system property. See Adding a New System Property for an Instance.
- Remove a previously added system property. See Removing a System Property for an Instance.

Overriding the Value of a System Property

You can specify a new value to override the existing value of the system property during run time using the BusinessEvents Enterprise Administrator agent. The new value is

effective only after the instance is redeployed.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. In the All Instance page, click the instance name for which you want to configure the properties.

   The Instance configuration page is displayed with all the properties.

2. In the **Configuration** tab, select **System Properties**.

   All the system properties for the instance are listed in the page with their default value.

3. Select the **Override** option and update the **New Value** column for the system property with a new value.

4. Click **Save** to save the configuration changes.

   The **Deployment Status** of the instance is now changed to Needs Deployment.

What to do next

Redeploy the instance to apply the changes to the instance (see Deploying Processing Unit Instances). After redeployment, the updated value of the System property is listed in the **Deployed Value** column.

Deleting the Override of a System Property

You can also delete a previously applied override value to the system property. The existing deployed value is listed in the **Deployed Value** column. After the delete operation, the value of the system property is changed back to the default value. The override value is deleted only after the instance is redeployed.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. On the All Instance page, click the instance name for which you want to configure the properties.

   The Instance configuration page is displayed with all the properties.

2. In the **Configuration** tab, select **System Properties**.

   All the system properties for the instance are listed in the page with their default value.

3. Select the **Delete** option to the delete the override value (**Deployed Value**) and change the system property back to the value specified in the **Default Value** column.

4. Click **Save** to save the configuration changes.

   The **Deployment Status** of the instance is now changed to Needs Deployment.

What to do next

Redeploy the instance to apply the changes to the instance (see Deploying Processing Unit Instances). After redeployment, the updated value of the System property is listed in the **Deployed Value** column.

Adding a New System Property for an Instance

In addition to properties already listed, you can add new custom system property.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. On the All Instance page, click the instance name, for which you want to configure the properties.

   The Instance configuration page is displayed with all the properties.

2. In the **Configuration** tab, select **System Properties**.

   All the system properties for the instance are listed in the page with their default values.

3. Click **Add New** to add a new property.

   A new entry is added to the list with empty Name and Value.

4. Enter **Name** and **Value** for the new property, and click **Save**.

   The **Deployment Status** of the instance is now changed to Needs Deployment.

What to do next

Redeploy the instance to apply the changes to the instance (see Deploying Processing Unit Instances). After redeployment, the updated value of the System property is listed in the **Deployed Value** column.

Removing a System Property for an Instance

You can remove a system property that you had previously added using the BusinessEvents Enterprise Administrator Agent.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. In the All Instance page, click the instance name, for which you want to configure the properties.

   The Instance configuration page is displayed with all the properties.

2. In the **Configuration** tab, select **System Properties**.

   All the system properties for the instance are listed in the page with their default value.

3. Click the **Remove** icon  for the property that you want to remove.

   The **Remove** icon is displayed only for those properties that are added from the BusinessEvents Enterprise Administrator Agent.

4. Click **Save** to save the configuration changes.

   The **Deployment Status** of the instance is now changed to Needs Deployment.

What to do next

Redeploy the instance to apply the changes to the instance (see Deploying Processing Unit Instances). After redeployment, the updated value of the System property is listed in the **Deployed Value** column.

Configuring BusinessEvents Properties of an Instance

Using BusinessEvents Enterprise Administrator Agent, you can configure BusinessEvents properties for an instance.

You can now perform the following tasks to configure BusinessEvents properties of the instance:

- Override the value of a BusinessEvents property. See Overriding the Effective CDD Value of a BusinessEvents Property.
- Delete a previously deployed override value. See Deleting the Override of a BusinessEvents Property.
- Add a new BusinessEvents property. See Adding a New BusinessEvents Property.
- Remove a previously added BusinessEvents property. See Removing a BusinessEvents Property.

Overriding the Effective CDD Value of a BusinessEvents Property

You can specify a new value to override the effective CDD value of the BusinessEvents property during run time using the BusinessEvents Enterprise Administrator Agent. The new value is effective only after the instance is redeployed.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. In the All Instance page, click the instance name for which you want to configure the properties.

   The Instance configuration page is displayed with all the properties.

2. In the **Configuration** tab, select **System Properties**.

   All the BusinessEvents properties for the instance are listed in the page with their default value.

3. Select the **Override** option and update the **New Value** column for the BusinessEvents property with a new value.

4. Click **Save** to save the configuration changes.

   The **Deployment Status** of the instance is now changed to Needs Deployment.

What to do next

Redeploy the instance to apply the changes to the instance (see Deploying Processing Unit Instances). After redeployment, the updated value of the BusinessEvents property is listed in the **Deployed Value** column.

Removing a BusinessEvents Property

You can remove a BusinessEvents property that you had previously added using the BusinessEvents Enterprise Administrator Agent.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. In the All Instance page, click the instance name, for which you want to configure the properties.

   The Instance configuration page is displayed with all the properties.

2. In the **Configuration** tab, select **BusinessEvents Properties**.

   All the BusinessEvents properties for the instance are listed in the page with their default value.

3. Click the **Remove** icon  for the property that you want to remove.

   The **Remove** icon is displayed only for those properties that are added from the BusinessEvents Enterprise Administrator Agent.

4. Click **Save** to save the configuration changes.

   The **Deployment Status** of the instance is now changed to Needs Deployment.

What to do next

Redeploy the instance to apply the changes to the instance (see Deploying Processing Unit Instances). After redeployment, the updated value of the BusinessEvents property is listed in the **Deployed Value** column.

Adding a New BusinessEvents Property

In addition to properties already listed, you can add new custom BusinessEvents property.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. In the All Instance page, click the instance name, for which you want to configure the properties.

   The Instance configuration page is displayed with all the properties.

2. In the **Configuration** tab, select **BusinessEvents Properties**.

   All the BusinessEvents properties for the instance are listed in the page with their default value.

3. Click **Add New** to add a new property.

   A new entry is added to the list with empty Name and Value.

4. Enter **Name** and **Value** for the new property, and click **Save**.

   The **Deployment Status** of the instance is now changed to Needs Deployment.

What to do next

Redeploy the instance to apply the changes to the instance (see Deploying Processing Unit Instances). After redeployment, the updated value of the BusinessEvents property is listed in the **Deployed Value** column.

Deleting the Override of a BusinessEvents Property

You can also delete a previously applied override value to the BusinessEvents property. The existing deployed value is listed in the **Deployed Value** column. After the delete operation, the value of the BusinessEvents property is changed back to the default value. The override value is deleted only after the instance is redeployed.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. In the All Instance page, click the instance name, for which you want to configure the properties.

   The Instance configuration page is displayed with all the properties.

2. In the **Configuration** tab, select **BusinessEvents Properties**.

   All the BusinessEvents properties for the instance are listed in the page with their default value.

3. Select the **Delete** option to the delete the override value (**Deployed Value**) and change the BusinessEvents property back to the value specified in the **Default Value** column.

4. Click **Save** to save the configuration changes.

   The **Deployment Status** of the instance is now changed to Needs Deployment.

What to do next

Redeploy the instance to apply the changes to the instance (see Deploying Processing Unit Instances). After redeployment, the updated value of the BusinessEvents property is listed in the **Deployed Value** column.

Configuring JVM Properties of an Instance

Using BusinessEvents Enterprise Administrator Agent, you can update the values of the JVM properties for an instance.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. In the All Instance page, click the instance name, for which you want to configure the properties.

   The Instance configuration page is displayed with all the properties.

2. In the **Configuration** tab, select **JVM Properties**.

   The following JVM properties for the instance are listed in the page:

   - Max Heap Size
   - Initial Heap Size

3. Enter new values for these JVM properties and click **Save**.

   The **Deployment Status** of the instance is now changed to Needs Deployment.

What to do next

Redeploy the instance to apply the changes to the instance (see Deploying Processing Unit Instances). After redeployment, the updated value of the JVM property is listed in the **Deployed Value** column.

Configuring Log Levels of an Instance

Using BusinessEvents Enterprise Administrator Agent, you can change the log level of an instance. You can either apply those changes to instances on run time only and not save them for future runs. Otherwise, you can also apply the changes to the instance after deployment so that the changes are persistent for future runs of the instance.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. In the All Instance page, click the instance name, for which you want to configure the properties.

   The Instance configuration page is displayed with all the properties.

2. In the **Configuration** tab, select **Log Levels**.

   Two sections: "Runtime" and "Deploy" are displayed for configuring the log levels.

3. If you want to apply a log level to only the running instance, perform the following steps in the Runtime section:

   a. Enter a new **Pattern** and select **Runtime Log Level** as required.

   b. Click **Apply**.

   The selected log level for the specified pattern is applied to the running instance. Redeployment is not required for the new log level to be effective.

4. If you want to apply a log level after deployment, perform the following steps in the Deploy section:

   a. Click **Add New**.

   b. Enter a new **Pattern** and select **New Log Level**.

   c. Click **Save**.

   The **Deployment Status** of the instance is now changed to Needs Deployment.

What to do next

If the new log level is applied under the Deploy section, redeploy the instance to apply the changes to the instance (see Deploying Processing Unit Instances). After redeployment, the updated log level is listed in the **Deployed Log Level** column.

Disabling Jetty Server Logs

In BusinessEvents Enterprise Administrator agent, if required you can disable Jetty server log when the log level is set to *debug* or *all*.

Procedure

1. Navigate to the *BE_HOME*/teagent/config folder and open log4j.properties for editing.
2. Add the following property and save the log4j.properties file to disable the Jetty server logs.

```
log4j.category.org.eclipse.jetty=error
```

Configuring Properties for Multiple Instances

Using BusinessEvents Enterprise Administrator Agent, you can apply configuration changes for properties (with the same value) to multiple instances without opening each instance.

Before you begin

Navigate to any of the deployment views for instance management. See Viewing the Instance Management Page for more details.

Procedure

1. In the All Instance page, select the instances, for which you want to configure the properties.
2. Click **More Operations** and select the property category, that you want to configure. The options are:
   - **Global Variables**
   - **System Properties**
   - **BusinessEvent Properties**
   - **JVM Properties**
   - **Deployed Log Levels**
   - **Runtime Log Levels**

   The Properties Configuration page is displayed with two tabs: **Same value properties** and **Different value properties**. The **Same values properties** tab displays the properties, which have the same value for all the select instances, and their value. The **Different value properties** tab displays different values of the properties for selected instances.

3. Select the **Same value properties** tab to update the properties to the same value to a new value. Enter the new value and click **Save**.

   The **Deployment Status** of the selected instances is now changed to Needs Deployment.

4. Select the **Different value properties** tab to update the properties with the different values, to a same common value. Select the **Change Value** checkbox, enter the new value, and click **Save**.

   The **Deployment Status** of the selected instances is now changed to Needs Deployment.

What to do next

Redeploy the instances again to apply the changes to instances (see Deploying Processing Unit Instances). After redeployment, the updated value of the properties are listed in the **Deployed Value** column.

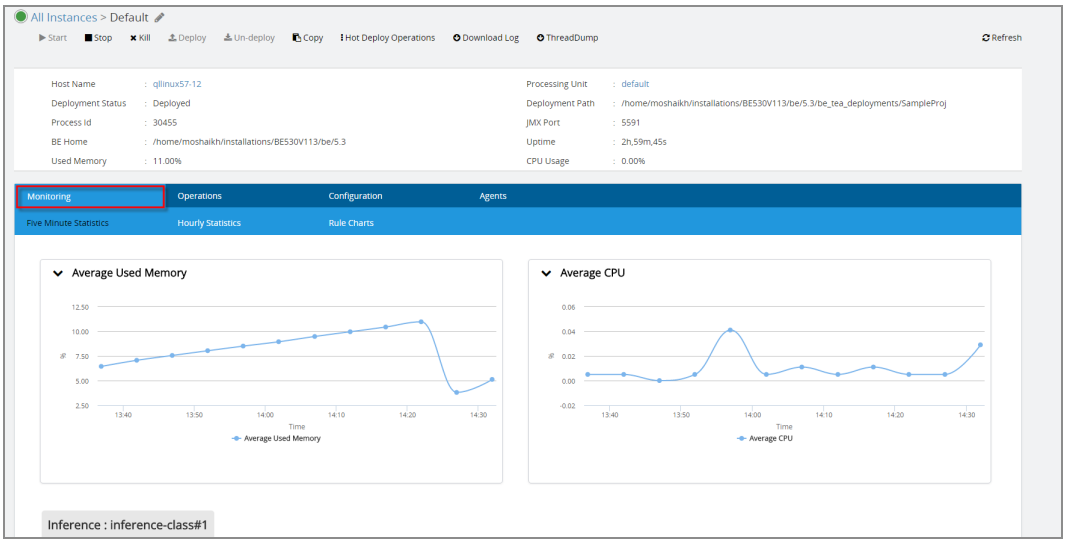
Instance Monitoring Charts

Using the BusinessEvents Enterprise Administrator Agent, you can view and monitor instances performance based on performance indicators.

The BusinessEvents Enterprise Administrator Agent provides various chart to monitor performances of instances. The charts are displayed only if the instance is running. In any of the deployment view, click on a running instance and select the **Monitoring** tab to view the charts for the instance. The three group of charts in the BusinessEvents Enterprise Administrator Agent are as follows:

- Five-Minute Statistics
- Hourly Statistics
- Rule Charts

Figure 6: Monitoring Charts for the Instance



Five-Minute Statistics Charts

These charts display the instance performance aggregated over an interval of five minutes. The five-minute charts are plotted for the last one hour.

The following charts are displayed by default:

Average Used Memory

The chart displays the five-minute averages of the used memory percentage.

Average CPU

The chart displays the five-minute averages of the CPU consumption.

Average RTC Transaction: Latency

The chart displays the five-minute averages of the RTC transaction latency.

Average RTC Transaction: Throughput

The chart displays the five-minute averages of the RTC transaction throughput.

Total Locks Held

The chart displays the five-minute averages of the totals of locks, local locks, and cluster locks.

Event Throughput

The chart displays the five-minute averages of the total number of events asserted in the inference engine of the specified destinations. You can specify the URI of destinations, to be monitored, in the BusinessEvents Enterprise Administrator Agent configuration file. See Configuring Destinations for Event Throughput Chart for more details.

RTCTransactionManagerReport:

The chart displays the five-minute statistics of :

- Totals successful transactions
- Total time for successful transactions
- Total number of errors
- Total database Queue Wait time
- Total number of database operations completed
- Total number of cache transactions
- Pending numbers of Cache Writes
- Pending numbers of Database writes
- Pending numbers of actions
- Numbers of locks waiting to be released
- Numbers of events pending to be acknowledged

Individual Agent Statistics:

The chart is dependent on the Entity URI and displays only for those entities that are selected in CDD. The chart displays the five-minute statistics of:

- Entity Average time in RTC
- Entity Average time in Post RTC
- Entity Average time in Pre RTC
- Number of hits in L1Cache
- Number of misses in L1Cache

Cache Statistics:

The chart is dependent on the Entity URI and displays only for those entities that are selected in CDD. The chart displays the five-minute statistics of:

- Total number of Put operations
- Total number of Get operations
- Total number of removal operations performed on the cache
- Average time to execute Put
- Average time to execute Get
- Average time to execute removals

Overall Inference Agent Statistics:

The chart displays the five-minute statistics of the totals of

- Total number of transactions published
- Total number of transactions subscribed
- Average time for publishing the transactions
- Average time for subscribing to the transactions

Note:

- To view one or more parameters on the chart, select the parameter. If you do not select any parameter all are displayed on the chart by default.
- The paramters are displayed on the chart depending on the configured query and the data in the InfluxDB.
- You can view all the entities and their parameters on a single chart. But it is recommended not to configure all the entities on a single chart.
- The following charts are displayed only for inference agents:
  - Average RTC transaction: latency
  - Average RTC transaction: throughput
  - Total locks held
  - Event throughput

Hourly Statistics Charts

The same charts as under the Five-Minute Statistics are rendered, except that the aggregation interval is one hour instead of five minutes. The hourly charts are plotted for up to one day old.

Rule Charts

The rule chart shows the worst performing rules.

You can control the number of worst performing rules to be displayed in the chart by adding a `<maxDataPoints>` property in the `EntityMetricViewConfig.xml` file. The `EntityMetricViewConfig.xml` file is located in the `BE_HOME/teagent/config` folder.

```
<section sectionId="3" displayName="Rule Statistics">
      <chart>
         <id>13</id>
         <chartType>column</chartType>
         <entity>agent</entity>
         <name>executionTimeChart</name>
         <description>chart which avg rule execution time per
rule</description>
          <……….>
          <maxDataPoints>10</maxDataPoints>
          <……….>
```

Note: This chart requires rules statistics data collection, which is not enabled by default for performance reasons. To enable it, set the following BusinessEvents properties in the processing unit instances:

- `com.tibco.be.metric.publish.enable=true`
- `be.stats.enabled=true`

See Configuring BusinessEvents Properties of an Instance to configure BusinessEvents properties for an instance.

Configuring Destinations for Event Throughput Chart

You can configure the destinations for a specific application, for which you want to view the Event Throughput chart.

Procedure

1. Navigate to the location *BE_HOME*/teagent/config/ and open the beEntityMap.xml file for editing.
2. Add the entry of the destination to be monitored for a specific application in the following syntax.

```
<app name="APPLICATION_NAME" >
       <entity-group type="destination" >
                <entity name="DESTINATION_NAME" alias="DESITNATION_ALIAS" />
       </entity-group>
</app>
```

   where,

   - *APPLICATION_NAME* - Name of the application which holds the destinations.
   - *DESTINATION_NAME*: URI of the destination to be monitored.
   - *DESTINATION_ALIAS* - Update this attribute if an alias is to be shown in the chart for a particular destination.

   Note:
   - For an application if the entry is not present in the configuration file, event throughput chart displays results for all destinations.
   - The property max-series in the beEntityMap.xml file specifies the number of chart series to be shown. Update this property to limit the number of series.

Rules and Alerts

In BusinessEvents Enterprise Administrator Agent, you can create rules on *monitored entities* based on their metrics.

The *monitored entities* are those entities for which certain metrics are computed. Using the BusinessEvents Enterprise Administrator Agent, you can write rules on these monitored entities, such that in the case a change in a specified metric, an action takes place or an alert is generated.

You can specify *set conditions* and *set actions* while authoring rules. When the specified set conditions are satisfied, the defined set actions of the rule are triggered. For example, you

can create a rule to *mark the cluster health as critical* when *the total number of running processing unit instances is less than 50% of the total number of processing unit instances.*

You can also specify *clear conditions* and *clear actions* to counter the effects of the set condition and set actions. When the clear conditions are met, their associated clear actions are triggered. Clear conditions and clear actions are enabled only if the corresponding set conditions are satisfied. For example, you can create a clear action to *mark the cluster health as normal* when *the total number of running processing unit instances is more than or equal to 50% of the total number of processing unit instances.*

You can also author a nested expression for the set condition and clear condition using the `AND` and `OR` operators.

The BusinessEvents Enterprise Administrator Agent provide two different views for Rules and Alerts.

Rules View

The Rules view displays the list of rules for the application deployment. The Rules view provides the option to configure rules and also to disable them if required.

Rules which determine the health of a monitored entity are called *health rules*. Health rules are considered as global rules spanning across the user base and since the health of a system cannot be different as seen by different users, only users with the RULE_AUTHOR_ADMIN role can author *health rules*.

Additionally, users with the RULE_AUTHOR_ADMIN role can view and modify rules created by non RULE_AUTHOR_ADMIN privileged users. If a rules admin user edits a non-admin rule and introduces a health action, the rule ownership is transferred to the rule admin user.

Alerts View

The Alerts view lists all the alerts generated by the rules. You can see the number of new generated alerts on the Alerts tab icon. In the Alerts view, new alerts are displayed using the bold font. The font of new alerts is changed back to the normal font, after clicking **Refresh** or after switching to another view.

You can use the column filter to view only the specific alerts. If required you can also clear out all alerts using the **Clear Alerts** icon. You can view alerts generated by your own rules or generated by RULE_ADMIN users. Similarly, you can clear only those alerts which are generated by your rules. The RULE_ADMIN users can view and clear all users alerts. These alerts are stored in memory, so in case of BusinessEvents engine restart, the previous are alerts are lost.

Creating an Alert Rule

You can create rules based on the metrics on monitored entities to generate alerts on certain conditions.
In the alert rule, you specify a set condition to take action if the condition is met. However, you must specify a clear condition and an associated clear action as well so that you can undo the earlier action.

Procedure

1. In the Rules view, click **Create New Rule**.

   The Create Rule wizard is displayed.

2. Enter the details for the new rules and click **Next**.

Field	Description
Name	The unique name of the rule. The agent displays an error if the rule with the same name exists for the application.
Monitored Entity	The entities for which metrics are computed. The values are: • Cluster • Processing Unit • RTC Transactions • Event Throughput • BusinessEvents Rules
Description	The summary for the rule

   The Set Conditions page of the wizard is displayed.

3. Enter the details to create the set condition and click **Next**.

Field	Description
Left dropdown 	The metrics of the monitored entity selected earlier. See Monitored Entities Reference for more details on the metrics for all monitored entities.
Center dropdown 	The comparison operators for the condition. The values are: • == • >= • >< • <= • <
Right text box 	The value of the metrics for creating the set condition.
Expression 	The option to add more conditions to create a complex expression. Multiple conditions are combined using the `AND` and `OR` operators.

