

TIBCO BusinessEvents® Administration

*Software Release 5.5
February 2018*

Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

ANY SOFTWARE ITEM IDENTIFIED AS THIRD PARTY LIBRARY IS AVAILABLE UNDER SEPARATE SOFTWARE LICENSE TERMS AND IS NOT PART OF A TIBCO PRODUCT. AS SUCH, THESE SOFTWARE ITEMS ARE NOT COVERED BY THE TERMS OF YOUR AGREEMENT WITH TIBCO, INCLUDING ANY TERMS CONCERNING SUPPORT, MAINTENANCE, WARRANTIES, AND INDEMNITIES. DOWNLOAD AND USE THESE ITEMS IS SOLELY AT YOUR OWN DISCRETION AND SUBJECT TO THE LICENSE TERMS APPLICABLE TO THEM. BY PROCEEDING TO DOWNLOAD, INSTALL OR USE ANY OF THESE ITEMS, YOU ACKNOWLEDGE THE FOREGOING DISTINCTIONS BETWEEN THESE ITEMS AND TIBCO PRODUCTS.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, The Power of Now, TIBCO ActiveMatrix, TIBCO ActiveMatrix BusinessWorks, TIBCO Administrator, TIBCO ActiveSpaces, TIBCO BusinessEvents, TIBCO Designer, TIBCO Enterprise Message Service, TIBCO Enterprise Administrator, TIBCO Enterprise Runtime for R, TIBCO FTL, TIBCO Hawk, TIBCO Live Datamart, TIBCO LiveView Web, TIBCO Runtime Agent, TIBCO Rendezvous, TIBCO StreamBase, and Two-Second Advantage are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Enterprise Java Beans (EJB), Java Platform Enterprise Edition (Java EE), Java 2 Platform Enterprise Edition (J2EE), and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation in the U.S. and 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.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION 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. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

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 "READ ME" FILES.

This product is covered by U.S. Patent No. 7,472,101.

Copyright © 2004-2018 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Figures	12
TIBCO Documentation and Support Services	13
Administration Overview	15
Engine Startup and Shutdown	17
Engine Startup Sequence	17
Engine Shutdown Sequence	18
Order of Precedence at Run time	18
Values Used to Establish the Engine Name	18
JVM-Level TRA File Configuration	20
Setting Property for Cache Based Object Management on AIX	20
Setting JMX Properties	21
TIBCO BusinessEvents Enterprise Administrator Agent	22
Starting BusinessEvents Enterprise Administrator Agent	23
Signing in to the TIBCO Enterprise Administrator Server	23
TIBCO BusinessEvents Enterprise Administrator Agent Monitoring	24
BusinessEvents Application Management	24
Machine Management	25
Viewing Machines List	26
Adding a Machine	26
Deleting a Machine	28
Editing Machine Details	29
Uploading External JAR Files	29
Auto-detecting BusinessEvents Installations on Host Machines	30
Viewing All Instances And BusinessEvents Installations on The Host Machine	30
Application Deployment Management	30
Creating a New Application Deployment	31
Importing Site Topology File	31
Exporting Application from TIBCO Administrator	33
Importing TIBCO Administrator Applications	33
Importing the BusinessEvents Enterprise Administrator Agent Application	34
Applying the Project Specific Master TRA File	35
Editing an Application Deployment	35
Deleting an Application Deployment	36
Exporting Application from TIBCO BusinessEvents Enterprise Administrator Agent	37
Hot-Deploying an BusinessEvents Application	37
Deployment Views	37

Processing Unit Instances Management	38
Viewing the Instance Management Page	38
Creating an Instance	39
Updating an Instance	41
Copying an Instance	41
Deleting an Instance	41
Deploying Processing Unit Instances	42
Undeploying Processing Unit Instances	42
Starting Processing Unit Instances	42
Stopping Processing Unit Instances	43
Killing an Instance	43
Invoking Instance Operations	43
Downloading Thread Dumps for Instances	44
Downloading Log Files for Instances	44
Viewing Log File for Instance	45
Hot-Deploying an Instance	45
Hot-Deploying Classes and Rule Template Instances	45
Processing Unit Instance Configuration	46
Configuring Global Variables of an Instance	47
Configuring System Properties of an Instance	47
Overriding the Value of a System Property	48
Deleting the Override of a System Property	48
Adding a New System Property for an Instance	49
Removing a System Property for an Instance	49
Configuring BusinessEvents Properties of an Instance	50
Overriding the Effective CDD Value of a BusinessEvents Property	50
Removing a BusinessEvents Property	50
Adding a New BusinessEvents Property	51
Deleting the Override of a BusinessEvents Property	51
Configuring JVM Properties of an Instance	52
Configuring Log Levels of an Instance	53
Disabling Jetty Server Logs	53
Configuring Properties for Multiple Instances	54
Instance Monitoring Charts	54
Configuring Destinations for Event Throughput Chart	56
Rules and Alerts	57
Creating a Alert Rule	57
Monitored Entities Reference	60
Alert Tokens Reference	63

Deployment Profiles	64
Adding an Application Deployment Profile	64
Editing an Application Deployment Profile	65
Deleting an Application Deployment Profile	65
Creating a Duplicate Deployment Profile	65
User Management	66
Roles and Permissions Reference	66
BusinessEvents Enterprise Administrator Agent Configuration Reference	71
I18n Support	75
Localizing BusinessEvents Enterprise Administrator Agent Messages	75
Command-line Interface	76
TIBCO BusinessEvents Enterprise Administrator Agent Commands Reference	77
Authentication and SSL Configurations	82
Configuring JMX Authentication	83
Configuring One-way SSL between Administrator Agent and Processing Unit Instance	83
Enabling SSL for The BusinessEvents Enterprise Administrator Agent Monitoring Page	84
Basic MM Configuration	85
MM Runtime Architecture	85
Software for Remote Start and Deployment	86
SSH	86
TIBCO Hawk	87
PsTools	87
TIBCO Hawk Configuration for Machine Level Metrics	87
Configuring TIBCO Hawk	88
JMX Properties and To-Be-Monitored Engine TRA Files	88
JMX Properties Configuration	88
Enabling Monitoring and Management	89
Enabling JMX MBeans Authentication	89
JMX Remote Port Number Setup at Run time	89
User Authorization for Administrator and User Roles	89
Site Topology	90
Configuring the Site Topology	91
Project, Master and Deployed Locations of CDD and EAR Files	91
Deployment-Specific Processing Units and Global Variables	92
Site Topology in TIBCO BusinessEvents Studio	92
Adding a Site Topology Diagram	92
Configuring the Site Topology	93
Site Topology Files for the MM Server	94
Site Topology Reference	94

Basic MM Settings in MM.cdd	99
Importing the emonitor Project for CDD Editing	99
Configuring the Basic Settings in the MM.cdd File	100
MM Agent Basic Configuration Reference	100
Broker Properties for Working with Coherence Cache Provider	105
Broker Properties Reference	105
Coherence WKA Cluster Discovery	106
Configuring the Project's CDD to Communicate with the Cluster	107
Configuring the MM.CDD File	107
Configuring the be-