   The Set Action page of the wizard is displayed.

4. Click **Add New**.

   The set action details page of the wizard is displayed.

5. Select the set action type, enter the action details, and click **Save**.

Action Type	Description

Set-Health-Action	Sets the health of the instance if the condition is satisfied. You need a special privilege to create health action. See Rules View for more details. The fields for health action are:
	• **Alert Level** - The level of the alert. The values are:
	  ◦ High
	  ◦ Medium
	  ◦ Normal
	  ◦ Low
	• **HealthValue** - The health status of the instance to be set if the condition is satisfied. The values are:
	  ◦ Critical
	  ◦ Warning
	  ◦ Normal
	• **Alert Text** - The text that is displayed in the alert. You can use alert tokens to insert dynamic values in alert text. See Alert Tokens Reference for list of alert tokens available in the BusinessEvents Enterprise Administrator Agent.

Email-Action	Sends an email to the specified recipients if the set condition is satisfied. Configure the SMTP server details in the BusinessEvents Enterprise Administrator Agent for the email action to work, see Email Action Configurations. The fields of the email action are:

Action Type	Description

	• **Alert Level** - The level of the alert. The values are:
	  ◦ High
	  ◦ Medium
	  ◦ Normal
	  ◦ Low
	• **To** - The email address of the primary recipient of the email
	• **Cc** - The email address of the recipient to which the email is copied
	• **Bcc** - The email address of the recipient to which the email is blind copied
	• **Subject** - The subject of the email
	• **Body** - The main body of the email

Log-Action	Logs the alert text to the logs, if the set condition is satisfied. The fields of the log action are:
	• **Alert Level** - Level of the alert. The values are:
	  ◦ High
	  ◦ Medium
	  ◦ Normal
	  ◦ Low
	• **Alert Text** - The text that is displayed in the alert. You can use alert tokens to insert dynamic values in alert text. See Alert Tokens Reference for list of alert tokens available in the BusinessEvents Enterprise Administrator Agent.

Alert-Only	Generates only the alert and takes no other action. The fields are:

Action Type	Description

	• **Alert Level** - Level of the alert. The values are:
	  ◦ High
	  ◦ Medium
	  ◦ Normal
	  ◦ Low
	• **Alert Text** - The text that is displayed in the alert. You can use alert tokens to insert dynamic values in the alert text. See Alert Tokens Reference for list of alert tokens available in the BusinessEvents Enterprise Administrator Agent.

The new action is listed in the Set Action page of the wizard.

6. If required, you can add more actions or click **Next**.

   The Clear Condition page of the wizard is displayed.

7. Set the clear condition similar to the set condition that was created earlier (see *Step 3*), and click **Next**.

   The Clear Action page of the wizard is displayed.

8. Set the clear action similar to the set action that was defined earlier (see *Step 4*), and click **Save**.

   A new rule is created with specified conditions and actions and now listed in the Rules view.

Monitored Entities Reference

Every monitored entity has some associated metrics that you can use to author rules.

The following monitored entities you can use while creating rules:

- Cluster
- Processing Unit
- RTC Transactions

- Event Throughput
- BusinessEvents Rules

Cluster

The cluster has the metrics that are computed for the entire cluster or application.

Cluster Metrics

Metric	Description
Processing Units Running (%)	The percentage of processing units in the Running state of the total deployed processing units in the cluster.
Processing Units in Normal (%)	The percentage of processing units in the Normal state of the total deployed processing units in the cluster.
Processing Units in Warning (%)	The percentage of processing units in the Warning state of the total deployed processing units in the cluster.
Processing Units in Critical (%)	The percentage of processing units in the Critical state of the total deployed processing units in the cluster.

Processing Unit

The processing unit entity has the metrics computed for an instance.

Processing Units Metrics

Metric	Description
Average CPU (%)	Average CPU usage percentage averaged over five minutes.
Average Used Memory (%)	Average memory usage percentage (evaluated over max allocated memory), averaged over five minutes.
Is Running	Check if it is in a running or stopped state.
Thread Count	Count of the running threads grouped in five minutes.

Metric	Description
Deadlocked Thread Count	Count of the deadlocked threads grouped in five minutes.
Processing Unit	The associated processing unit name.
Processing Unit Name	The associated instance name.

RTC Transactions

The monitored entity has the metrics for the RTC transactions ran in the inference engine.

RTC Transactions Metrics

Metric	Description
Pending Locks	The count of pending locks to release groups in five minutes
RTC Transaction Count	The number of RTC Transactions grouped in five minutes
Processing Unit	The associated processing unit name
Processing Unit Name	The associated instance name
Agent Name	The name of the agent

Event Throughput

The monitored entity has the metrics for the throughput of the events ran in the inference engine.

Event Throughput Metrics

Metric	Description
Event Throughput	Total number of events asserted in the inference engine in a five-minute interval.

Metric	Description
Destination URI	The URI of the destination where the event was received
Processing Unit	The associated processing unit name
Processing Unit Name	The associated instance name
Agent Name	The name of the agent

BusinessEvents Rules

The monitored entity has the metrics for the BusinessEvents rules ran in the inference engine.

BusinessEvents Rules Metrics

Metric	Description
Rule Execution Time	The average rule execution time in milliseconds
Rule Name	The rule URI
Processing Unit	The associated processing unit name
Processing Unit Name	The associated instance name
Agent Name	The name of the agent

Alert Tokens Reference

While setting the alert texts, you can use certain tokens as placeholders for dynamic values. These placeholders are substituted with actual values that triggered the rule at run time.

Alert Tokens for Alert Texts

Alert Tokens	Description
${agent.name}	The name of the agent
${alert.priority}	The priority of the alert created by the associated action
${alert.timestamp}	The time at which the action was triggered
${alert.type}	The type of the condition (set or clear) which triggered the action
${application.name}	The application deployment name
${average.cpu.usage}	The average CPU usage percentage, averaged over five minutes
${average.memory.usage}	The average memory usage percentage that is evaluated over max allocated memory, and averaged over five minutes.
${berule.exec.time}	The average rule execution time in milliseconds
${berule.name}	The BusinessEvents rule URI
${deadlocked.threadcount}	The count of the deadlocked threads grouped in five minute
${destination}	The URI of the destination where the event was received
${entity.health}	The health of the rule entity at the set or clear action of the rule
${event.throughput}	The value of the event throughput in milliseconds.
${instance.name}	The associated instance name
${isrunning}	If the instance is in running or stopped state
${pending.locks}	The count of pending locks to release, grouped in five minutes.

Alert Tokens	Description
${percent.critical}	The percentage of processing units in the Critical state of the total deployed processing units in the cluster
${percent.normal}	The percentage of processing units in the Normal state of the total deployed processing units in the cluster
${percent.running}	The percentage of processing units in the Running state of the total deployed processing units in the cluster
${percent.warning}	The percentage of processing units in the Warning state of the total deployed processing units in the cluster
${processing.unit.name}	The associated processing unit name
${rule.owner.name}	The name of the rule owner
${total.threadcount}	The count of the running threads, grouped in five minutes
${transaction.rate}	The number of RTC transactions, grouped in five minutes

Deployment Profiles

In BusinessEvents Enterprise Administrator Agent, a deployed application manages multiple profiles. The deployment profile is used if you want to deploy an application in different environments using different set of properties. A deployment profile is a collection of global variables, system properties, and BusinessEvents properties. You can associate a deployment profile with an application. An application can have multiple deployment profiles based on the environment but only one is selected as the active profile.

The **Profiles** view displays all the profiles associated with the application. You can add a profile, or edit or delete an existing profile. You can select the active deployment profile for the application to start the agents with the settings configured in the selected profile. When an active profile is selected, the application instance status changes to **Needs Deployment**. At the time of instance deployment, all profile files are copied to the deployment location. When an instance is deployed, the `-p` parameter with `<active_profile>.properties` is added in the `<application_name>.sh` file of the application.

If a profile is selected, the settings on the configuration tab for global variables, system properties, and BusinessEvents properties are ignored. At runtime, the agent loads the settings from the configured profile.

Adding an Application Deployment Profile

You can add a new deployment profile that you can associate with the application and add global variables, system properties, and BusinessEvents properties for it.

Procedure

1. In the **Applications Deployment** tab, select the application that you want to deploy.
2. On the left panel, click **Profiles**.
3. In the **Profiles** view, click **Add Profile**.

   The Add Application Profile wizard is displayed.
4. In the Add Application Profile wizard, select the tab for the category of property, which you want to add, and click **Add New**.

   A new row is added under the selected property category.
5. Enter the **Name** and **Value** of the new property.

   When you click the **Name** field, you can also select from the existing properties of the application. These existing properties are displayed from the Processing Unit Instance Configuration page. For details about how to manage existing properties, see Processing Unit Instance Configuration.
6. Repeat *Step 2* to *Step 3*, till you have added all the required properties, and click **Save**.

   The **Save** button is active only if there is any change in the property value.

Editing an Application Deployment Profile

You can edit an application profile and update the global variables, system properties, and BusinessEvents properties.

Procedure

1. In the **Applications Deployment** tab, select the application that you want to deploy.

2. On the left panel, click **Profiles**.
3. In the **Profiles** view, click the **Edit** icon  for the profile which you want to edit.

   The Edit Application Profile wizard is displayed.
4. In the Edit Application Profile wizard, select the tab for the category of property, which you want to edit. Update the **Name** and **Value** of the properties as required. Click **Add New** if you want to add a new property.
5. Click **Save**.

   The **Save** button is active only if there is any change in the property value.

Deleting an Application Deployment Profile

You can delete an application profile if you no longer require that application profile.

Procedure

1. In the **Applications Deployment** tab, select the application that you want to deploy.
2. On the left panel, click **Profiles**.
3. In the **Profiles** view, click the **Delete** icon  for the profile that you want to delete.

   The Delete Application Profile confirmation dialog is displayed.
4. Click **Delete Application Profile** to delete the profile.

Creating a Duplicate Deployment Profile

If you want to create a deployment profile similar to an existing deployment profile, you can create a duplicate of an existing deployment profile. The new deployment profile contains the same variables as the existing deployment profile. You can then edit the newly deployment profile as per your requirement.

Before you begin

At least one deployment profile must be listed in the deployment profile list.

Procedure

1. In the **Applications Deployment** tab, select the application that you want to deploy.
2. On the left panel, click **Profiles**.
3. In the **Profiles** view, select the deployment profile from where you want to copy the variables and click **Copy Profile**.
4. On the Copy Profile window, enter the name of the new profile and click **Copy Profile**.

   A new deployment profile with the specified name is listed on the Profiles page.

What to do next

You can select the newly created deployment profile and edit it as per your requirement. See Editing an Application Deployment Profile for more details.

Using Python Deployment Scripts to Connect to the TIBCO Enterprise Administrator Server when SSL Is Enabled

You can use Python deployment scripts to connect to the TIBCO Enterprise Administrator Server when SSL is enabled for its connection with the TIBCO BusinessEvents Enterprise Administrator Agent.

Procedure

1. Create certificate files and update the keystore and truststore as follows:
   a. Create client and server keystore and truststore files in the JKS or JCEKS format.
   b. Convert keystore from the JKS or Cossacks format to the PKCS12 format.
   c. Convert the PKCS12 format to PEM files using OpenSSL with an encrypted PEM password.
2. Copy the certificate files to a specific folder. For example, `D:/tibco/tea/tea/certs/localhost`.

3. Update the TIBCO Enterprise Administrator server configuration file `tea.conf` located in the TEA installation configuration folder(for example, `C:/ProgramData/tea240hf1/tibco/cfgmgmt/tea/conf`) based on the type of SSL configuration:

   For one-way SSL:

   ```
   tea.http.keystore="D:/tibco/tea/tea/certs/localhost/httpserversslke
   ys.jceks"

   tea.http.truststore="D:/tibco/tea/tea/certs/localhost/httpserverssl
   trusts.jceks"
   tea.http.keystore-password=password
   tea.http.truststore-password=password
   tea.http.key-manager-password=password
   tea.http.cert-alias=httpserver
   tea.http.want.client.auth=false
   tea.http.need.client.auth=false
   ```

   For two-way SSL:

   ```
   tea.http.keystore =
   "D:/tibco/tea/tea241hf1/certs/localhost/httpserversslkeys.jceks"
   tea.http.truststore =
   "D:/tibco/tea/tea241hf1/certs/localhost/httpserverssltrusts.jceks"
   tea.http.keystore-password = "password"
   tea.http.truststore-password = "password"
   tea.http.key-manager-password = "password"
   tea.http.cert-alias = "httpserver"
   tea.http.want.client.auth = true
   tea.http.need.client.auth = true
   tea.http.client.keystore =
   "D:/tibco/tea/tea241hf1/certs/localhost/httpclientsslkeys.jceks"
   tea.http.client.truststore =
   "D:/tibco/tea/tea241hf1/certs/localhost/httpclientssltrusts.jceks"
   tea.http.client.keystore-password = "password"
   tea.http.client.truststore-password = "password"
   tea.http.client.key-manager-password = "password"
   tea.http.client.cert-alias = "httpclient"
   ```

   Note: For one-way SSL set the `tea.http.want.client.auth` and `tea.http.need.client.auth` properties to `false`.

4. Update the server URL the `be-teagent.props` file located at `BE_HOME/teagent/config`. For example, `be.tea.server.url=https://localhost:8777/tea` and add `be.tea.agent.host=localhost`.
5. Log in to TIBCO Enterprise Administrator server with the browser URL. For example `https://localhost:8777/tea`.
   If you have enabled two-way SSL, add the certificate files to the browser.
   For more information, see Signing in to the TIBCO Enterprise Administrator Server.
6. Run the Python deployment script. For example:

```
%BE_HOME%\teagent\cli\python>applicationsMgmt.py -ssl true -t
"https://localhost:8777" -u admin -p admin -sc
"D:\tibco\tea\tea240hf1\certs\localhost\server.pem"
createdeployment -d testPython -c
"D:/tibco/be/be611/be/6.1/examples/standard/FraudDetection/FraudDet
ection/fd.cdd" -e
"D:/tibco/be/be611/be/6.1/examples/standard/FraudDetection/fd.ear"
```

> **Note:** In one way SSL, you need to provide the server certificate only and in two way SSL, you need to provide both the server and client certificates.

User Management

You can configure user's roles and permission in the TIBCO Enterprise Administrator server.

The BusinessEvents Enterprise Administrator Agent uses the TIBCO Enterprise Administrator server framework for user management. The users for the BusinessEvents Enterprise Administrator Agent are configured in the TIBCO Enterprise Administrator server. An administrator can then assign users to different groups and roles. Each user role has some permissions enabled for it. The user with the assigned role can perform all the operations which are enabled for the associated role. See Roles and Permissions Reference for list of default roles and permissions available in BusinessEvents Enterprise Administrator Agent. Refer to the *TIBCO Enterprise Administrator User Guide* for adding users and managing roles.

Roles and Permissions Reference

You can configure user's roles and permission in the TIBCO Enterprise Administrator server.

By default, the following roles are configured in the BusinessEvents Enterprise Administrator Agent:

APP_MANAGER

The user with the APP_MANAGER role has all permissions in BusinessEvents Enterprise Administrator Agent.

DEPLOYER

The user with the DEPLOYER role has permissions required for processing unit instance deployment in BusinessEvents Enterprise Administrator Agent.

OPERATOR

The user with the OPERATOR role has permissions required for starting and stopping the processing unit instance in BusinessEvents Enterprise Administrator Agent.

RULE_AUTHOR

The user with the RULE_AUTHOR role has permissions to create non-health rules in BusinessEvents Enterprise Administrator Agent.

Non-health rules are those which do not alter the state of cluster health based on a rule.

RULE_AUTHOR_ADMIN

The user with RULE_AUTHOR_ADMIN role has permissions perform all rules and alerts operations in BusinessEvents Enterprise Administrator Agent.

VIEW_ALL

The user with the VIEW_ALL role has read-only permissions in BusinessEvents Enterprise Administrator Agent.

The following table lists all the default permissions and indicates which permissions are enabled for which role by default. If required, new roles and permission can be added in the TIBCO Enterprise Administrator by the administrator. Refer to the *TIBCO Enterprise Administrator User Guide* for adding roles and permissions.

Default Roles and Permissions of BusinessEvents Enterprise Administrator Agent

Permissions	APP_ MANAGER	DEPLOYER	OPERATOR	RULE_ AUTHOR	RULE_ AUTHOR_ ADMIN	VIEW_ ALL
`CREATE_ DEPLOYMENT_ PERMISSION` Create or import an application deployment	Yes	No	No	No	No	No
`UPDATE_ DEPLOYMENT_ PERMISSION` Edit or update an application deployment	Yes	No	No	No	No	No
`DELETE_ DEPLOYMENT_ PERMISSION` Delete an application deployment	Yes	No	No	No	No	No
`CREATE_HOST_ PERMISSION` Create a host	Yes	Yes	No	No	No	No
`UPDATE_HOST_ PERMISSION` Edit or update a host	Yes	Yes	No	No	No	No
`DELETE_HOST_ PERMISSION`	Yes	Yes	No	No	No	No

Permissions	APP_ MANAGER	DEPLOYER	OPERATOR	RULE_ AUTHOR	RULE_ AUTHOR_ ADMIN	VIEW_ ALL
Delete a host						
`CREATE_ INSTANCE_ PERMISSION` Create a processing unit instance	Yes	Yes	No	No	No	No
`UPDATE_ INSTANCE_ PERMISSION` Edit a processing unit instance	Yes	Yes	No	No	No	No
`DELETE_ INSTANCE_ PERMISSION` Delete a processing unit instance	Yes	Yes	No	No	No	No
`START_PU_ INSTANCE_ PERMISSION` Start a processing unit instance	Yes	Yes	Yes	No	No	No
`STOP_PU_ INSTANCE_ PERMISSION` Stop a processing unit	Yes	Yes	Yes	No	No	No

Permissions	APP_ MANAGER	DEPLOYER	OPERATOR	RULE_ AUTHOR	RULE_ AUTHOR_ ADMIN	VIEW_ ALL
instance						
KILL_ INSTANCE_ PERMISSION Kill a processing unit instance	Yes	Yes	Yes	No	No	No
HOT_DEPLOY_ PERMISSION Hot Deploy	Yes	Yes	No	No	No	No
COPY_ INSTANCE_ PERMISSION Copy a processing unit instance	Yes	Yes	No	No	No	No
UPDATE_GV_ VAR_ PERMISSION Update Global Variables	Yes	Yes	No	No	No	No
UPDATE_ SYSTEM_PROPS_ PERMISSION Update System Properties	Yes	Yes	No	No	No	No
UPDATE_JVM_ PROPS_ PERMISSION	Yes	Yes	No	No	No	No

Permissions	APP_ MANAGER	DEPLOYER	OPERATOR	RULE_ AUTHOR	RULE_ AUTHOR_ ADMIN	VIEW_ ALL
Update JVM Property						
UPDATE_LOG_ LEVEL_ PERMISSION Update Log Level	Yes	Yes	No	No	No	No
SUSPEND_ AGENT_ PERMISSION Suspend running BusinessEvents agent	Yes	Yes	Yes	No	No	No
RESUME_AGENT_ PERMISSION Resume suspended BusinessEvents agent	Yes	Yes	Yes	No	No	No
DEPLOY_ INSTANCE_ PERMISSION Deploy a processing unit instance	Yes	Yes	No	No	No	No
UNDEPLOY_ INSTANCE_ PERMISSION Un-deploy a	Yes	Yes	No	No	No	No

Permissions	APP_ MANAGER	DEPLOYER	OPERATOR	RULE_ AUTHOR	RULE_ AUTHOR_ ADMIN	VIEW_ ALL
processing unit instance						
UPLOAD_TRA_ PERMISSION Upload the TRA file	Yes	Yes	No	No	No	No
DEPLOY_ CLASSES_ PERMISSION Deploy the classes	Yes	Yes	No	No	No	No
UPLOAD_ CLASSES_ PERMISSION Upload the classes	Yes	Yes	No	No	No	No
DEPLOY_RULE_ TEMPLATE_ PERMISSION Deploy rule template instance	Yes	Yes	No	No	No	No
CREATE_RULE_ PERMISSION Create an alert rule	No	No	No	Yes	Yes	No
UPDATE_RULE_ PERMISSION	No	No	No	Yes	Yes	No

Permissions	APP_ MANAGER	DEPLOYER	OPERATOR	RULE_ AUTHOR	RULE_ AUTHOR_ ADMIN	VIEW_ ALL
Update an alert rule						
DELETE_RULE_ PERMISSION Delete an alert rule	No	No	No	Yes	Yes	No
GET_RULES_ PERMISSION View all alert rules	No	No	No	Yes	Yes	No
RULE_ADMIN_ PERMISSION Create health rules	No	No	No	No	Yes	No
CLEAR_ALERTS_ PERMISSION Clear alerts	No	No	No	Yes	Yes	No
GET_ALERTS_ PERMISSION View alerts	No	No	No	Yes	Yes	No
read Read permission	Yes	Yes	Yes	Yes	Yes	Yes
UPDATE_BE_ PROPS_ PERMISSION Update	Yes	Yes	No	No	No	No

Permissions	APP_ MANAGER	DEPLOYER	OPERATOR	RULE_ AUTHOR	RULE_ AUTHOR_ ADMIN	VIEW_ ALL
BusinessEvents property						
`ADD_PROFILE_ PERMISSION` Add a new deployment profile	Yes	Yes	No	No	No	No
`UPDATE_ PROFILE_ PERMISSION` Update the existing deployment profile	Yes	Yes	No	No	No	No
`DELETE_ PROFILE_ PERMISSION` Delete the existing deployment profile	Yes	Yes	No	No	No	No
`SET_DEFAULT_ PROFILE_ PERMISSION` Set the default deployment profile	Yes	Yes	No	No	No	No

BusinessEvents Enterprise Administrator Agent Configuration Reference

The BusinessEvents Enterprise Administrator Agent properties file provides you with various configuration properties for the agent.

The BusinessEvents Enterprise Administrator Agent properties file is located at *BE_HOME*\teagent\config\be-teagent.props.

⚠ **Caution:** Code snippets in the PDF can have undesired line breaks because of space constraints. Before directly copying and running them in your program, they must be verified.

General Configurations

General Configuration Properties

Property	Description
be.tea.agent.application.datastore	Specifies path to the directory that contains the TIBCO BusinessEvents Enterprise Administrator Agent configuration datastore The default value is *BE_HOME*/teagent/config/repo.
be.tea.agent.enable.gc.charts	Specifies whether to enable additional views (Garbage collector and Memory Pool chart) The default value is false.
be.tea.agent.jmx.port	Specifies the port for JMX connection. The default value is 5566.
be.tea.agent.jmx.usesingleport	*Optional*. Specifies whether single port is used for the JMX connection. The default value is true.
be.tea.agent.jmx.usessl	*Optional*. Specifies whether SSL is enabled for the JMX connection. The default value is false.
be.tea.agent.metric.compute.queue.size	*Optional*. Specifies the job queue size for the metric computation jobs thread pool.

Property	Description
	Ensure that you set the properties value twice the value of `be.tea.agent.metric.worker.thread.count`. The default value is 64.
`be.tea.agent.metric.compute.thread.count`	*Optional*. Specifies the number of threads to perform the metrics computation jobs. Do not reduce to less than the number of hierarchies as defined in the schema. The default value is 32.
`be.tea.agent.metrics.rules.attr.map.file`	Do not change. Specifies path to the XML file that contains rules attribute and entity mapping configurations The default value is *`BE_HOME`*`/teagent/config/RuleEntityAttrMap.xml`.
`be.tea.agent.metrics.view.config.file`	Do not change. Specifies path to the XML file that contains monitoring views configurations The default value is *`BE_HOME`*`/teagent/config/EntityMetricViewConfig.xml`.
`be.tea.agent.poller.delay`	*Optional*. Specifies the application instances status poller interval (in milliseconds) The default value is 30000.
`be.tea.agent.port`	Specifies TIBCO BusinessEvents Enterprise Administrator Agent listening port The default value is 9777.
`be.tea.agent.resource.base`	Specifies path to the directory that contains TIBCO BusinessEvents Enterprise Administrator Agent web resources The default value is *`BE_HOME`*`/teagent`.
`be.tea.agent.retry.interval`	*Optional*. Specifies the interval (in milliseconds) after which TIBCO BusinessEvents Enterprise Administrator Agent retries to register to TIBCO Enterprise Administrator server The default value is 5000.
`be.tea.agent.rta.config.plugin.dir`	Do not change. Specifies path to the XML file which contains monitoring metric collector plugins configurations The default value is *`BE_HOME`*`/teagent/config/plugins`.
`be.tea.agent.rules.actions.scan.frequency`	*Optional*. Specifies the interval (in milliseconds) before scanning for rule actions The default value is 5000.
`be.tea.agent.schema.store`	Do not change. Specifies path to the directory from where the server loads the schema files The default value is *`BE_HOME`*`/teagent/config/schema`.

Property	Description
`be.tea.agent.view.beentity.config.file`	Do not change. Specifies path to the XML file that contains BusinessEvents entity monitoring configurations The default value is *`BE_HOME`*`/teagent/config/BeEntityMap.xml`.
`be.tea.agent.worker.queue.size`	*Optional*. Specifies the job queue size of various thread pools. Keep it low, because unless the queue is full, no new threads are added to the worker thread pool. The default value is 4.
`be.tea.agent.worker.thread.count`	*Optional*. Specifies the maximum number of worker threads for thread pools The default value is 16.
`be.tea.agent.worker.thread.count.min`	*Optional*. Specifies the minimum number of threads to retain in the thread pools even at non-peak loads Keep it less than or equal to `be.tea.agent.worker.thread.count`. The default value is 1.
`be.tea.agent.worker.thread.idle.timeout`	*Optional*. Specifies the time interval (in seconds) after which idle threads of the thread pool are stopped, till the thread count reaches the value of `be.tea.agent.worker.thread.count.min` . The default value is 300.
`be.tea.server.url`	Specifies TIBCO Enterprise Administrator server URL The default value is http://localhost:8777/tea.
`be.tea.agent.host`	Registers BE TEAgent with hostname on TEAServer. `be.tea.agent.host=<host name>` at *`BE_HOME`*`\teagent\config\be-teagent.props`.

JMX Authentication Configurations

You can configure the JMX authentication properties to manage the JMX functionality and authentication.

JMX Authentication Properties

Property	Description
`be.tea.agent.jmx.enabled`	Enables or disables the JMX connection for TIBCO BusinessEvents Enterprise Administrator Agent. The valid values are true or false. The default value is true.

Property	Description
`be.tea.agent.jmx.authentication.type`	Specifies the type of authentication used. For the BusinessEvents Enterprise Administrator Agent JMX connections, only file-based authentication is supported. The default value is file.
`be.tea.agent.jmx.authentication.enabled`	Enables or disables the JMX authentication TIBCO BusinessEvents Enterprise Administrator Agent. The valid values are true or false. The default value is false.
`be.tea.agent.jmx.auth.file.location`	Specifies the absolute file path and name of the password file. This file is used for file-based authentication and stores users:passwords:roles details. Each project can have a different file.
`be.tea.agent.login.config.file.location`	Specifies the absolute filepath to the JAAS file.
`be.tea.agent.jmx.username`	Specifies the username for JMX authentication. This property is required for TIBCO BusinessEvents Enterprise Administrator Agent monitoring APIs when JMX authentication is enabled.
`be.tea.agent.jmx.password`	Specifies the password for JMX authentication. This property is required for TIBCO BusinessEvents Enterprise Administrator Agent monitoring APIs when JMX authentication is enabled. **Note:** You can use an encrypted JMX Password. You can encrypt the password using `studio-tools` encrypt utility.

Log Action Configurations

Log Action Configuration Properties

Property	Description
`be.tea.agent.log.alert.format`	Specifies the format (XML or TEXT) in which the alert text is logged for the rule. The default value is TEXT
`be.tea.agent.log.level`	Specifies the logging level for TIBCO BusinessEvents Enterprise Administrator Agent.

Property	Description
	The following log levels are available: • TRACE • DEBUG • INFO • WARN • ERROR • FATAL The default value is INFO.

Email Action Configurations

You can configure the SMTP server details in the properties file to execute the email action in the alert rule. See Creating an Alert Rule for more details on creating an alert rule.

Email Action Configuration Properties

Property	Description
`be.tea.agent.mail.from`	Specifies the email address to be used for sending the notification email
`be.tea.agent.mail.retry.count`	Optional. Specifies the number of times the server tries to send email The default value is 3.
`be.tea.agent.mail.retry.interval`	Optional. Specifies the interval (in milliseconds) before retrying to send the email The default value is 2000.
`be.tea.agent.mail.smtp.authentication`	Optional. Specifies whether the authentication is checked or not for the email server The default value is false.
`be.tea.agent.mail.smtp.host`	Specifies the host name of the SMTP email server
`be.tea.agent.mail.smtp.password`	Specifies the sender's email password for authentication to email server
`be.tea.agent.mail.smtp.port`	Optional. Specifies the port at which the SMTP email server is listening The default value is 25.

Property	Description
be.tea.agent.mail.smtp.user	Specifies the sender's username for authentication to email server

SSH Configurations

You can provide the SSH connection-related configuration.

SSH Configuration Properties

Property	Description
be.tea.agent.ssh.connection.timeout	Specifies the SSH connection timeout (in milliseconds) The default value is 30000.
be.tea.agent.ssh.kex.algorithm	Specify the key exchange algorithm for password-less authentication. The values are: • diffie-hellman-group1-sha1 • diffie-hellman-group14-sha1 • diffie-hellman-group-exchange-sha1 • diffie-hellman-group-exchange-sha256
be.tea.agent.ssh.privatekey.file	Specifies the path to the private key file for password-less SSH authentication
be.tea.agent.ssh.privatekey.passphrase	Specifies passphrase to the private key file (if required) The default value is 3.

Localization Configurations

You can specify configurations related to your local language.

Localization Configuration Properties

Property	Description
be.tea.agent.message.file	Specifies the file path to the BusinessEvents Enterprise Administrator agent messages file. See Localizing BusinessEvents Enterprise Administrator Agent Messages for details on how to use this property.

Persistence Configurations

You can specify properties to persist the TIBCO BusinessEvents Enterprise Administrator Agent data in a non-heap memory or on a disk.

Note: Ensure to import the schema for a database you are using from the location *BE_HOME*\teagent\bin.

The schemas for Postgres and Oracle databases are included with TIBCO BusinessEvents.

Persistence Configuration Properties

Property	Description
tea.agent.jdbc.driver	Specifies driver name for a database
tea.agent.jdbc.url	Specifies connection URL for a database
tea.agent.jdbc.user	Specifies username to access a database
tea.agent.jdbc.password	Specifies password to access a database
be.tea.agent.persistence.provider	Set to database to persist TIBCO BusinessEvents Enterprise Administrator Agent data The default value is memory.

i18n Support

You can configure TIBCO BusinessEvents Enterprise Administrator Agent to support multibyte characters for localization.

Using multibyte characters you can provide application, instance, and machine name in international lingual characters. To enable the multibyte character support in TIBCO BusinessEvents Enterprise Administrator Agent:

- Ensure that the `java.property.file.encoding=UTF-8` property is present in the `be-teagent.tra` file.
- Also, add the same property (`java.property.file.encoding=UTF-8`) in the *`TEA_HOME`*`/bin/tea.tra` file.

Localizing BusinessEvents Enterprise Administrator Agent Messages

You can localize the log messages recorded in the `be-teagent.log` in your language by using the message files. These message files are located at *`BE_HOME`*`/teagent/config/messages/` folder.
The message file contains BusinessEvents Enterprise Administrator agent success and error messages in the supported language. Messages are specified for each identifier separated by equal sign (=).

The following table lists the supported languages and provided respective message file.

Supported Language	Message File at *`BE_HOME`*`/teagent/config/messages/`
Arabic	`messages_ar.properties`
English	`messages_en_US.properties`
French	`messages_fr.properties`

Supported Language	Message File at *BE_HOME*/teagent/config/messages/
German	messages_de.properties
Italian	messages_it.properties
Korean	messages_ko.properties
Simplified Chinese	messages_zh_CN.properties
Spanish	messages_es.properties

Procedure

1. Open *BE_HOME*\teagent\config\be-teagent.props file for editing.
2. Update the be.tea.agent.message.file property with the path of the message file of your language.

   For example, for German translation, the value of the property is:

   ```
   be.tea.agent.message.file=C:/tibco/be/
   <version>/teagent/config/messages/messages_de.properties
   ```

3. Save the be-teagent.props file and restart the engine.

Command-line Interface

Using BusinessEvents Enterprise Administrator Agent, you can perform most of the application and configuration management operations from the Python-based command-line interface.

To successfully execute the command-line interface operations, ensure that:

- TIBCO Enterprise Administrator server is running.
- TIBCO BusinessEvents Enterprise Administrator Agent is running.
- TIBCO BusinessEvents Enterprise Administrator Agent is registered with the TIBCO Enterprise Administrator server.

- Python is installed and the path is set.
- The python `jsonpickle` module is installed.
- The value of the *PYTHONPATH* environment variable is set to *BE_HOME*`\teagent\cli\python`.

The python script takes the following command arguments:

- (Optional)Secured connection (true or false)
    - Server SSL certificate path
    - Client SSL certificate path
- TIBCO Enterprise Administrator server URL
- TIBCO Enterprise Administrator server username
- TIBCO Enterprise Administrator user password
- Operation name
- Operation arguments

TIBCO BusinessEvents Enterprise Administrator Agent Commands Reference

You can run the configuration and application management commands for BusinessEvents Enterprise Administrator Agent in the Python-based command-line interface. The python script files are at *BE_HOME*`\teagent\cli\python`.

- Configuration Management Commands (configurationMgmt.py)
- Application Management Commands (applicationsMgmt.py)
- Administrator to BusinessEvents Enterprise Administrator Agent Migration Commands (adminToAgentMigration.py)
- Arguments for Commands

Configuration Management Commands (`configurationMgmt.py`)

Without SSL

The syntax for running the `configurationMgmt.py` python script file is:

```
python configurationMgmt.py -t SERVERURL -u USERNAME -p USERPWD -gt
GTIMEOUT commandnamecommandparameters
```

With SSL

The syntax for running the `configurationMgmt.py` python script file with SSL is:

```
python configurationMgmt.py -ssl SSLENABLED -t SERVERURL -u USERNAME -p
USERPWD -sc SERVERCERT -cc CLIENTCERT -gt GTIMEOUT
commandnamecommandparameters
```

editMachine

Command to edit the machine details:

```
editmachine [-h] -m MACHINENAME [-n NEWMACHINENAME] [-i IPADDRESS]
[-o {windows,unix,os-x}] [-b BEHOME] [-t BETRA] [-u USER] [-p PWD] [-
s=
SSHPORT] [-f DEPLOYMENTPATH]
```

deleteMachine

Command to delete the machine:

```
deletemachine [-h] -m [MACHINENAMES ...]
```

discoverBEHomes

Command to discover BusinessEvents installation on the machine:

```
discoverbehomes [-h] -m MACHINENAME [-s=SAVE]
```

uploadExternalJars

Command to upload an external JAR file:

```
uploadexternaljars [-h] -m MACHINENAME -b BEHOME -z JARFILES
```

editDeployment

Command to edit an application deployment:

```
editdeployment [-h] -d APPLICATIONNAME [-c CDDFILE] [-e EARFILE]
```

editInstance

Command to edit an application instance:

```
editinstance [-h] -d APPLICATIONNAME -i INSTANCENAME [-u PU] [-m
MACHINENAME] [-p JMXPORT] [-f DEPLOYMENTPATH] [-ju JMXUSER] [-jp
JMXPASS] [-bh BEHOME]
```

saveGlobalVariable

Command to update a global variable for one or multiple instances. Only one variable can be updated at a time. Some variables cannot be updated.

```
saveglobalvariable [-h] -d APPLICATIONNAME [-i [INSTANCES ...]] -n
VARNAME -v VARVALUE
```

saveSystemProperty

Command to update or add a system property for one or multiple instances:

```
savesystemproperty [-h] -d APPLICATIONNAME [-i [INSTANCES ...]] -n
PROPNAME -v PROPVALUE
```

saveJVMProperty

Command to update JVM property for one or multiple instances:

```
savejvmproperty [-h] -d APPLICATIONNAME [-i [INSTANCES ...]] -n
PROPNAME -v PROPVALUE
```

saveBEProperty

Command to update BusinessEvents property for one or multiple instances:

```
savebeproperty [-h] -d APPLICATIONNAME [-i [INSTANCES ...]] -n
PROPNAME -v PROPVALUE
```

saveDeploymentVariables

Command to bulk upload deployment variables:

```
savedeploymentvariables [-h] -d APPLICATIONNAME [-i [INSTANCES ...]] -
tp TYPE -p PROPFILE
```

Application Management Commands (`applicationsMgmt.py`)

Without SSL

The syntax for running the `applicationsMgmt.py` python script file is:

```
python applicationsMgmt.py -t SERVERURL -u USERNAME -p USERPWD -gt
GTIMEOUT commandnamecommandparameters
```

With SSL

The syntax for running the `applicationsMgmt.py` python script file with SSL is:

```
python applicationsMgmt.py -ssl SSLENABLED -t SERVERURL -u USERNAME -p
USERPWD -sc SERVERCERT -cc CLIENTCERT -gt GTIMEOUT
commandnamecommandparameters
```

addMachine

Command to add a machine:

```
addmachine [-h] -m MACHINENAME -i IPADDRESS -o {windows,unix,os-x} -b
BEHOME -t BETRA -v BEVERSION -u USER -p PWD -s=SSHPORT -f
DEPLOYMENTPATH [-abh ADDBEHOME]
```

createDeployment

Command to create a new application using the specified CDD and EAR files:

```
createdeployment [-h] -d APPLICATIONNAME -c CDDFILE -e EARFILE
```

deleteDeployment

Command to delete the application:

```
deletedeployment [-h] -d APPLICATIONNAME
```

createInstance

Command to create an instance of an application. The JMX username and password is governed by the policy in the CDD file:

```
createinstance [-h] -d APPLICATIONNAME -i INSTANCENAME -u PU -m
MACHINENAME -p JMXPORT [-f DEPLOYMENTPATH] [-ju JMXUSER] [-jp JMXPASS]
[-bh BEHOME]
```

copyInstance

Command to copy an existing instance of an application. The JMX username and password is governed by the policy in the CDD file:

```
copyinstance [-h] -d APPLICATIONNAME -i INSTANCENAME -n
NEWINSTANCENAME -u PU -m MACHINENAME -p JMXPORT -f DEPLOYMENTPATH [-ju
JMXUSER] [-jp JMXPASS] [-bh BEHOME]
```

deleteInstance

Command to delete an instance:

```
deleteinstance [-h] -d APPLICATIONNAME [-i [INSTANCES ...]]
```

deploy

Command to deploy application instances based on the specified machine, processing unit, or agent class:

```
deploy [-h] -d APPLICATIONNAME [-m MACHINE | -u PU | -a AGENTCLASS] [-
i [INSTANCES ...]]
```

undeploy

Command to undeploy application instances based on the specified machine, processing unit, or agent class:

```
undeploy [-h] -d APPLICATIONNAME [-m MACHINE | -u PU | -a AGENTCLASS]
[-i [INSTANCES ...]]
```

start

Command to start application instances based on the specified machine, processing unit, or agent class:

```
start [-h] -d APPLICATIONNAME [-m MACHINE | -u PU | -a AGENTCLASS] [-i
[INSTANCES [INSTANCES ...]]]
```

stop

Command to stop application instances based on the specified machine, processing unit, or agent class:

```
stop [-h] -d APPLICATIONNAME [-m MACHINE | -u PU | -a AGENTCLASS] [-i
[INSTANCES ...]]
```

hotdeploy

Command to hotdeploy an application provided by the EAR file:

```
hotdeploy [-h] -d APPLICATIONNAME -e EARFILE
```

downloadLogs

Command to download logs for multiple instances:

```
downloadlogs [-h] -d APPLICATIONNAME -l DOWNLOADLOCATION -lt
{be,as,td} [-i [INSTANCES ...]]
```

exportTeaDeployment

Command to export the application and all its components such as, CDD file, EAR file, and configuration XML file in an archive file:

```
exportteadeployment [-h] -d APPLICATIONNAME -l DOWNLOADLOCATION
```

importTeaDeployment

Command to import an application deployment:

```
importteadeployment [-h] -z ZIPFILE
```

hotDeployDtRt

Command to hot deploy decision table and business rules:

```
hotdeploydtrt [-h] -d APPLICATIONNAME -i INSTANCENAME -o {dt,rt} -z
ZIPFILE
```

checkStatus

Command to check the status of a machine or an application:

```
checkstatus [-h] -d APPLICATIONNAME -m MACHINE [-i INSTANCENAME] [-an
AGENTNAME] [-da DISPLAYALL]
```

docker_createdeployment

Command to create a Docker deployment:

```
docker_createdeployment [-h] -d APPLICATIONNAME
```

docker_addmachine

Command to add a Docker machine:

```
docker_addmachine [-h] -m MACHINENAME -i IPADDRESS [-o
{windows,unix,os-x}] [-u USER] [-p PWD] [-s SSHPORT]
```

docker_createinstance

Command to create a Docker instance:

```
docker_createinstance [-h] -d APPLICATIONNAME -i INSTANCENAME -m
MACHINENAME -p JMXPORT [-ju JMXUSER] [-jp JMXPASS] [-u PU]
```

listappsandinstances

Command to list all applications and all instances of an application:

```
listappsandinstances [-h] [-d APPLICATIONNAME]
```

uploadmastertra

Command to provide a Master Application TRA file at the application level:

```
uploadmastertra [-h] -d APPLICATIONNAME -m MACHINENAME [-ul
UPLOADLOCATION] [-tf TRAFILE] [-tp TRAPATH] [-td ISDELETED]
```

Administrator to BusinessEvents Enterprise Administrator Agent Migration Commands (`adminToAgentMigration.py`)

Without SSL

The syntax for running the `adminToAgentMigration.py` python script file is:

```
python adminToAgentMigration.py -t SERVERURL -u USERNAME -p USERPWD
commandnamecommandparameters
```

With SSL

The syntax for running the `adminToAgentMigration.py` python script file with SSL is:

```
python adminToAgentMigration.py -ssl SSLENABLED -t SERVERURL -u USERNAME
-p USERPWD -sc SERVERCERT -cc CLIENTCERT
{migrateapplications,migrateuserandroles} ...commandnamecommandparameters
```

migrateapplications

Command to migrate an application that is exported from TIBCO Administrator to TIBCO BusinessEvents Enterprise Administrator agent. See Exporting Application from TIBCO Administrator for more details.

```
migrateapplications -z EXPORTEDZIPFILE
```

migrateuserandroles

Command to migrate user and role from the application that is exported from TIBCO Administrator to TIBCO BusinessEvents Enterprise Administrator agent:

```
migrateuserandroles -x EXPORTEDXMLFILE
```

Arguments for Commands

The following table lists the arguments provided to the BusinessEvents Enterprise Administrator Agent commands and their description. Any arguments with spaces are enclosed in double quotes, for example, `-o "OX/X,Unix/Linux Based"`.

Parameters	Description
`[]`	Identifies an optional parameter
`[-h]`	Help to display parameters for the current command
`-ssl SSLENABLED`	(Optional) Enable SSL for the python script execution. The values are: • true • false The default value is `false`.
`-sc SERVERCERT`	Path of the server SSL certificate.
`-cc CLIENTCERT`	Path of the client SSL certificate.
`-t SERVERURL`	The TIBCO Enterprise Administrator server URL

Parameters	Description
	If no value is provided, then the default value (http://localhost:8777) is used.
`-u USERNAME`	TIBCO Enterprise Administrator server username
`-p USERPWD`	TIBCO Enterprise Administrator server password
`-gt GTIMEOUT`	Optional parameter to set a timeout value while running the python script files. The default value is 30 seconds. **Note:** The maximum timeout that you can set is 120 seconds. To set a timeout greater than that, change the default value of a `tea.agents.request-timeout` property in the TIBCO Enterprise Administrator server configuration file (`tea.conf`).
`-a AGENTCLASS`	Agent class
`-an AGENTNAME`	Agent name
`-da DISPLAYALL`	Displays the status of all instances and agents of a machine or application
`-m MACHINENAME`	The current machine name
`-n NEWMACHINENAME`	The new machine name after changing the current machine name
`-i IPADDRESS`	The IP address of the machine
`-o {windows,unix,os-x}`	Type of the operating system of the machine
`-b BEHOME`	The location where TIBCO BusinessEvents is installed
`-t BETRA`	The location of the be-engine.tra file
`-u USER`	The username for the machine.

Parameters	Description
`-p PWD`	The password for the username of the machine
`-s SSHPORT`	The SSH port number
`-f DEPLOYMENTPATH`	The deployment path location
`-d APPLICATIONNAME`	The application name
`-c CDDFILE`	The location of the CDD file of the application
`-e EARFILE`	The location of the EAR file of the application
`-z JARFILES`	The location of the JAR file of the application
`-z ZIPFILE`	The location of the ZIP file of the application
`-i INSTANCENAME`	The current application instance name
`[-i [INSTANCES ...]]`	Identifies multiple instance names. For example, `-i instance1 ins2 ins3`.
`-n NEWINSTANCENAME`	The new application instance name after changing the current application instance name
`-u PU`	Processing unit name
`-p JMXPORT`	JMX port number
`-ju JMXUSER`	JMX username
`-jp JMXPASS`	JMX password
`-n VARNAME`	Global variable name.
`-v VARVALUE`	Global variable value
`-tp VARTYPE`	Type of variable that needs to be updated.

Parameters	Description
	Possible values are as follows: • GV • BE • JVM • SYS
`-tp TYPE`	Type of deployment variables to be uploaded in bulk
`-p PROPFILE`	Property file that contains the deployment variables
`-n PROPNAME`	System, BusinessEvents, or JVM property name
`-v PROPVALUE`	System, BusinessEvents, or JVM property value
`-p PROPERTY_FILE_PATH`	The location of the property file that contains the deployment variables. **Note:** One type of property file must contain only one type of deployment variables.
`-x EXPORTEDXMLFILE`	The location of the XML file of the application exported from the TIBCO Administrator.
`-z EXPORTEDZIPFILE`	The location of an external archive file (ZIP or JAR).
`-bh BEHOME`	TIBCO BusinessEvents installation location
`-abh ADDBEHOME`	Additional TIBCO BusinessEvents installation location if multiple TIBCO BusinessEvents installations are present.
`-s SAVE`	Save the location of BusinessEvents installation for the machine.
`-o {dt,rt} DEPLOYTYPE`	Type of artifact to be hot deployed. The values are: • `DT` - Decision table

Parameters	Description
	• RT - Business rules (rule template instance)
-l DOWNLOADLOCATION	Download location of log files.
-lt {be,as,td} LOGTYPE	Type of the logs to be downloaded. The values are: • BE - BusinessEvents logs • AS - Legacy ActiveSpaces logs • TD - Thread dumps
-m MACHINE	The current machine
-v BEVERSION	Version of the current TIBCO BusinessEvents installation
-ul UPLOADLOCATION	The location to upload the master TRA file
-tf TRAFILE	The master TRA file
-tp TRAPATH	The location of the master TRA file on the host machine
-td ISDELETED	Specifies whether to delete the TRA file The values are: • true • false The default value is false.

Authentication and SSL Configurations

The TIBCO Enterprise Administrator supports both one-way (server side) and two-way (server side as well as client side) SSL authentication. You can configure SSL between the Web browser and the TIBCO Enterprise Administrator as well as between the TIBCO Enterprise Administrator and the agent.

SSL Configuration for Web Browser and TIBCO Enterprise Administrator Server Connection

Refer to the *TIBCO Enterprise Administrator User Guide* for more details on configuring SSL authentication between the web browser and TIBCO Enterprise Administrator server.

SSL Configuration for TIBCO Enterprise Administrator Server and BusinessEvents Enterprise Administrator Agent Connection

Refer to the *TIBCO Enterprise Administrator User Guide* for more details on configuring SSL authentication between the TIBCO Enterprise Administrator server and TIBCO BusinessEvents Enterprise Administrator Agent.

Add the BusinessEvents Enterprise Administrator Agent side properties to the `be-teagent.tra` prefixed with `java.property`.

SSL Configuration for BusinessEvents Enterprise Administrator Agent and Processing Unit Instance Connection

See Configuring One-way SSL between Administrator Agent and Processing Unit Instance for more details.

Configuring JMX Authentication

You can activate user for JMX connection between BusinessEvents Enterprise Administrator Agent and BusinessEvents processing unit instances.

Procedure

1. Add the following **System Properties** for each processing unit instance:
   - `be.engine.jmx.connector.authenticate=true`
   - `be.auth.type=file|ldap`
   - `be.auth.file.location=<location of file that stores the users/passwords/roles>`

   See Adding a New System Property for an Instance for more details.
2. Perform additional configuration, see User Authentication for more details.

What to do next

In the processing unit instance configuration page, specify the JMX username and password for each instance. See Creating an Instance and Updating an Instance for more details.

Configuring One-way SSL between Administrator Agent and Processing Unit Instance

To enable one-way SSL authentication, configure SSL properties in the BusinessEvents Enterprise Administrator Agent as well as BusinessEvents processing unit instances.

Procedure

BusinessEvents Enterprise Administrator Agent side SSL configuration

1. Add the following properties to the `be-teagent.tra` file:
   - `java.property.javax.net.ssl.trustStore=`*<location of the truststore file>*
   - `java.property.javax.net.ssl.trustStorePassword=`*<password of the truststore file>*

> **Note:** Ensure that all BusinessEvents instances public certificates are stored in a single trust store.
>
> To do this, you can also use the keytool utility as follows:
>
> ```
> keytool -import -alias pu1 -file <pu1 certificate> -keystore <path to mytruststore>
> keytool -import -alias pu2 -file <pu2 certificate> -keystore <path to mytruststore>
> ```
>
> Where, *pu1* and *pu2* are two BusinessEvents processing unit instances.

BusinessEvents processing unit instance side configuration

2. Add the following **System Properties** for each processing unit instance using the BusinessEvents Enterprise Administrator Agent user interface:
   - `be.engine.jmx.connector.ssl=true`
   - `javax.net.ssl.keyStore=`*<location of the keystore file>*
   - `javax.net.ssl.keyStorePassword=`*<password of the keystore file>*

   See Adding a New System Property for an Instance for more details.

Enabling SSL for The BusinessEvents Enterprise Administrator Agent Monitoring Page

If you want to securely access the TIBCO BusinessEvents Enterprise Administrator Agent monitoring page, then you can enable SSL for it.

Procedure

1. Open the *BE_HOME*`\teagent\bin\be-teagent.tra` file for editing.
2. Add the following properties and their values in the `be-teagent.tra` file, and save the file.
   - `java.property.javax.net.ssl.keyStore=`*<location of the keystore file>*
   - `java.property.javax.net.ssl.keyStoreType =`*<type of keystore>*
   - `java.property.javax.net.ssl.keyStorePassword=`*<password of the keystore file>*
   - `java.property.javax.net.ssl.trustStore=`*<location of the truststore file>*
   - `java.property.javax.net.ssl.trustStoreType=`*<type of truststore>*
   - `java.property.javax.net.ssl.trustStorePassword=`*<password of the truststore file>*
3. Open the *BE_HOME*`\teagent\config\be-teagent.props` file for editing.
4. Add the property `be.tea.agent.jmx.usessl=true` in the `be-teagent.props` file, and save the file.

Enterprise Archive (EAR) Files

You can build an enterprise archive file using a TIBCO BusinessEvents Studio dialog, and also using a command-line utility.

Certain files (and folder names) are excluded from the EAR. To maintain the list of exclusions, in TIBCO BusinessEvents Studio, select **Window > Preferences > TIBCO BusinessEvents > Code Generation > Ignored Resources**.

Note: Do not store the EAR file in a project folder, because this will include the previous EAR file when you build the EAR file again, needlessly increasing the size.

EAR File Encoding

- The default encoding of the EAR files generated by TIBCO BusinessEvents Studio is `ISO8859-1`. This is also the default encoding of TIBCO Administrator. To upload an EAR file to TIBCO Administrator, the EAR file encoding must match the TIBCO Administrator encoding.
- To change the default EAR file encoding, define a global variable named *MessageEncoding* and set its value to the desired encoding. For example, *UTF-8*.

EAR Files and the Studio Tools Utility

You can build an EAR file with the Studio Tools command-line utility. The `buildear` operation within the `studio-tools` utility is useful for automation purposes, for example, in testing environments.

By default, the EAR files are built in memory. The compiler does not use the file system during code generation. Studio generates, compiles, and packages all resources in memory until the build process is completed.

Before you build an EAR file during hot deployment of the new concept and concept properties, make sure to add the following property in the `studio-tools.tra` file:

```
java.property.com.tibco.be.hotdeploy.concept=true
```

Building an EAR File in TIBCO BusinessEvents Studio

EAR files are built in memory by default. The compiler does not use the file system during code generation. Studio generates, compiles, and packages all resources in memory until the build process is completed.

Procedure

1. In TIBCO BusinessEvents Studio Explorer, highlight the project name, then from the top menu select **Project > Build Enterprise Archive**.

   If you see a message asking you to save all project resources, click **Yes**. (This message means that an unsaved resource editor is open.)

2. At the Build Enterprise Archive dialog, complete the values according to the guidelines provided in Enterprise Archive Reference.

   > **Note:** When building an EAR file in memory for a large project, the JVM may run out of `PermGenSpace` and/or heap space. In such cases, edit the *BE-HOME*`/studio/eclipse/studio.ini` and *BE-HOME*`/studio/bin/studio-tools.tra` file to set appropriate values for the JVM settings. By default the heap size is set to `-XX:MaxPermSize=256m`.

3. Click **Apply** to save the configuration details.

   To revert to the version already saved, click **Revert**.

4. Click **OK** to build the archive.

Enterprise Archive Reference

This reference is used to build the Enterprise Archive (EAR) file.

Enterprise Archive Reference

Field	Description
Name	Name of this EAR configuration. (Not the EAR filename.)

Field	Description
	Default value is the project name. **Note:** For deployment using TIBCO Administrator, the configuration Name field value must match the project name. The project does not deploy if they are different. The actual EAR file name, however, can differ from the configuration name.
Author	Person responsible for the EAR file. Default value is the currently logged-on username.
Description	Optional description.
Archive Version	Increments on each build of the EAR. You can also manually enter a version identifier.
Generate Debug Info	Select this checkbox to use the debugger. Default setting is `checked`.
Include all service level global variables	Select to include service level global variables.
Include generated source code in archive	Select to include source code files in archive file. By default the generated source is not included in the EAR file.
Clean project before build	Select to remove any existing cached project information. This operation is similar to running **Project > Clean** for the project.
File Location	Browse to the directory in which you want to store the EAR file and enter an EAR filename.

Building an EAR File at the Command Line

You can build an EAR file using the command-line interface.

Procedure

1. Navigate to *BE_HOME*/studio/bin/ and open a command prompt.
2. Execute a command with the following format (all on one line) at a command prompt:

   ```
   studio-tools -core buildEar [-h] [-s] [-x] [-jc] [-v] [-noBuild] [-
   showWarnings] [-o <outputArchiveFile>] -p <projectDir> -pl<project
   lib path><path separator><project lib path> -cp<extended classpath>
   -clean
   ```

   For example:

   ```
   studio-tools -core buildEar -o c:\FD.ear -p
   D:\Workspace\FraudDetection
   ```

   See Options for Building an EAR File.

   Note: When building an EAR file in memory for a large project, the JVM may run out of PermGenSpace and/or heap space. In such cases, edit the *BE_HOME*/studio/eclipse/studio.ini and *BE_HOME*/studio/bin/studio-tools.tra file to set appropriate values for the JVM settings. By default the heap size is set to -XX:MaxPermSize=256m.

3. When testing a project, run it at the command line using the following format:

   BE_HOME/be-engine [-h] [--propFile *startup property file*] [--propVar *varName=value*] [-p *custom property file*] [-n *engine name*] [-d] [-c *CDD file*] [-u *processing unit ID*] [*EAR file*]

Options for Building an EAR File

These options are used to build an EAR file on the command line.

TIBCO BusinessEvents Studio Tools Options for Building an EAR File

Option	Description
`-core buildEar`	Specifies the `buildear` operation for building EAR files.
`-h`	(Optional) Displays help.
`-x`	(Optional) Overwrites the specified output file if it exists.
`-s`	(Optional) Include the generated source code in EAR file.
`-jc`	(Optional) Includes the JAR files specified in the Java classpath while building the EAR file.
`-o`	(Optional) Specifies the filename for the output EAR file. If not specified the EAR file is the same as the final (leaf) directory name in the *`projectDir`* path.
`-p`	Absolute path to the TIBCO BusinessEvents Studio project directory. The EAR file is built using this project.
`-pl`	(Optional) Specifies list of project library file paths to be used, separated by a path separator.
`-cp`	(Optional) Specifies the extended classpath to be used.
`-v`	(Optional) Performs validation checks to ensure that all the resource requirements are met. For example, it checks if the required fields are completed, names are valid, the syntax in rules is correct and no unknown functions are called. For more information, see *Validating Project or Project Resource* in *Developers Guide*.
`-showWarnings`	(Optional) Displays warnings found during the validation process. **Default:** False
`-noBuild`	(Optional) Use this option along with the validate option (`-v`) to validate a project without building an EAR for it.

Option	Description
`-clean`	(Optional) Use this option to remove any existing cached project information.

Engine Management at the Command Line

When testing a project, run it at the command line.

To run the TIBCO BusinessEvents engine at the command line, use this command:

```
BE_HOME/be-engine [-h] [--propFile startup property file] [--propVar
varName=value][-p custom property file] [-n engine name] [-d] [-c CDD
file] [-u processing unit ID]  [EAR file]
```

For options to use, see Command Line Startup Option Reference.

Command Line Startup Option Reference

Engine startup options used for testing.

Command Line Startup Options

Option	Description
`-h`	Displays this help.
`--propFile`	When you execute `be-engine`, it searches for a property file of the same name in the working directory. This property file provides startup values and other parameters to the executable. You can specify the path and filename of a startup property file explicitly using the `--propFile` parameter. For example, if you start the engine from a directory other than *BE_HOME*`/bin`, then you would generally use `--propFile` to specify *BE_HOME*`/bin/be-engine.tra`.
`--propVar`	Used to provide a value for a specified variable. This value overrides any other design-time value. The format is `propVar-`*varName*`=value` . For example to specify the value of the *%jmx_port%* variable used in TRA files

Option	Description
	to configure a JMX connection, you might use this: `--propVar jmx_port=4567`.
`-p`	Allows you to pass one or more supplementary property files to `be-engine`. Specify the path and filename. This is not needed very often. See Supplementary Property Files .
`-n`	Allows you to provide a name for the TIBCO BusinessEvents engine. The name provided here is used in the console and in log files. If you do not provide a name, the host name of the machine is used.
`-d`	Starts the debugger service on the engine for remote debugging.
`-c`	Specify the path and filename for the Cluster Deployment Descriptor (CDD) file. TIBCO BusinessEvents looks first in the file system, and then in the EAR file. The default is `default.cdd`.
`-u`	Specify the processing unit ID you want to use for this engine. This ID must exist in the CDD file you reference in the `-c` option. The default is `default`.
EAR filename and path	Specify the path and filename for the EAR file you want to use. If you do not specify the EAR file name then the engine will use the property `tibco.repourl` as the EAR file path and name. To use this property, add it to the `be-engine.tra` file. If you deploy using TIBCO Administrator this property is added to the generated TRA file automatically.

Supplementary Property Files

Supplementary property files can be used in addition to `be-engine.tra` (or the alternate file you specified using `--propFile`).

In TIBCO BusinessEvents 4.0 and later versions, property files are less likely to be needed, because only system level configuration is done in the TRA files. Configuration formerly done in TRA files is now done using the Cluster Deployment Descriptor file. Supplementary property files typically have a `.cfg` or `.tra` extension. Properties are defined as a list of name-value pairs. If a property name includes spaces, escape them using a back slash (\).

Order of Precedence

Values in supplementary property files override the values in the startup property file. Values provided at the command line override values in the supplementary property files. If you specify multiple property files that include different values for the same parameters, TIBCO BusinessEvents uses the value in the left-most file in the command line.

For example, consider this command line:

```
be-engine -p first.cfg -p second.cfg -p third.cfg
```

If `second.cfg` and `third.cfg` set different values for (as an example) *tibco.clientVar.MyVar*, and `first.cfg` does not include this parameter, TIBCO BusinessEvents uses the value in `second.cfg`. However, if `first.cfg` also includes a value for *tibco.clientVar.MyVar*, `TIBCO BusinessEvents` uses the value in `first.cfg`.

Setting up TIBCO BusinessEvents Engine as a Windows NT Service

You can configure the TIBCO BusinessEvents engine, Rule Management Server (RMS), or Views to start as a Windows NT service.

Note: TIBCO Hawk must be installed on the same machine for installing the BusinessEvents engines as a Windows NT service.

To set up the engines, follow these steps:

Procedure

1. Open the required TRA file for editing:
   - *BE_HOME*/bin/be-engine.tra for the TIBCO BusinessEvents engine
   - *BE_HOME*/rms/bin/be-rms.tra for RMS
   - *BE_HOME*/views/bin/be-views.tra for Views
2. Add the following environment paths under the common environment variables:

   ```
   tibco.env.HAWK_HOME=<absolute path where TIBCO Hawk is installed>
   ```

3. Edit the arguments for the application property to provide the absolute path to the EAR file:

   ```
   tibco.env.APP_ARGS=<absolute path of the EAR file>
   ```

   The value of the tibco.env.APP_ARGS property depends on the type of engine:
   - Absolute path of *BEprojectName*.ear file for the TIBCO BusinessEvents engine
   - *BE_HOME*/rms/bin/RMS.ear for Rule Management Server (RMS)
   - Absolute path of *ViewsProjectName*.ear file for Views

     Optionally you can provide name to the engine using the -n option of the tibco.env.APP_ARGS property. For example, for TIBCO BusinessEvents engine for FraudDetection project

     ```
     tibco.env.APP_
     ARGS=C\:/tibco/be/6.0/examples/standard/FraudDetection/fd.ear -
     n fddef
     ```

4. Edit the TIBCO standard classpath property to include classpath for RV and HAWK. Append %PSP%%HAWK_HOME%/lib%PSP%%RV_HOME%/lib to the existing value of the tibco.env.STD_EXT_CP property.

5. Add the following properties to define the Windows NT service configurations:

```
ntservice.name=<short name for Windows NT service>
ntservice.displayname=<full description of the service>
ntservice.starttype=<type of start, whether automatic or manual>
ntservice.binary.path.absolute=<absolute path of engine executable>
ntservice.interactive=false
```

   The `ntservice.binary.path.absolute` property identifies the absolute path of the respective executable:

   - *BE_HOME*`/bin/be-engine.exe` for the TIBCO BusinessEvents engine
   - *BE_HOME*`/rms/bin/be-rms.exe` for RMS
   - *BE_HOME*`/views/bin/be-views.exe` for Views

     For example, for TIBCO BusinessEvents engine the Windows NT service configurations are:

```
ntservice.name=TIBBEFD
ntservice.displayname=TIBCO BusinessEvents FD Default
ntservice.starttype=automatic
ntservice.binary.path.absolute=C\:/tibco/be/6.0/bin/be-
engine.exe
ntservice.interactive=false
```

6. Add the following properties to provide CDD file to the engine:

```
tibco.clientVar.CDD=<absolute path of the CDD file>
```

   The value of the `tibco.clientVar.CDD` property depends on the type of engine:

   - Absolute path of *BEprojectName*`.cdd` file for the TIBCO BusinessEvents engine
   - *BE_HOME*`/rms/bin/RMS.cdd` for RMS
   - Absolute path of *ViewsProjectName*`.cdd` file for Views

     For example, for TIBCO BusinessEvents engine for the FraudDetection project

```
tibco.clientVar.CDD=C\:/tibco/be/
6.0/examples/standard/FraudDetection/FraudDetection/fd.cdd
```

7. Set the processing unit ID (PUID) for the engine to the PUID that is set up in the CDD file.

   ```
   tibco.clientVar.PUID=<processing unit ID mentioned in the CDD file>
   ```

   For example, the PUID in the `fd.cdd` file is set as `default` for the BusinessEvents engine, then the value of PUID in the TRA file is also `default`.

   ```
   tibco.clientVar.PUID=default
   ```

8. (Optional) Specify the path of the log file in the `Engine.Log.Dir` property. For example:

   ```
   Engine.Log.Dir C:/temp/logs
   ```

9. Save and close the TRA file.
10. Open the command prompt and browse to the `bin` directory of the respective engine.
11. In command prompt run the commands to install or uninstall the Windows NT service.

    For example, to install the TIBCO BusinessEvents engine as Windows NT service:

    ```
    BE_HOME/bin>be-engine.exe -install
    ```

    For example, to uninstall the TIBCO BusinessEvents engine as Windows NT service:

    ```
    BE_HOME/bin>be-engine -uninstall
    ```

Result

To verify, if the service is set up correctly, browse to the `bin` directory of the respective engine in the command prompt and run the engine without any argument. If the service does not start check logs for the cause.

Note: For any change to the TRA file, run `-uninstall` and `-install` commands again for the changes to take effect.

Deployment with TIBCO Administrator

You can use TIBCO Administrator for deploying, hot deploying, undeploying, starting and stopping TIBCO BusinessEvents engines.

Within an Enterprise Archive Resource (EAR) file, a TIBCO BusinessEvents Archive (BAR) file contains the compiled agent files for all agents. When you upload an EAR file, The BAR file appears here in the TIBCO Administrator UI:

Application Management > application_name > Configuration > application_name.bar

The default value of *application_name* is provided by the name field in the EAR file.

Note:

- The only supported transport option in this release is the `local` option.
- Message encoding: For deployment with TIBCO Administrator the message encoding specified in the CDD file General settings must match the TIBCO Administrator domain's message encoding. The default message encoding for TIBCO BusinessEvents and TIBCO Administrator is ISO8859-1.
- Troubleshooting: When you deploy with TIBCO Administrator, remember to check the TIBCO Administrator logs (as well as TIBCO BusinessEvents logs) when troubleshooting TIBCO BusinessEvents deployment or runtime issues.

TIBCO Administration Domains

You can use an existing domain or create another one (using Domain Utility) for your TIBCO BusinessEvents applications and the hardware they run on.

The first time you log in to the TIBCO Administrator user interface after installing the software, use the username and password entered during installation. You can then create additional users and passwords as needed. If TIBCO Administrator was already installed before you installed TIBCO BusinessEvents, you might have to contact the person responsible for administering the software to get login credentials for an existing administration domain.

Property Overrides and Precedence

Properties set in TIBCO Administrator are added to the TRA file that TIBCO Administrator generates at deploy time (based on the default `be-engine.tra` file). However, See Order of Precedence at Run time for more details.

Global variables that are overridden at the deployment level, however, are stored in a different location. See Overriding of Global Variables in TIBCO Administrator .

Using AppManage for Scripted Deployment to a Domain

Instead of using the TIBCO Administrator user interface, you can perform scripted deployment to a TIBCO Administrator domain using the AppManage utility. Use of AppManage is explained in *TIBCO Runtime Agent Scripting Deployment User's Guide*.

Deploying a Project in a TIBCO Administrator Domain

To deploy a project in a TIBCO Administration Domain, you must update the `be-engine.tra` files on all machines to include the TIBCO Hawk information. You must also build the EAR file and perform other tasks, as needed.

Procedure

1. Open the *BE_HOME*`/bin/be-engine.tra` file for editing.
2. If it is not already present, add the following variable and set the value to the TIBCO Hawk home:

```
tibco.env.HAWK_HOME=Hawk_Home
```

3. If it is not already present, append the following to the value of the standard classpath, `tibco.env.STD_EXT_CP`:

```
    %PSP%%HAWK_HOME%/lib%PSP%
```

4. If it is not already present, append the following to the value of the standard path, `tibco.env.PATH`:

```
%PSP%%HAWK_HOME%/bin%PSP%
```

5. Save the file.

> **Note:** You can update the engine TRA file for any specific application arguments. Use `tibco.env.APP_ARGS` for generic application specific arguments.

Other Deployment Tasks

You might need to perform additional tasks while deploying a project with the TIBCO Administrator.

Set default or specific CDD file and processing unit names

If you want to let the target engines find the CDD at a default location, name the CDD file `default.cdd` and keep it in the root of the project folder. If you want to let the target engines choose a processing unit by default, then in the CDD file, name one of the processing units `default`. When you deploy, the processing unit named `default` starts. Note that these names are case-sensitive.

You can also specify different CDD file and processing unit values at deploy time.

Enable hot deployment, as needed

If you want to be able to hot deploy changes to the running engine, open the CDD file in the TIBCO BusinessEvents Studio project, select the processing unit or units you will deploy, and check the Hot Deploy checkbox. See "Agent and Processing Unit Configuration" in the *TIBCO BusinessEvents Developer Guide*.

Enable service-settable global variable overrides and build the EAR

Service settable global variables are only available if the "Include All Service Level global variables" checkbox in the Build Enterprise Archive dialog is selected. Select as needed.

Then build the EAR. See Building an Enterprise Archive (EAR File). See Enterprise Archive (EAR) Files for details.

Enable custom log configurations

To use custom log4j logconfiguration for a project through teagent, do the following settings:

- **Add a custom property**

  Add the property `log4j2.configurationFile` for a project. For a detailed process, see Adding a New BusinessEvents Property.

- **Add a custom log configuration file to the engine classpath**

  On the Machine Management page, by using the Upload External Artifacts wizard, upload your log configuration XML file to the selected *BE_HOME*. For a detailed process, see Uploading External Artifacts .

  > **Note:** Keep a different name for your custom log configuration file from the default `log4j2.xml` file name.

- **Disable CDD Log Configurations**

  To disable the CDD Log Configurations, open the CDD file for the project in the TIBCO BusinessEvents Studio and uncheck the **Enable** option in the Log Configurations.

  This setting makes TIBCO BusinessEvents use your custom log configuration XML file from the engine classpath instead of the CDD configurations.

  For a detailed process, see the Configuring Log Configurations topic in *TIBCO BusinessEvents Configuration Guide*.

Overriding of Global Variables in TIBCO Administrator

Global variables defined in a project appear in TIBCO Administrator if they are configured to do so at design time.

Levels of Override

You can override global variable default values as follows:

Deployment Level

If the Deployment Settable checkbox is selected at design time in the Global Variable editor, you can override it at the deployment level. Overrides set at the deployment level are used in all deployed engines.

Service Level (Same Scope as DeploymentLevel in TIBCO BusinessEvents)

If the Deployment Settable checkbox and the Service Settable checkbox are both checked at design time in the Global Variable editor, you can override at the service level or service instance level. However, overrides set at the service level are used for all engines because all services (all BARs, that is all PU definitions) are merged when deployed using TIBCO Administrator.

Engine Instance Level

If the Deployment Settable checkbox and the Service Settable checkbox are both selected at design time, you can override at the service instance level. Overrides set at the service instance level are used for the specific engine (PU instance) represented by that service instance.

> ⚠ **Caution:** Caution: Overriding a global variable at the service or service instance level breaks the connection with higher-level overrides for that global variable. By default, global variable overrides done at the application level are propagated to all lower-level global variable settings at run time. However, when you override a global variable at the service level, TIBCO Administrator no longer propagates application-level overrides for that variable to the service or service instance levels at runtime. Similarly, if you override a global variable at the service instance level, any subsequent overrides you make to that global variable at the service level (or application level) are ignored at run time.
>
> This behavior applies to overrides made using the `appmanage` utility as well as those made using the TIBCO Administrator UI.

Specifying Global Variable Groups

If global variables are defined in the TIBCO BusinessEvents project using groups, specify the group path using forward slashes. For example, if a variable *JMSuri* is located under a

group called `URIs`, specify the variable as `tibco.clientVar.URIs/`*`JMSuri`*.

Enabling Service Settable Global Variables

Service settable global variables are only available if the `Include All Service Level global variables` checkbox in the Build Enterprise Archive dialog is selected.

Runtime Location of Global Variable Override Settings

The runtime location of override settings depends on the level at which the override was done:

Deployment level override

These are in the following folder.

`TRA_HOME``/domain/`*`domain_name`*`/datafiles/`*`application_name`*`_root`

Service and service instance level overrides

These are in the TRA file generated by the TIBCO Administrator.

Project Deployment

After performing all required actions and building an EAR file, you are ready to configure the system for deployment and deploy it.

Note: Do not use the fault tolerance features of TIBCO Administrator. Instead, use the tab **Agent Classes > AgentClassName > Max Active** setting. To maintain one active and one standby agent, deploy two agents of the same class and set the Max Active setting to `1`. (You can also deploy more than two agents and set the property to a larger number for different use cases.)

Note: The only supported transportation option is **local**.

Deploying a Project EAR in a TIBCO Administrator Domain

Procedure

1. Ensure that the following are started on the machine whose engine properties you want to change:
   - TIBCO Administrator service for the administration domain.
   - TIBCO Hawk service for the administration domain.
2. Start the TIBCO Administrator GUI:
   - Windows: **Start > Programs > TIBCO > TIBCO Administrator Enterprise Edition *version* > TIBCO Administrator**
   - Web browser: `http://`*`host-name`*`:`*`port`*`/` (where *`host-name`* is the machine name and *`port`* is the HTTP port specified at installation. It is 8080 by default.)
3. Select the administration domain for the application and provide the username and password assigned during installation, or other administrator user credentials.
4. Depending on the application you are deploying, proceed with the steps described either in Deploying a Project EAR for the First Time or in Deploying a Project EAR for an Existing Application.

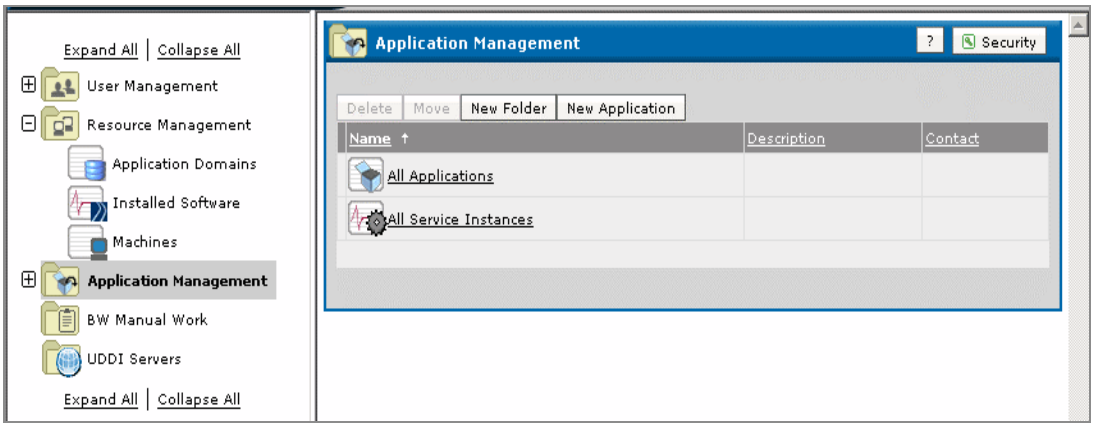
Deploying a Project EAR for the First Time

If you are deploying a new application, perform these steps after selecting the administration domain and providing the username and password.

Procedure

1. Click **Application Management** (in the left panel).

2. Click the **New Application** button.



3. At the Upload EAR File dialog, click **Browse** and select the EAR file you want to deploy.

   Click **OK.**

4. At the New Application Configuration dialog, set the Application Parameters and Services settings as desired (click **Help** for details). You can change default names:

   Name: Set by default to the TIBCO BusinessEvents Studio project name

   Deployment Name: Set by default to the TIBCO BusinessEvents Studio project name prepended with the domain name.

5. Click **Save**.

   If the application does not appear in the list of applications, check Deploying a Project EAR in a TIBCO Administrator Domain and ensure you have met all prerequisites.

6. Continue with the steps described in Deploying on a Service Level.

Deploying a Project EAR for an Existing Application

When deploying an existing application, you can navigate to the Configuration Builder panel by expanding the explorer nodes on the left to **Application Management > application_name > Configuration**.

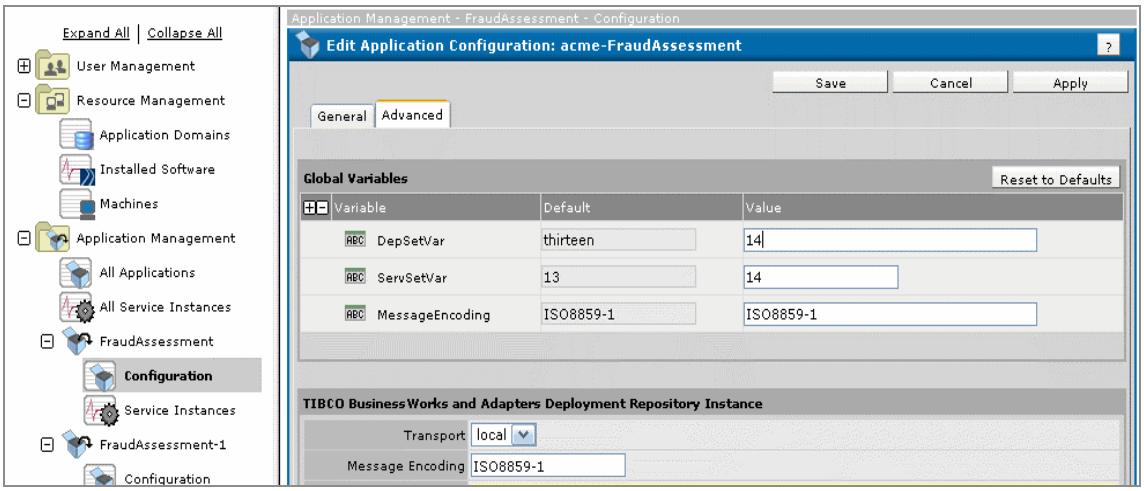
Procedure

1. To set deployment wide settings such as deployment-wide global variable overrides, select the application (which is at the top level in the hierarchy) and select the **Advanced** tab.

   > **Note:** See Overriding of Global Variables in TIBCO Administrator for important considerations and cautions about how to override global variables.

2. Ensure that the Transport field is set to **local**. Local is the only supported transportation option.

   An example application level **Advanced** tab is shown next:



3. Click **Save** when you are done. The Configuration Builder panel appears again.
4. Continue with the steps described in Deploying on a Service Level.

Deploying on a Service Level

In the Configuration Builder panel, perform these steps to set project-related settings that apply to all deployed engines on all machines.

Procedure

1. Select the *application-name*`.bar` entry. It is one level below the top level in the hierarchy.

2. Select its **Advanced** tab.

   Note: For TIBCO BusinessEvents, settings at both the application and service levels affect the entire deployment.

3. As desired, enter the name of the CDD file and processing unit (PUID) you want to use for *all* deployed engines. You can use a relative or absolute path.

   Note: The TIBCO BusinessEvents engine looks for the CDD and processing unit as follows:

   - The engine looks first in the file system, under the given path. If the path is specified as a relative path, it is relative to the working directory (in this case:
     `...tibco/tra/domain/domainName/application/appName/`
   - If no CDD is found in the file system, the engine looks within the EAR, under the given path. If the path is specified as a relative path, it is relative to the project root.

4. You can also override any service-settable global variable values as desired. Values entered here apply to all deployed engines.

   Note:
   - Service-settable global variables are only available if the Include All Service Level Global Variables checkbox in the Build Enterprise Archive dialog is selected before generating the EAR file.
   - Global Variable Overrides: If you override a global variable at a lower level, subsequent changes at higher levels are ignored.

5. Click **Save** when you are done. The Configuration Builder panel appears again.
6. Select the machines in the administration domain to which you will deploy the application.
7. In the Configuration Builder panel, click the service (*`application`*`.bar`) name.

   The service name is nested under the application name. In the **General** tab, Target Machines panel, the current machine is available by default.

8. Select **Add to Additional Machines** and select the machines to which you will deploy the application.

   > **Note:** You can select the same machine more than one time if you want to deploy the application more than once on a machine. For example, you would do this when you want to deploy two different processing units to one machine.

9. Click **Save**.
10. Continue with the steps described in Deploying on an Instance Level.

Deploying on an Instance Level

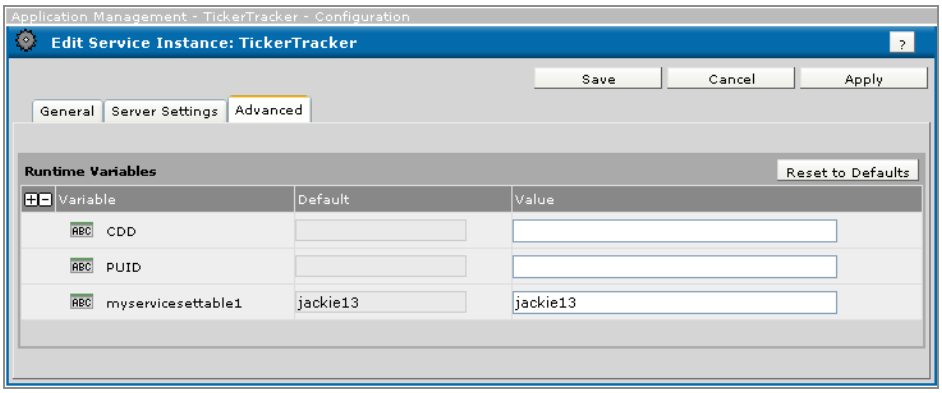
These steps will set project-related settings that apply to deployments on specific machines.

Before you begin

Make sure you have finished all the steps as described in Deploying on a Service Level.

Procedure

1. At the Configuration Builder panel, select a machine-level entry (*machineName - projectName*). These entries appear below the *application-name*.bar entry. Then select its **Advanced** tab. You see a dialog similar to the following:



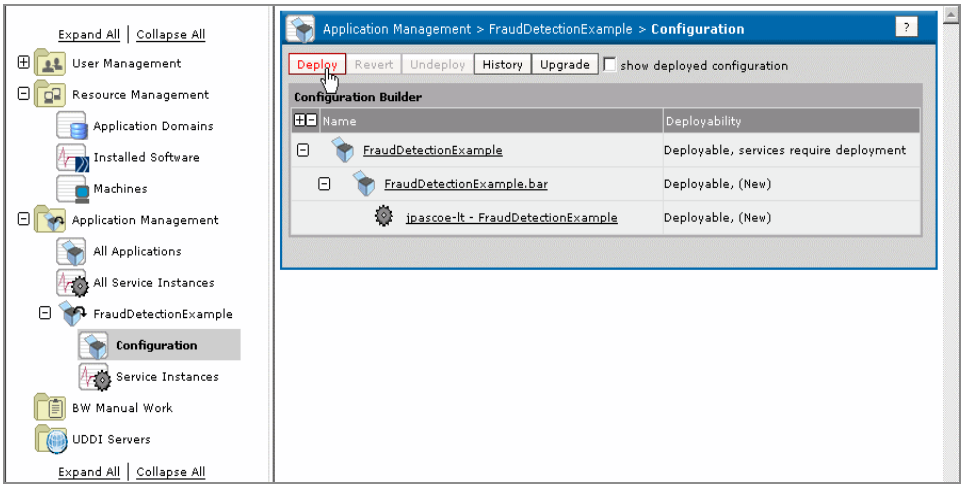
   Here you can also override any service-settable global variable values as desired. (See Overriding of Global Variables in TIBCO Administrator for important information.)

2. Click **Save** when you are done. The Configuration Builder panel appears again.

   The same project-related options are available here as at the .bar level, but here they apply only to an engine deployed to the selected machine. As desired, enter the name of the CDD file and processing unit (PUID) you want to use for this deployed engine. You can use a relative or absolute path. The same project-related options are available here as at the .bar level, but here they apply only to an engine deployed to the selected machine. Enter the name of the CDD file and processing unit (PUID) you want to use for this deployed engine. You can use a relative or absolute path.

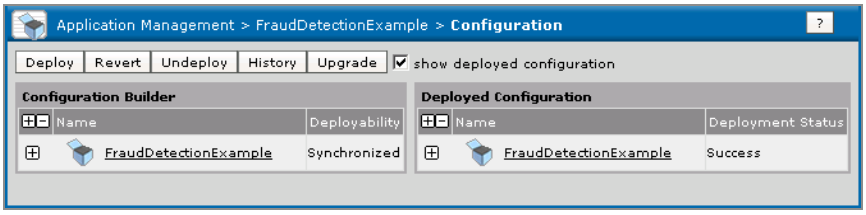
3. You are now ready to deploy. By default, an engine starts when you deploy it. You can also start and stop engines as separate actions.

   Navigate to the main Configuration Builder dialog and click **Deploy**.



4. At the Deploy Configuration dialog, configure settings if desired then click **OK**. The application deploys, and the Configuration dialog displays again.

   You can select the **Show deployed configuration** checkbox to display the Deployed Configuration panel and verify success:



Hot Deployment

You can make certain changes to a TIBCO BusinessEvents project and apply them to a running engine, without having to shut down the engine. This is known as hot deployment.

In an active agent, the hot deployment process waits for the current RTC cycle to complete and then injects the changes before the next RTC cycle starts. You can only hot deploy to an application that was enabled for hot deployment *before* it was deployed. When enabled for hot deployment, the application listens for changes in the EAR file. When you replace an EAR file, TIBCO BusinessEvents detects the change and performs hot deployment. See Enabling Hot Deployment.

The permitted changes available to you depend partly on the type of object management in use. The permitted changes are listed in the section Modifications Allowed in Hot Deployment. If you attempt to hot deploy an EAR file that includes unsupported modifications, TIBCO BusinessEvents rejects the EAR file.

Performing hot deployment requires changing the execution code at run time. This is made possible using the `-javaagent` option. The `-javaagent` option is provided in the `be-engine.tra` file as shipped.

This section explains how to hot deploy with TIBCO Administrator, and also to an engine that was started at the command line.

Modifications Allowed in Hot Deployment

You can make only certain changes during a hot deployment. Also, supported modifications for Cache OM are more limited than those for In Memory OM.

Hot Deployment Supported Modifications

Resource	New	Modify	Delete
Rules	Yes	Yes	Yes
Rule Functions	Yes	Yes	Yes

Resource	New	Modify	Delete
Concepts***	Yes		
Global Variables	Yes	Yes	Yes
Simple Events*	Yes		
Score Cards*	Yes		
Time Events*	Yes		
State Machines and States**	Yes		
State Machine Transitions	Yes	Yes	Yes
State Machine Timeout Expressions	Yes	Yes	Yes
State Machine Timeout Actions	Yes	Yes	Yes
State Machine Entry Actions	Yes	Yes	Yes
State Machine Exit Actions	Yes	Yes	Yes
State Machine Event Timeout Actions	Yes	Yes	Yes
Channels and Destinations*			
Concept Property***	Yes		

*** Cache object management**

When Cache object management is used, hot deployment is available only for rules, rule functions, global variables, event timeout actions, and the following state machine components: transitions, entry and exit actions, timeout expressions and actions.

**** For state machine hot deployment**

You can only hot deploy new state machines (and state machine states) that are associated with new concepts, that is, concepts added in the same hot deployment. Adding a state machine or state machine component that is associated with an existing

concept modifies that concept, and concept modification is not allowed. Also see notes for Cache object management above.

***** Used only when cache is enabled**

This is an alter space feature used only when the cache is enabled. You can add new concepts and properties to the existing concepts and it is supported only for the concept types that are cache-only.

The new concept and properties work in rule conditions and actions and in rule functions, which should be properly saved to cache and backing store if any. It is supported for Cache OM with shared nothing backing store or no backing store.

With no backing store, select the **Store Properties As Individual Fields** checkbox in the CDD file under **Cluster > Object Management: [Cache]** configuration. This property is selected by default when using shared nothing persistence.

Adding a concept and concept property is only supported for concept types that are cache-only. Adding a concept property of the type `Contained Concept` with the contained concept type set to an existing concept is not supported. When adding a concept, it takes the default domain object settings.

Enabling Hot Deployment

As a safety measure, hot deployment is disabled by default. You must enable hot deployment for specific processing units and then deploy those processing units and start them. You can then perform hot deployment to the running engines (processing units) that are enabled for hot deployment.

Procedure

1. In TIBCO BusinessEvents Studio Explorer, open the CDD file in the CDD editor.
2. Select the **Processing Unit** tab.
3. Select a processing unit and select the **Hot Deploy** checkbox.

   Repeat for all processing units you want to enable for hot deployment.

   See Agent and Processing Unit Configuration in *TIBCO BusinessEvents Developer Guide* for more details.

4. Start the TIBCO BusinessEvents application using the CDD file you updated.

   For details on performing a hot deployment, see Hot Deployment in a TIBCO Administrator Domain and Performing Hot Deployment Outside a TIBCO Administrator Domain.

Hot Deployment in a TIBCO Administrator Domain

You can perform hot deployment of the TIBCO BusinessEvents project after it has been deployed to a TIBCO Administrator domain.

This procedure assumes that the following requirements are met:

- The processing units that you want to hot deploy were already enabled for hot deployment before they were deployed (see Enabling Hot Deployment).
- Your project complies with the requirements for deploying to TIBCO Administrator.
- You have modified the TIBCO BusinessEvents Studio project and built the EAR file, following the limitations shown in Modifications Allowed in Hot Deployment.
- The new EAR file has the same name as the existing one.

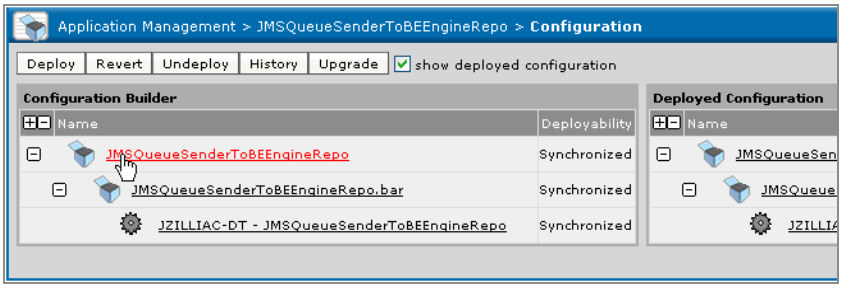
Performing Hot Deployment in a TIBCO Administrator Domain

You can perform hot deployment of the TIBCO BusinessEvents project after it has been deployed to a TIBCO Administrator domain.

Procedure

1. As needed, ensure that all the following are started on the machine running the processing unit or units you want to hot deploy to:
   - TIBCO Administrator service for the administration domain.
   - TIBCO Hawk service for the administration domain.

2. Start the TIBCO Administrator GUI:
   - Windows: **Start > Programs > TIBCO > TIBCO Administrator Enterprise Edition > version > TIBCO Administrator**.
   - Web browser: `http://host-name:port/` (where *host-name* is the machine name and *port* is the HTTP port specified during installation, 8080 by default)
3. Select the administration domain for the application and provide the username and password assigned during installation, or other administrator user credentials.
4. Expand to **Application Management > application_name > Configuration**.
5. In the Configuration Builder panel, select the application (at the base of the tree).



6. In the Edit Application Configuration dialog, click **Upload New EAR File**.
7. At the Upload EAR File dialog, click **Browse**, select the EAR file you want to deploy, and click **OK**.
8. Confirm the upload by clicking **OK** again, then click **Save**. Verify that the Deployability column displays Deployable.
9. Click **Deploy**. You see the Deploy Configuration dialog.
10. Clear these checkboxes (if they are selected):
    - **Stop running services before deployment**.
    - **Start successfully deployed services**.
    - **Force redeployment of all services**.

      (When the Stop running services before deployment checkbox is selected, you see an additional setting, Stop the services that have not stopped after (seconds). It is removed when you clear the checkbox.)

11. Click **OK**. TIBCO Administrator performs the hot deployment of your modified TIBCO BusinessEvents project. If deployment is successful, the Deployed Configuration panel in the Configuration dialog displays Success in the Deployment Status column.

Performing Hot Deployment Outside a TIBCO Administrator Domain

You can perform hot deployment when the TIBCO BusinessEvents project has not been deployed to a TIBCO Administrator domain only if the deployed application was enabled for hot deployment before it was deployed. You need to modify the project as needed and build the EAR file.

Procedure

1. In TIBCO BusinessEvents Studio, modify the TIBCO BusinessEvents project according to your needs. See Modifications Allowed in Hot Deployment for a list of modifications you can make.

   Then rebuild the project EAR file.
2. Verify that the new EAR file have the same name as the existing one.
3. Replace the EAR File that was used to start the engine with the modified EAR file.
4. Ensure that the modified EAR file has the same name and is placed in the same directory as the EAR file that was used to start the engine. The engine notices the changed file and performs the hot deployment at the next RTC cycle.

User Authentication

User authentication can be set using a file-based system and integration with an LDAP system.

To set up authentication, add and configure the appropriate properties in the project CDD.

Pluggable JAAS Login Module

User authentication is performed using a JAAS login module. Java Authentication and Authorization Service (JAAS) is a pluggable part of the Java security framework.

With advanced configuration, you can substitute a different implementation of the JAAS login module than the one provided, or you can add the provided login module to your existing JAAS login configuration file (thus providing multi-stage authentication).

Authentication Options

You can choose between file-based and LDAP user authentication.

File-Based Authentication

This method authenticates a user against user data stored in a file-based repository. This method is not recommended for production purposes. In file-based authentication, define a list of user names, passwords, and roles in the file (default) `users.pwd` file. This file is commonly referred to as the password file.

LDAP Authentication

This method authenticates users against a directory server using LDAP as a protocol. TIBCO BusinessEvents applications can leverage this information to authenticate users. The role information is configured through an LDAP attribute like the `nsroledn` attribute in Oracle Directory Server. The LDAP attribute differs in different directory server products. The details of configuring LDAP authentication are beyond the scope of this documentation. Consult your LDAP product documentation.

To use LDAP authentication, add roles, MM_ADMINISTRATOR and MM_USER in the LDAP directory for the relevant users.

Note:

- MM_ADMINISTRATOR: Users with this role can execute methods, for example, to deploy, start, and stop engines, and invoke method operations.
- MM_USER: Users with this role can view MM Console, but cannot deploy, start, or stop engines, or invoke method operations.

Authentication In Various Components

Authentication is used in components of various TIBCO BusinessEvents products:

TIBCO BusinessEvents Decision Manager RMS Component

File-based authentication is enabled by default for the TIBCO BusinessEvents Decision Manager RMS component and LDAP authentication is supported.

This component also uses authorization. Authorization details are provided in Access Control Configuration .

Authentication Configuration

Using the provided JAAS login module, you can select file-based authentication or LDAP-based authentication and configure each authentication option.

You can use a different authentication type and a different password file or LDAP settings for each TIBCO BusinessEvents product that uses authentication. You can actually do so for each engine (processing unit) configured for authentication, but this is not usually needed.

Enabling Authentication and Selecting Authentication Type

You can select either file-based authentication or LDAP-based authentication and enable it for the project.

Procedure

1. In TIBCO BusinessEvents Studio, import and open the relevant project and open its CDD file, as follows:

   For TIBCO BusinessEvents Decision Manager:

   - *BE_HOME*/rms/project/BRMS > RMS.cdd > Cluster properties > RMS property group

2. In the CDD file add the following property if it is not present and specify the value as desired:

   `be.auth.type=[file|ldap]`

3. Do one of the following:

   - To configure LDAP authentication, add and configure the LDAP properties shown in Authentication Property Reference for the TRA File. Familiarity with LDAP is required. Details are not provided in this guide.
   - To configure file-based authentication, see Configuring File-Based Authentication.

     Note: For Active Directory Configuration, authentication requires the domain name, for example, `abc@acme.com`, and not distinguished name (which is used with Oracle Directory Server). If you are using Active Directory for authentication, ensure that the `userPrincipalName` attribute is set on AD server.

Configuring File-Based Authentication

Configure file-based authentication and enable it for the project.

Procedure

1. In the CDD file, add (or configure) the property `be.auth.file.location` and set the value to the location of your password file.
2. Locate and open the password file. Its location is specified in the CDD file.

   See Enabling Authentication and Selecting Authentication Type for default location details.

3. Add each user on a separate line using this format:

```
Username:password:role,role,role;
```

Do not use spaces. For example, here are some entries that might be used in TIBCO BusinessEvents Decision Manager:

```
Mark:A31405D272B94E5D12E9A52A665D3BFE:BUSINESS_USER,APPROVER;
James:21232f297a57a5a743894a0e4a801fc3:RULE_ADMINISTRATOR;
```

Note:

- You must hash the password with the MD5 (Message-Digest 5) hashing algorithm.
- Roles are used for access control (authorization). Access control is used only by TIBCO BusinessEvents Decision Manager. See Access Control Configuration .

Authentication Property Reference for the TRA File

These are authentication properties used in the TRA file.

Property	Notes
`java.property.be.engine.jmx.connector.port`	Specify this property in each relevant engine TRA files to open the JMX connector port for monitoring and management. It is also used for hot deployment of decision tables in TIBCO BusinessEvents Decision Manager (see *TIBCO BusinessEvents Decision Manager User Guide* for details).

Property	Notes
	Note: You can also set the JMX connector port for deployment with TIBCO Administrator using this CDD property: `be.engine.jmx.connector.port`
`java.property.be.engine.jmx.connector.authenticate`	Set to `true` to enable authentication. Set to `false` (or leave commented) to disable authentication. The default is `false`.

For JMX authentication properties configurations in the BusinessEvents Enterprise Administrator Agent properties file, see BusinessEvents Enterprise Administrator Agent Configuration Reference.

Common Authentication Properties for the CDD File

These are common authentication properties used in the CDD file.

Property	Notes
`java.security.auth.login.config`	Provides the absolute location for the login module configuration used by JAAS. Only advanced users should change this value (additional configuration is also needed). The locations of the provided files are as follows: • TIBCO BusinessEvents Decision Manager: *BE_HOME*`/rms/config/security/jaas-config.config`

Property	Notes
`be.auth.type`	Specifies the authentication mechanism. The valid values are as follows: • `file`: File-based authentication. Uses a password file. • `ldap`: LDAP-based authentication Uses a pre-existing LDAP setup in use in your environment. Add and configure the properties shown in . Default is `file`
`be.auth.file.location`	Specifies the absolute filepath to and name of the password file. This file is used for file-based authentication. Each project can have a different file. The locations of the provided files are as follows: • TIBCO BusinessEvents Decision Manager: *BE_HOME*`/rms/config/security/users.pwd`.

LDAP Authentication Properties for the CDD File

These are the LDAP authentication properties used in the CDD file.

Property	Notes
`be.auth.ldap.type`	Use this property only if you want to use OpenLDAP for LDAP authentication. The property is not required for Oracle Directory Server or Windows Active Directory server. Set his property `openldap` to use the RMS server with OpenLDAP

Property	Notes
`be.auth.ldap.port`	Specifies the port for LDAP authentication.
`be.auth.ldap.adminDN`	Specifies the base distinguished name (DN) for admin login. For example: `cn=Directory Administrators, dc=na, dc=tibco, dc=com.`
`be.auth.ldap.adminPassword`	Specifies the password for the LDAP administrator DN.
`be.auth.ldap.baseDN`	Specifies the base tree in LDAP under which users can be searched. For example, `dc=na, dc=tibco, dc=com`.
`be.auth.ldap.roleAttr`	Specifies the name of the attribute used by the LDAP server for role information of a user. Set the value to `member` for RMS server with OpenLDAP The default value is `nsroledn` (for Oracle Directory Server).
`be.auth.ldap.uidattr`	Specifies the name of the attribute used by the LDAP server for username information. Allowable values are as follows: • `uid` for Oracle Directory Server • `cn` for ActiveDirectory. The default value is `uid`.
`be.auth.ldap.useRoleDN`	Set this property to true to use the fully qualified name of the attribute used by the LDAP server for role information of a user.

Property	Notes
	Set this property to false to use only the name of the attribute, which is shown in the notes for the `be.auth.ldap.uidAttr` property. The default value is `true`.
`be.auth.ldap.objectClass`	Specifies the `ObjectClass` attribute value for DS. Many object classes can exist, for example, `inetOrgPerson` on Oracle Directory Server, and `user` on Active Directory. If the search should span all object classes, keep this value empty or specify an asterisk ("*").
`be.mm.user.role`	*Optional*. Specify the value of the property to the LDAP `group_name` which have user rights in BusinessEvents.
`be.mm.admin.role`	*Optional*. Specify the value of the property to the LDAP `group_name` which have admin rights in BusinessEvents.
`be.auth.ldap.dnAttr`	Specifies the name of the attribute that contains the fully qualified name. The default value is `distinguishedName`.
`be.auth.ldap.ssl`	Specifies a secure connection to the LDAP host is to be established. The default value is `false`. Set the property to `true` to enable a secure connection.
`be.auth.ldap.filter.<attributeName>`	While using LDAP for authentication, you can filter users who get access to TIBCO BusinessEvents WebStudio based on their attributes on the LDAP server. The `<attribute>`

Property	Notes
	in the property is the name of the attribute. Set the value of the property to the value of the attribute that a user should have to get access to TIBCO BusinessEvents WebStudio. For example, if you have a large number of LDAP users and they need to be authenticated based on the attribute of their roles, TIBCO BusinessEvents can filter them based on their roles.

Access Control Configuration

Access control is a core product feature used by RMS projects and available in TIBCO BusinessEvents WebStudio.

For each RMS project, set up an access control file where you group the project resources as desired, giving each group (or individual resource) an ID. Use these IDs to assign permissions to each user role.

Access is defined using roles. If file-based authentication is used, roles are defined and assigned to users in the password file (by default called `users.pwd`). If LDAP-based authentication is used, roles are defined and assigned to users in the LDAP directory.

Note:

- You must use only the roles defined in the password file or LDAP directory (depending on authentication type used) when configuring the access control file.
- User role and username should not be same. Each username and role name should be unique.

Guidelines for Configuring Access Control

A project's access control file is an XML file named *`RMSProjectName`*`.ac`.

The ACL file is stored in the directory specified by the `RMS.cdd` property `ws.projects.acl.location`.

In the access control file `resources` element, you can create `resource` elements to define groups of resources to suit your needs. Give each resource element an ID. In the `entries` element, add one `entry` element for each user role to define the access permissions for that role, using the resource IDs and `action` elements. This brief summary is provided so you can understand the following guidelines. For details, see Structure of the Access Control File.

After migrating a project from earlier versions to TIBCO BusinessEvents version 6.0.0, ensure to change the name of the resource SHAREDASCON (if configured in the .ac file) to SHAREDASLEGACY in the *RMSProjectName*.ac file.

Note: Replace the XML special character in the role names (if present) in the access control file with the following characters:

- "&" by "&"
- " ' " by "'"

You can use two general approaches to setting permissions. The general aim is to simplify the setup, minimizing the number of permissions you have to set in the access control file.

Allow everything and specify exceptions

One approach is to grant wide permissions using large resource groupings, and then selectively deny permissions within those groupings.

For example, suppose you define two resources as follows:

```
<resource name="/Concepts/*" id="AllP" type="PROPERTY"/>
<resource name="/Concepts/Person/CustID" id="CID" type="PROPERTY"/>
```

The first resource element defines a resource group consisting of all concept properties in the /Concepts project folder. The second element specifies one property in one concept. (The setup details are explained later in the chapter.)

Then you define permissions using those resources. For example, for a role named CallCenter you might set up permissions as follows:

```
<entry>
  <role name="CallCenter"/>
  <permissions>
    <permission resourceref="#AllP">
      <action type="read">ALLOW</action>
    </permission>
    <permission resourceref="#CID">
      <action type="read">DENY</action>
    </permission>
  </permissions>
</entry>
```

With these settings, you give users with the `CallCenter` role the `read` permission for all properties in the `/Concepts` directory except the `custID` property.

Tip: An example of an access control file (`CreditCardApplication.ac`), giving all permissions available for the credit card application example, is located in the following directory: *`BE_HOME`*`\rms\config\security`

Deny everything and specify exceptions

Another approach is to deny all permissions (which is the default setting for all permissions) and then give permissions to specific resources or groups of resources as needed.

Combining two approaches

You can combine these two approaches in one access control file. For example, you can give broad permissions to one project folder, and then specify exceptions within that folder. For another folder you might give permissions selectively.

Structure of the Access Control File

The access control file for a project is an XML file.

The access control file has the following elements:

```
<resources>
  <resource id="id" type="ResourceType"/>
  <resource id="id" name="ProjectPath" type="ResourceType"/>
  . . .
</resources>
<entries>
  <entry>
    <role name="RoleName"/>
    <permissions>
      <permission resourceref="#id">
        <action type="ActionType">[ALLOW|DENY]</action>
      </permission>
      . . .
    </permissions>
```

```
  </entry>
  . . .
</entries>
</acl>
```

- The `entries` element contains one `entry` for each role. For each role, you define one set of `permissions`. Each `permission` has of the following attributes:
- The `resourceref` attribute references a `resource ID` defined in the resources element. It identifies a resource or set of resources.
- The `name` attribute specifies the `project path` to the resource or resources. (The `name` attribute is not used when you specify permissions for an entire resource type.)
- The resource `type` attribute specifies what types of resources in the specified `name` attribute project path are included in the permission.
- The action `type` attribute specifies an `action type`, for example, `create`. This attribute determines the kind of action a user has permission to do, for the specified resource or resources.

Access Control Files

XML files with the extension `.ac` are used to create access control settings.

You can create or modify an *`RMSProjectName`*`.ac` file using any XML editor. This section explains the elements used to define access control, the ways to add or edit access control files. It also explains where to place the files so that they can be used by the RMS, Decision Manager components, and TIBCO BusinessEvents WebStudio.

Examples shipped with the product contain access control files that you can use as models.

Required Location of Access Control Files

The access control file for an RMS project must be placed in the location specified by the RMS server CDD property `ws.projects.acl.location`. An RMS project's ACL file must be named using the format *`RMSProjectName`*`.ac`.

Specification and Grouping of Project Resources

In the `resources` element, you can group the project resources in whatever way supports the permissions you want to set.

Give each grouping or individual resource an ID that is used when defining the permissions.

Grouping Resources by Resource Type

The broadest resource grouping is provided by setting permissions at the level of resource type. This method groups all resources of that type in the project. To set a resource type resource group, associate an ID with a resource type, and do not use the `name` attribute:

```
<resource id="ID" type="ResourceType"/>
```

For example: `<resource id="C" type="CONCEPT"/>`

Using Resource Type as a Filter

How you specify the resource group is partly determined by the resource type attribute. The resource `type` can act as a filter. For example, suppose in the `name` attribute you specify a directory that includes events and concepts. If you set the `type` attribute to `"CONCEPT"` then the ID associated with this grouping is used to set permissions only on the concepts in that folder (and its subdirectories).

You could create a second grouping whose `type` specifies `"EVENT"` so that you can set permissions on events in that folder branch separately.

Specifying an Individual Resource

To specify an individual resource, provide the *project path* to the resource in the `name` attribute. The project path is the folder path to the ontology entity, as seen in the Explorer panel. The example below shows how to specify an ID that is associated with the `FirstName` property of the `Person` concept:

```
<resource name="/Concepts/Person/FirstName" id="FN" type="PROPERTY"/>
```

Grouping Resources Using Wildcards

You can associate groups of resources with an ID using the wildcard character in the project path. The asterisk (*) is used as the wildcard character. For example:

```
<resource name="/someFolder/* "id="AllP" type="PROPERTY"/>
```

Grouping Resources by Resource Type

The broadest resource grouping is provided by setting permissions at the level of resource type. This method groups all resources of that type in the project. To set a resource type resource group, associate an ID with a resource type, and do not use the `name` attribute:

```
<resource id="ID" type="ResourceType"/>
```

For example: `<resource id="C" type="CONCEPT"/>`

See Resource Types and Corresponding Action Types for a list of resource types, and the action types that are valid for each resource type.

Permissions Definition

Define a list of resource IDs according to the way you want to group resources and actions.

All items included under one resource ID must be of the same resource type (or type of activity, such as checking out a project).

For each user role, add a set of permissions.

```
    <role name="Administrator"/>
    <permissions>
       <permission resourceref="#PRJ">
          <action type="checkout">ALLOW</action>
       </permission>
. . . . . .
    </permissions>
```

Each `resourceref` points to a resource ID. Create permissions using the actions available for the resource type specified for that ID, such as `create`, `read`, and `modify`.

See Resource Types and Corresponding Action Types the resource types and their available action types.

By default, all permissions are denied. If a certain permission is not explicitly given to a role, then the role does not have the permission. This approach ensures unauthorized users do not accidentally gain access to restricted resources.

Permissions are not hierarchical. That is, a `create` permission does not imply a `modify` permission or a `read` permission. All privileges are mutually exclusive. So, for example, if you want users to be able to modify some resources of a certain resource type, be sure to also give users the ability to view that resource type.

In TIBCO BusinessEvents Decision Manager, most TIBCO BusinessEvents project resources have only a `read` action type.

Resource Types and Corresponding Action Types

Permissions for a user role are defined using the action types available for each resource type.

Note: If a resource ID specifies a set of resources, the permission applies to that set of resources only. If it specifies a resource type, then the permission applies to all resources of that resource type.

Resource Types and Their Allowable Action Types

Resource Type	Allowable Action Types	(If action is ALLOW) Enables Users to. . .
`PROJECT`	`checkout`	Check out TIBCO BusinessEvents project resources. **Note:** Users can check out only those resources they are allowed to `read`.
	`update`	Update TIBCO BusinessEvents project resources that were checked out earlier.

Resource Type	Allowable Action Types	(If action is ALLOW) Enables Users to. . .
		Users can update only those resources they are allowed to `read`.
	`gen_deploy`	Use the Generate Deployable RMS menu option for building EAR files or class files.
	`commit`	Commit the modified/deleted TIBCO BusinessEvents project resources.
	`approval`	Review the worklist items in a project. (WebStudio only)
`CATALOGFUNCTION`	`invoke`	Invoke catalog functions in decision tables (`RULEFUNCTIONIMPL` resource type) that the users are allowed to `modify`.
`CHANNEL`	`read`	View channels.
`CONCEPT`	`read`	View concepts.
`DOMAIN`	`read`	View domain models.
	`create`	Create domain models.
`EVENT`	`read`	View events.
`PROPERTY`	`read`	View resource properties. If no resources are specified, then users can view properties of all resources that they are allowed to view (`read`).
`RULE`	`read`	View rules (rule source code).

Resource Type	Allowable Action Types	(If action is ALLOW) Enables Users to. . .
RULEFUNCTION	read	View rule functions (rule function source code).
	add_impl	Add decision tables (RULEFUNCTIONIMPL resource type). If specific rule functions are not listed, then users can add decision tables to all rule functions they are allowed to view (read). (Other permissions that apply to decision tables are set on the resources used in the decision table.)
	del_impl	Delete decision tables (RULEFUNCTIONIMPL resource type).
RULEFUNCTIONIMPL	read	View decision tables (RULEFUNCTIONIMPL resource type). Add columns in the Condition area of the decision table. Add rows and modify cells in existing rows and columns.
	modify	Add columns in the Action area of decision tables.
WSDL	read	View WSDL files.
XSD	read	View XSD files.
WebStudio Only Permissions		
RULETEMPLATE	read	Checkout rule templates.

Resource Type	Allowable Action Types	(If action is ALLOW) Enables Users to. . .
RULETEMPLATEINSTANCE	read	View business rules.
	add_inst	Create business rule for the rule template.
	del_inst	Delete business rule.
RULETEMPLATEVIEW	read	Checkout rule template views.

Permissions ALLOW and DENY

The value of the action element is one of the keywords ALLOW or DENY.

The value of the keyword determines whether the specified permission is given or denied.

DENY is the default value. You only need to set the DENY value explicitly when you have given ALLOW permissions at a higher level, and want to make individual exceptions within that broad scope.

The values ALLOW and DENY are case sensitive, so use uppercase letters only.

TIBCO Hawk Microagent Methods

To augment the monitoring and management functions in the TIBCO Administrator, the TIBCO BusinessEvents engine is instrumented with a TIBCO Hawk microagent that can be used to perform many administrative functions.

The provided methods have the following purpose:

- To enable the TIBCO Administrator to perform certain actions, for example, `GetExecInfo()`, `stopApplicationInstance()`, `getHostInformation()`
- To provide information about what is happening in the TIBCO BusinessEvents engine, for example, `getRules()`, `getDestinations()`, `getTotalNumberRulesFired()`
- To make certain changes in the TIBCO BusinessEvents engine without stopping it, for example, `activateRule()`, `reconnectChannels()`.

TIBCO BusinessEvents embeds a TIBCO Hawk microagent whose methods enable you to monitor and manage deployed TIBCO BusinessEvents applications. You can use TIBCO Hawk or the Hawk Console in TIBCO Administrator.

For more information, see:

- *TIBCO Administrator Server Configuration Guide* has more details on working with microagents and methods using TIBCO Administrator.
- *TIBCO Hawk Methods Reference* provides detailed documentation about TIBCO Hawk microagents and methods.

The provided methods are:

- activateRule()
- deactivateRule()
- execute()
- getChannels()
- getCacheRecoveryInfo()
- getDestinations()
- getLocksInfo()

- getCpuUsage()
- getThreadInfo()
- getEvent()
- GetExecInfo()
- getHostInformation()
- getInstance()
- GetLoggerNamesWithLevels()
- getMemoryUsage()
- getNumberOfEvents()
- getNumberOfInstances()
- getOMInfo()
- getRule()
- getRules()
- getScorecard()
- getScorecards()
- getSessionInputDestinations()
- getSessions()
- getStatus()
- getTotalNumberRulesFired()
- getTraceSinks()
- reconnectChannels()
- resetTotalNumberRulesFired()
- resumeChannels()
- resumeDestinations()
- resumeRuleServiceProvider()
- setLogLevel()
- startFileBasedProfiler()

- stopFileBasedProfiler()
- suspendRuleServiceProvider ()

Enabling the TIBCO Hawk Microagent

To use Hawk methods, enable the TIBCO Hawk microagent in the TIBCO BusinessEvents engine property file. You can also define the transport mode for the Hawk microagent in the property file. Alternatively, you can set properties in CDD to customize the TIBCO Hawk microagent as per cluster, processing unit, and agent.

Procedure

Open the `be-engine.tra` property file at *`BE_HOME`*`/bin/` or the project CDD for editing and add or set the following properties:

- To enable the Hawk microagent, set the property to true:

```
Hawk.Enabled = true
```

- Set the property value to the installation location of TIBCO Hawk:

```
tibco.env.HAWK_HOME
```

- If you are using non-default transport parameters for TIBCO Hawk, add the following properties for setting Hawk service, network, and daemon parameters:

```
Hawk.Service=9999
Hawk.Network=
Hawk.Daemon=
```

- To configure the Hawk transport, set the property to either `tibrv` or `tibtcp` transport mode.

  ```
  be.hawk.hma.transport
  ```

  The default value is `tibrv`.

  If you use `tibtcp` mode then set the following additional properties applicable to `tibtcp` transport mode.

Property	Description
`be.hawk.hma.tcp.self.url`	Set the IP address of an instance that is running the TIBCO BusinessEvents engine and a port number that is available: `<host_IP>:<port>` For example: `localhost:2556` **Note:** If multiple TIBCO BusinessEvents engines are running on the same machine then use separate ports for each engine. For example, if Cache1 binds to port 2571 then Inference1 can use port 2572.
`be.hawk.hma.tcp.agent.ami.url`	Set to the Hawk Agent IP and AMI session port: `<hawk_agent_IP>:<AMI_session_port>` For example: `localhost:2571`
`be.hawk.hma.tcp.key.store`	Keystore file

Property	Description
be.hawk.hma.tcp.trust.store	Trust store file
be.hawk.hma.tcp.key.store.password	Password for the keystore file
be.hawk.hma.tcp.key.password	Encrypted key password
be.hawk.hma.tcp.trust.store.password	Password for the trust store file **Note:** You can use the tibhawkpassword utility at <HAWK_HOME>/bin to encrypt the password. Use the password without double quotes. For more information, see the *TIBCO Hawk Installation, Configuration, and Administration Guide*.
be.hawk.hma.tcp.ssl.protocol	Protocol for a secure connection The default value is TLSv1.2.
be.hawk.hma.tcp.enabled.algorithms	Algorithms to be used for the security protocol The default value is TLS_RSA_WITH_AES_128_CBC_SHA.

Note: To enable TIBCO Hawk Console, set the following property to true in the *domain_name*.tra file at *TIBCO_Admin_HOME*/domain/*domain_name*/bin/tibcoadmin/:

```
hawk.console.enabled=true
```

activateRule()

Activates a RuleSet in the Session.

Type

ACTION

Parameters

Name	Description
`Session`	Name of the Session (optional).
`URI`	URI of the RuleSet.

Returns

Type	Description
`Session`	Name of the Session.
`URI`	URI of the RuleSet.
`Activated`	Returns true if the RuleSet is activated.

deactivateRule()

Deactivates a RuleSet in the Session

Type

ACTION

Parameters

Name	Description
`Session`	Name of the Session (optional)
`URI`	URI of the RuleSet

Returns

Type	Description
`Session`	Name of the Session.
`URI`	URI of the RuleSet.
`Deactivated`	Returns true if the RuleSet is deactivated.

execute()

Invokes the following list of special commands.

- printRuleNetwork
- getJoinTable
- workingMemoryDump
- startFileBasedRecorder
- stopsFileBasedRecorder

Type

ACTION_INFO

Parameters

Name	Description
`Command`	The special command to run.
`Parameters`	Parameters

The list of special commands with their parameters and return value is as follows:

Command	Description
`printRuleNetwork`	Displays Rete Network of a particular one or more Sessions. **Parameters** **Name** / **Description** `ruleSession Name` — Name of the RuleSession. **Returns**

Command	Description
	Type / **Description** `Line` — Row Number. `Name` — Name of the RuleSession. `Value` — Rete network object.
`getJoinTable`	Retrieves a Join table Id from one or more Sessions. **Note:** If the Join table Id does not exist, then the hawk encounters the exception that the Join table {Table Id} does not exist. **Parameters** **Name** / **Description** `Table Id` — ID of the Join table of a specific session. The initial value of the table id is int 0. **Returns**

Command	Description
	Type / **Description** `Line` — Row Number of the record. `Name` — Table Id of the Join table. `Value` — Objects in a specific row of the Join table.
`workingMemoryDump`	Displays a dump of the working memory objects for a particular one or more Sessions. **Parameters** **Name** / **Description** `ruleSession Name` — Name of the RuleSession. **Returns** **Type** / **Description** `Line` — Row Number.

Command	Description
	Type / **Description** `Name` — Name of the RuleSession. `Value` — Objects in a specific row.
`startFileBasedRecorder`	Starts recording and maintains logs in a file based on the mode of recording. **Parameters** **Name** / **Description** `Session Name` — Name of the RuleSession for which recording is done. `Directory` — The directory path is where the log of the session recording is maintained. `Mode` — The recording of the session is done in the following different modes:

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Recording Mode</th><th>Description</th></tr>
<tr><td>c</td><td>For record asserted.</td></tr>
<tr><td>u</td><td>For record modified.</td></tr>
<tr><td>d</td><td>For record retracted.</td></tr>
<tr><td>s</td><td>For the record is a scheduled time event.</td></tr>
<tr><td>r</td><td>For the record rule, fired.</td></tr>
<tr><td>a</td><td>For the record action run.</td></tr>
<tr><td>x</td><td>For the record event, expiry.</td></tr>
<tr><td>f</td><td>For the record function, run.</td></tr>
<tr><td>e</td><td>For the record event, expired.</td></tr>
</table>
</td></tr>
</table>
For example, <code>inference-class /home/myuser c</code>
</td></tr>
</table>

Command	Description
	Returns **Type** / **Description** `Line` — Row Number of the record. `Name` — Name of the RuleSession. `Value` — Objects in a specific row.
`stopsFileBasedRecorder`	Stops the recording and ends the log in the log file. **Parameters** **Name** / **Description** `ruleSession Name` — Name of the rule Session. **Returns** **Type** / **Description** `Line` — Row Number of the record.

Command	Description
	Type / **Description** `Name` — Name of the RuleSession. `Value` — Objects in a specific row.

getChannels()

Retrieves channel information.

Type

INFO

Parameters

Name	Description
`URI`	URI of the Channel (optional)

Returns

Type	Description
`Line`	Row Number

Type	Description
`URI`	URI of the Channel
`State`	Current state of the Channel

getCacheRecoveryInfo()

Retrieves the cache recovery information.

Timeout (milliseconds): 10000

Type

Open, Synchronous, IMPACT_INFO

Arguments

Name	Description
`Session`	Name of the Session

Returns

Name	Description
`Line`	Row number

Name	Description
`Session`	Name of the Session.
`NumberOfHandlesLoaded`	Number of Handles loaded in the session.
`NumberOfHandlesInError`	Number of Handles not loaded due to errors.
`NumberOfHandlesInStore`	Number of Handles in the underlying CacheStore.

getDestinations()

Retrieves Destination information.

Type

INFO

Parameters

Name	Description
`Channel URI`	URI of the Channel (optional).
`Destination Name`	Name of the Destination (optional).

Returns

Type	Description
`Line`	Row Number.
`Channel URI`	URI of the Channel.
`Destination URI`	URI of the Destination.
`Nb in`	Number of Events in.
`Rate in`	Rate of Events in.
`Nb out`	Number of Events out.
`Rate out`	Rate of Events out.

getLocksInfo()

Retrieves information about the Locks.

Timeout (milliseconds): 10000

Type

Open, Synchronous, IMPACT_INFO

Arguments

None

Returns

Name	Description
`Total Locks`	Total number of locks held.
`Local Locks`	Number of locks held at the local level.
`Cluster Locks`	Number of locks held at the cluster level.

getCpuUsage()

Retrieves engine CPU usage information.

Timeout (milliseconds): 10000

Type

Open, Synchronous, IMPACT_INFO

Arguments

None

Returns

Name	Description
`ProcessCpuTime`	The total CPU time used by a process in nanoseconds
`AvailableProcessors`	Number of available CPUs
`ProcessCpuLoad`	The current CPU load in the engine. The value is mentioned in the range of 0 to 1. The ProcessCpuLoad of 0.0 indicates that none of the CPU is running while value 1.0 indicates that all CPUs are actively running.

getThreadInfo()

Returns general thread information, execution information, and the synchronization statistics of a specific thread or all threads of a specific TIBCO BusinessEvents engine.

Timeout (milliseconds): 10000

Type

Open, Synchronous, IMPACT_INFO

Arguments

Name	Description
`Thread Name`	Name of the thread.

Returns

Name	Description
`Thread Name`	Name of the thread.
`Thread Id`	Id of the thread.
`Thread State`	State of the thread.
`User Time`	CPU time that the thread has run in user mode in nanoseconds.
`isInNative`	Whether the native code is running through the java Native Interface (JNI).
`Suspended`	Whether the thread is suspended.
`Blocked Count`	Returns the total number of times that the thread associated with this thread blocked to enter or re-enter a monitor.
`Blocked Time`	Returns the approximate accumulated elapsed time (in milliseconds) that the thread associated with this thread has blocked to enter or re-enter a monitor since thread contention.
`Lock Name`	Returns the string representation of the monitor lock that the thread associated with this thread is blocked to enter or waiting to be notified through object.wait.method.

Name	Description
`Lock Owner name`	Returns the name of the thread which holds the monitor lock of an object on which the thread associated with this thread is blocking.
`Lock Owner Id`	Returns the Id of the thread which holds the monitor lock of an object on which the thread associated with this thread is blocking.
`Waited Count`	Returns the total number of times that the thread associated with this thread waited for notification.
`Waited Time`	Returns the approximate accumulated elapsed time in milliseconds that the thread associated with this thread has waited for notification since thread contention monitoring.
`Uncaught Exception`	Uncaught thread exception.
`Stack Trace`	Stack trace of the thread.

getEvent()

Retrieves an Event from a Session.

Type

INFO

Parameters

Name	Description
`Session`	Name of the Session.
`Id`	ID of the Event.
`External`	Set true to use the event's external ID and false to use the internal ID.

Returns

Type	Description
`Line`	Row number.
`Session`	Name of the Session.
`Type`	Attribute or Property.
`Name`	Name of the Attribute or Property.
`Value`	Value of the Attribute or Property.

GetExecInfo()

Retrieves engine execution information.

Type

INFO

Parameters

No parameters.

Returns

Type	Description
`Status`	Engine status (ACTIVE, SUSPENDED, STANDBY or STOPPING).
`Uptime`	Elapsed time since RuleSessionProvider was started (milliseconds).
`Threads`	Number of RuleSessions in engine.
`Version`	Engine version.

getHostInformation()

Retrieves host information properties.

Type

INFO

Parameters

Name	Description
`Name`	Name of the host information property that you want to retrieve (optional).

Returns

Type	Description
`Name`	Property Name
`Value`	Property Value

getInstance()

Retrieves an Instance from the Session.

Type

INFO

Parameters

Name	Description
`Session`	Name of the Session

Name	Description
`Id`	ID of the Instance.
`External`	Set true to use the event's external ID and false to use the internal ID.

Returns

Type	Description
`Line`	Row Number.
`Session`	Name of the Session.
`Type`	Attribute or Property.
`Name`	Name of the Attribute or Property.
`Value`	Value of the Attribute or Property.

GetLoggerNamesWithLevels()

Retrieves the list of registered loggers with their current log level.

Type

INFO

Parameters

None

Returns

A MAP of the registered logger names with their current log level.

getMemoryUsage()

Retrieves the engine memory usage information.

Type

INFO

Parameters

None.

Returns

Type	Description
`Max`	Maximum memory size of the JVM, in bytes.
`Free`	Estimate of the free memory available to the JVM, in bytes.
`Used`	Estimate of the memory used in the JVM, in bytes.
`PercentUsed`	Estimate of the percentage of max memory used.

getNumberOfEvents()

Retrieves the total number of events existing in a Session.

Type

INFO

Parameters

Name	Description
`Session`	Name of the Session

Returns

Type	Description
`Line`	Row number.
`Session`	Name of the Session.
`Number of Events`	Total Number of Events.

getNumberOfInstances()

Retrieves the total number of instances existing in a Session.

Type

INFO

Parameters

Name	Description
`Session`	Name of the Session

Returns

Type	Description
`Line`	Row number.
`Session`	Name of the Session.
`Number of Instances`	Total Number of Instances.

getOMInfo()

Retrieves Object Store information of a Session.

Type

INFO

Parameters

Name	Description
`Session`	Name of the Session.

Returns

Type	Description
`Line`	Row number.
`Session`	Name of the Session
`Property`	Property name.
`Value`	Property value.

getRule()

Retrieves the Rules of a given RuleSet.

Type

INFO

Parameters

Name	Description
`Session`	Name of the Session.
`URI`	URI of the RuleSet.

Returns

Type	Description
`Line`	Row Number.
`Session`	Name of the Session.
`URI`	URI of the RuleSet.
`Rule`	Name of the Rule.
`Priority`	Priority of the rule.

getRules()

Retrieves Rulesets from the Session.

Type

INFO

Parameters

Name	Description
`Session`	Name of the Session.

Returns

Type	Description
`Line`	Row Number.
`Session`	Name of the Session.
`URI`	URI of the RuleSet.
`Activated`	Is the RuleSet activated.

getScorecard()

Retrieves a Scorecard of a Session.

Type

INFO

Parameters

Name	Description
`Session`	Name of the Session
`URI`	URI of the Scorecard.

Returns

Type	Description
`Line`	Row number.
`Session`	Name of the Session.
`Type`	Attribute or Property.
`Name`	Name of the Attribute or Property.
`Value`	Value of the Attribute or Property.

getScorecards()

Retrieves all the Scorecards of a Session.

Type

INFO

Parameters

Name	Description
`Session`	Name of the Session.

Returns

Type	Description
`Line`	Row Number.
`Session`	Name of the Session.
`Id`	ID of the Scorecard.
`External Id`	External ID of the Scorecard.
`Type`	Class of the Scorecard.

getSessionInputDestinations()

Retrieves destinations enabled for input.

Type

INFO

Parameters

Name	Description
`Session`	Name of the Session.

Returns

Type	Description
`Line`	Row number.
`Destination`	Destination URI.
`Preprocessor`	Destination preprocessor URI.

getSessions()

Retrieves session names.

Type

INFO

Parameters

No parameters.

Returns

Type	Description
`Line`	Row number.
`Session`	Name of the Session.

getStatus()

Retrieves basic status information about the engine.

Type

INFO

Parameters

No parameters.

Returns

Type	Description
`Instance ID`	Instance ID of the application.
`Application Name`	Name of the application.

Type	Description
`Uptime`	Time elapsed since startup.
`Process ID`	Process ID of the application.
`Host`	Name of the host machine on which this application is running.

getTotalNumberRulesFired()

Retrieves the total number of rules fired.

Type

INFO

Parameters

Name	Description
`Session`	Name of the Session

Returns

Type	Description
`Line`	Row Number.

Type	Description
`Session`	Name of the Session.
`Number of Rules Fired`	Total number of rules fired since the last reset.

getTraceSinks()

Retrieves information about trace sinks.

Type

INFO

Parameters

Name	Description
`Role Name`	Name of a Role (optional)
`Sink Name`	Name of a Sink (optional)

Returns

Type	Description
`Line`	Row Number.
`Instance ID`	Instance ID of the application.
`Application Name`	Name of the application.
`Sink Name`	Sink Name.
`Sink Type`	Sink Type (for example, fileSink, rvSink).
`Description`	Sink Description (for example, filename=file).
`Role`	Sink Role (for example, error, warn, debug).

reconnectChannels()

Restarts all channels or a single channel.

Type

ACTION

Parameters

Name	Description
`URI`	URI of the channel to restart (all channels are restarted if no URI is provided).

Returns

Nothing.

resetTotalNumberRulesFired()

Resets the total number of rules fired to zero.

Type

ACTION

Parameters

Name	Description
`Session`	Name of the Session

Returns

Returns nothing.

resumeChannels()

Resumes the channel.

Type

ACTION

Parameters

Name	Description
`URI`	URI of the Channel to resume (optional).

Returns

Returns nothing.

resumeDestinations()

Resumes Destinations.

Type

ACTION

Parameters

Name	Description
`Channel URI`	URI of the Channel that contains the Destination.
`Destination Name`	Name of the Destination (optional).

Returns

Returns nothing.

resumeRuleServiceProvider()

Resumes the RuleServiceProvider.

Type

ACTION

Parameters

Has no parameters.

Returns

nothing

setLogLevel()

Sets a specific log level for a specific logger. When setting the log level, the system runs through all the log level configurations and the last match supersedes all previous log level configurations. The wildcard character, an asterisk (*), can be used to select all or a pattern to match the logger names.

Type

ACTION

Parameters

Name	Description
`Name or Pattern`	Name of the logger or pattern to match the logger name.
`Log Level`	Sets the log level to one of the following: FATAL, ERROR, WARN, INFO, DEBUG, ALL, or OFF

Returns

Nothing.

SetLogLevel(Stringnameorpattern String Level)

This API can be used to set a specific log level for a specific logger, such as `SetLogLevel` ("`as.kit`", "`debug`") will set "`as.kit`" to debug whereas `SetLogLevel`("`as*`", "`debug`") sets all loggers starting with "`as`" to debug.

The system runs through all level configurations when setting the level and the last match supersedes previous configurations.

If you decide to invoke `SetLogLevel`("`as*`","`debug`") and later change to `SetLogLevel` ("`as.kit`","`info`"), then "`as.kit`" will be INFO.

Similarly, if you decide to invoke `SetLogLevel`("`as.kit`","`debug`") and later change to `SetLogLevel`("`as*`","`info`"), then "`as.kit`" will be INFO.

You can specify more than one family of loggers with different log levels via the CDD log configuration.

For example,

```
<roles>dashboard*:debug sql*:debug as*:info</roles>
```

The log configuration is processed left to right, and therefore

```
<roles>as*:info as.kit:debug</roles>
```

will set "as.kit" to debug and

```
<roles>as.kit:info as*:debug</roles>
```

will set "as.kit" to debug.

startFileBasedProfiler()

Turns on BusinessEvents Profiler and starts collecting data for a specified duration.

Type

ACTION

Parameters

Name	Description
`Session`	Name of the session.
`FileName`	Name of the output file where the profile data is collected.
`Level`	Sets the level of depth at which the profile data is collected. The levels can be as follows: • -1 for All level • 1 for only RTC level
`Duration`	Time duration in seconds that the profile data is collected. If <= 0, Profiler will be on until explicitly turned off.

Returns

Nothing.

Prints the log statement on the console.

stopApplicationInstance()

Shuts down the engine. All checkpoint files will be preserved and the engine's operating system process will exit.

Type

ACTION

Parameters

No parameters.

Returns

Returns nothing.

stopFileBasedProfiler()

Turns off the BusinessEvents Profiler and stops writing the profile data into the specified file when the Profiler turned on.

Type

ACTION

Parameters

Name	Description
`Session`	Name of the session.

Returns

Nothing.

Prints the log statement on the console.

suspendChannels()

Suspends channels.

Type

ACTION

Parameters

Name	Description
`URI`	URI of the Channel to suspend (optional).

Returns

Nothing.

suspendDestinations()

Suspends Destinations.

Type

ACTION

Parameters

Name	Description
`Channel URI`	URI of the Channel that contains the Destination.
`Destination Name`	Name of the Destination (optional).

Returns

Nothing.

suspendRuleServiceProvider ()

Suspends the RuleServiceProvider.

Type

ACTION

Parameters

None

Returns

Nothing

getInstanceDetails()

Retrieves the details of an instance.

Type

INFO

Parameters

None

Returns

Type	Description
`Application Name`	Name of the BusinessEvents application.
`Cluster Name`	Name of the cluster on which the BusinessEvents application is deployed.
`Application Instance ID`	Instance ID of the application.
`Application Instance Status`	Status of the application instance.
`Uptime`	Time duration for which the application is in a running state.
`Process ID`	Unique ID of the application process.

Type	Description
`Processing Unit`	Processing unit of the application on which it is deployed.
`Instance Type`	Attribute or property of the instance .
`Deployment Path`	Path where the application is deployed.
`Deployment Status`	Status of the deployed application.
`JMX Port`	JMX port of the instance.
`Used Memory`	Memory utilised by a specific instance.
`CPU Usage`	CPU utilised by a specific instance.
`BE_HOME`	Path to BE_HOME.

getInstanceMetrics()

Retrieves the metrics of an instance.

Type

IMPACT_INFO

Parameters

None

Returns

Type	Description
`Cluster Name`	Name of the cluster on which the BusinessEvents application is deployed.
`AppInstance ID`	Instance ID of the application.
`Used Memory`	Memory utilised by a specific instance.
`CPU Usage`	CPU utilised by a specific instance.
`TotalLocks`	Total number of locks held.
`LocalLocks`	Number of locks held at the local level.
`ClusterLocks`	Number of locks held at the cluster level.
`RtcTxnLatency`	Displays RTC transaction latency in an application.
`RtcTxnThroughput`	Displays RTC transaction throughput in an application.

Type	Description
`NumStaticRules`	Number of static rules.
`TotalRulesFired`	Total number of rules fired.
`TotalEventsProcessed`	Total number of events processed.

getEntityThroughput

Retrieves the throughput information of an entity.

Type

IMPACT_INFO

Parameters

None

Returns

Type	Description
`Cluster Name`	Name of the cluster on which the BusinessEvents application is deployed.
`Application Instance ID`	Instance ID of the application.

Type	Description
`DestURI`	URI of the Destination.
`Event Throughput`	Total number of events asserted in the inference engine.

getGlobalVariables()

Retrieves the global variables defined in the BusinessEvents application.

Type

INFO

Parameters

None

Returns

Type	Description
`Cluster Name`	Name of the cluster on which the BusinessEvents application is deployed.
`Application Instance ID`	Instance ID of the application.
`GvName`	Displays the name of the global variable.

Type	Description
`GvValue`	Displays the value of the global variable.

getBEAppProperties()

Retrieves the properties set in the CDD file of the BusinessEvents application.

Type

INFO

Parameters

None

Returns

Type	Description
`Cluster Name`	Name of the cluster on which the BusinessEvents application is deployed.
`Application Instance ID`	Instance ID of the application.
`PropName`	Name of the application property.
`PropValue`	Value of the application property.

getSystemJVMProperties()

Retrieves the server and the JVM properties set in the CDD file of the BusinessEvents application.

Type

INFO

Parameters

None

Returns

Type	Description
`Cluster Name`	Name of the cluster on which the BusinessEvents application is deployed.
`Application Instance ID`	Instance ID of the application.
`SysPropName`	Name of the system property.
`SysPropValue`	Value of the system property.

getRTCTransactionManagerMetrics()

Retrieves the metrics the RTC Transaction Manager of the BusinessEvents application.

Type

IMPACT_INFO

Parameters

None

Returns

Type	Description
`Cluster Name`	Name of the cluster on which the BusinessEvents application is deployed.
`AppInstance ID`	Instance ID of the application.
`TotalSuccessfulTxns`	Total number of the successful transactions.
`TotalTimeForSuccessfulTxns`	Total time taken for the successful transactions.
`TotalDBOprs`	Total number of database operations completed.
`TotalErrors`	Total number of errors.
`TotalCacheTxns`	Total number of cache transactions.
`PendingActions`	Number of pending actions.

Type	Description
`PendingCacheWrites`	Number of pending cache writes.
`PendingDBWrites`	Number of pending database writes.
`TotalDBQueueWaitTime`	Total database queue wait time.
`LocksToBeReleased`	Number of locks waiting to be released.
`EventsToBeAcknowledged`	Number of events pending to be acknowledged.

getInferenceAgentMetrics()

Retrieves the metrics of inference agents in a BusinessEvents application.

Type

IMPACT_INFO

Parameters

None

Returns

Type	Description
`Cluster Name`	Name of the cluster on which the BusinessEvents application is deployed.
`Application Instance ID`	Instance ID of the application.
`TotalTxnPublished`	Total number of transactions published.
`AvgTimeForPubTxns`	Average time for publishing the transactions.
`LocalCacheHitRatio`	Displays the Local Cache Hit ratio.

getEntityMetrics()

Retrieves the metrics of an entity.

Type

IMPACT_INFO

Parameters

None

Returns

Type	Description
`Entity Name`	Name of the entity in the BusinessEvents application.
`Cluster Name`	Name of the cluster on which the BusinessEvents application is deployed.
`Application Instance ID`	Instance ID of the application.
`EntityAvgTimeInRTC`	Entity average time in RTC.
`EntityAvgTimePostRTC`	Entity average time in post RTC.
`EntityAvgTimePreRTC`	Entity average time in pre RTC.
`TotalL1Hits`	Number of hits in L1 cache.
`TotalL1Misses`	Number of misses in L1 cache.

getEntityCacheMetrics()

Retrieves the metrics of an entity cache.

Type

IMPACT_INFO

Parameters

None

Returns

Type	Description
`Entity Name`	Name of the entity in the BusinessEvents application.
`Cluster Name`	Name of the cluster on which the BusinessEvents application is deployed.
`Application Instance ID`	Instance ID of the application.
`TotalPutOprns`	Total number of Put operations.
`AvgTimeForPutExec`	Average time to execute Put operation.
`TotalGetOprns`	Total number of Get operations.
`AvgTimeForGetExec`	Average time to execute Get operation.
`TotalRemoveOprns`	Total number of removal operations performed on the cache.
`AvgTimeForRemoveOprns`	Average time to execute removal operations.

TIBCO BusinessEvents JMX Methods

TIBCO BusinessEvents supports JMX-based management and monitoring of deployed applications by using the Mbean methods that are exposed through JMX. You can use tools such as JConsole to view these exposed MBean methods.

Note: JMX and JConsole are available in the JDK and not in the JRE provided with TIBCO BusinessEvents. A separate installation of the JDK is required.

The provided Mbean methods and operations are as follows:

- `com.tibco.be > Methods > Channels`
  - GetChannels()
  - GetDestinations()
  - GetSessionInputDestinations()
  - ReconnectChannels()
  - ResumeDestinations()
  - SuspendDestinations()
- `com.tibco.be > Methods > Engine`
  - GetHostInformation()
  - GetLoggerNamesWithLevels()
  - GetMemoryUsage()
  - GetNumberOfEvents()
  - GetNumberOfInstances()
  - GetVersionInfo()
  - SetLogLevel()
  - StopEngine()

- `com.tibco.be > Methods > WorkingMemory`
  - ActivateRule()
  - DeactivateRule()
  - GetRule()
  - GetRules()
  - GetRuleSessions()
  - GetTotalNumberRulesFired()
  - ResetTotalNumberRulesFired()

GetChannels()

Retrieves channel Information.

Type

INFO

Parameters

Name	Description
`URI`	URI of the channel (optional)

Returns

Type	Description
`Line`	Row number.
`Channel URI`	URI of the channel.
`State`	Current state of the channel.

GetDestinations()

Retrieves destination Information.

Type

INFO

Parameters

Name	Description
`Channel URI`	URI of the channel (optional)
`Destination Name`	Name of the destination (optional)

Returns

Type	Description
`Line`	Row number
`Channel URI`	URI of the channel
`Destination URI`	URI of the destination
`Num Events Received`	Number of events received at the destination
`Received Events Rate`	Received events rate at the destination
`Num Events Sent`	Number of events sent out from the destination

Type	Description
`Received Events Rate in last stats interval`	Received events rate at the destination in last stats interval
`Suspended`	Is the destination suspended

GetSessionInputDestinations()

Retrieves destinations enabled for input.

Type

INFO

Parameters

Name	Description
`Session`	Name of the session.

Returns

Type	Description
`Line`	Row number.
`Destination`	Destination URI.
`Preprocessor`	Destination preprocessor URI.

ReconnectChannels()

Restarts all channels or a single channel.

Type

ACTION

Parameters

Name	Description
`URI`	URI of the channel to restart (all channels are restarted if no URI is provided).

Returns

Nothing.

ResumeDestinations()

Resumes all destinations or a given destination.

Type

ACTION

Parameters

Name	Description
`Channel URI`	URI of the channel that contains the destination
`Destination Name`	Name of the destination (optional)

Returns

Nothing.

SuspendDestinations()

Suspends all destinations or a given destination.

Type

ACTION

Parameters

Name	Description
`Channel URI`	URI of the channel that contains the destination
`Destination Name`	Name of the destination (optional)

Returns

Nothing.

GetHostInformation()

Retrieves the host information properties.

Type

INFO

Parameters

Name	Description
Name	Name of the host information property that you want to retrieve (optional).

Returns

Type	Description
Name	Property name
Value	Property value

GetLoggerNamesWithLevels()

Retrieves the list of registered loggers with their current log level.

Type

INFO

Parameters

None

Returns

A MAP of the registered logger names with their current log level.

GetMemoryUsage()

Retrieves the engine memory usage information.

Type

INFO

Parameters

None.

Returns

Type	Description
`Max`	Maximum memory size of the JVM in bytes.
`Free`	Estimate of the free memory available to the JVM in bytes.
`Used`	Estimate of the memory used in the JVM in bytes.
`PercentUsed`	Estimate of the percentage of memory used.

GetNumberOfEvents()

Retrieves the total number of events existing in a session.

Type

INFO

Parameters

Name	Description
`Session`	Name of the session

Returns

Type	Description
`Line`	Row number
`Session`	Name of the session
`Number of Events`	Total number of events

GetNumberOfInstances()

Retrieves the total number of instances existing in a session.

Type

INFO

Parameters

Name	Description
`Session`	Name of the session

Returns

Type	Description
`Line`	Row number
`Session`	Name of the session
`Number of Instances`	Total number of Instances

GetVersionInfo()

Fetches the application name, version and TIBCO BusinessEvents engine version.

Type

INFO

Parameters

None.

Returns

Type	Description
`Application Name`	Name of an application
`Application Version`	Version of an application
`Engine Version`	Version of the TIBCO BusinessEvents engine

SetLogLevel()

Sets a specific log level for a specific logger. When setting the log level, the system runs through all the log level configurations and the last match supersedes all previous log level configurations. The wildcard character, an asterisk (*), can be used to select all or a pattern to match the logger names.

Type

ACTION

Parameters

Name	Description
`Name or Pattern`	Name of the logger or pattern to match the logger name
`Log Level`	Sets the log level to one of the following: FATAL, ERROR, WARN, INFO, DEBUG, ALL, or OFF

Returns

Nothing.

StopEngine()

Triggers the engine to shutdown.

Type

ACTION

Parameters

None

Returns

Nothing

ActivateRule()

Activates a RulesSet in a session.

Type

ACTION

Parameters

Name	Description
Session	Name of the session (optional)
URI	URI of the RulesSet

Returns

Type	Description
Session	Name of the session
URI	URI of the RulesSet
Activated	Returns true if the RuleSet is activated.

DeactivateRule()

Deactivates a RulesSet in a session.

Type

ACTION

Parameters

Name	Description
Session	Name of the session (optional)
URI	URI of the RulesSet

Returns

Type	Description
Session	Name of the session
URI	URI of the rulesset
Deactivated	Returns true if the RuleSet is deactivated.

GetRule()

Retrieves the rules of a given RulesSet.

Type

INFO

Parameters

Name	Description
Session	Name of the session
URI	URI of the RulesSet

Returns

Type	Description
`Internal ID`	Internal ID number of the rule
`Priority`	Priority of the rule
`Line`	Row number
`Session`	Name of the session
`URI`	URI of the RuleSet

GetRules()

Retrieves RulesRets from the session.

Type

INFO

Parameters

Name	Description
`Session`	Name of the session

Returns

Type	Description
`Activated`	Is the RulesSet activated

Type	Description
`Line`	Row number
`Rule Fired Count`	Total number of rules fired in the RulesSet
`Session`	Name of the session
`URI`	URI of the RulesSet

GetRuleSessions()

Fetches the rule session name(s) from the engine.

Type

INFO

Parameters

None.

Returns

Type	Description
`Line`	Row number
`Session`	Name of the session

GetTotalNumberRulesFired()

Retrieves the total number of rules fired.

Type

INFO

Parameters

Name	Description
Session	Name of the session

Returns

Type	Description
Line	Row number.
Session	Name of the session.
Number of Rules Fired	Total number of rules fired since the last reset.

ResetTotalNumberRulesFired()

Resets the total number of rules fired to zero.

Type

ACTION

Parameters

Name	Description
Session	Name of the session

Returns

Nothing.

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for this product is available on the TIBCO BusinessEvents® Enterprise Edition Documentation page.

To directly access documentation for this product, double-click the file at the following location:

TIBCO_HOME`/release_notes/TIB_businessevents-enterprise_6.3.1_docinfo.html`

where *TIBCO_HOME* is the top-level directory in which TIBCO products are installed. On Windows, the default *TIBCO_HOME* is `C:\tibco`. On UNIX systems, the default *TIBCO_HOME* is `/opt/tibco`.

Other TIBCO Product Documentation

When working with TIBCO BusinessEvents Enterprise Edition, you may find it useful to read the documentation of the following TIBCO products:

- TIBCO ActiveSpaces®: It is used as the cluster, cache, or store provider for the TIBCO BusinessEvents Enterprise Edition project.
- TIBCO FTL®: It is used as the cluster provider for the TIBCO BusinessEvents Enterprise Edition project.

How to Access Related Third-Party Documentation

When working with TIBCO BusinessEvents® Enterprise Edition, you may find it useful to read the documentation of the following third-party products:

- Apache Ignite
- Apache Kafka
- Confluent Kafka Schema Registry
- TIBCO Messaging - Schema Repository for Apache Kafka
- Apache Pulsar
- GridGain
- Apache Cassandra
- Grafana
- InfluxDB
- OpenTelemetry
- Control Plane
- Apache Maven

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to

gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

TIBCO, the TIBCO logo, the TIBCO O logo, TIBCO BusinessEvents, ActiveMatrix, ActiveMatrix BusinessWorks, ActiveSpaces, TIBCO Administrator, TIBCO Designer, Enterprise Message Service, TIBCO FTL, Hawk, and TIBCO Runtime Agent are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (https://www.cloud.com/legal) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: https://scripts.sil.org/OFL

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG 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. See the “readme” file for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at https://www.cloud.com/legal.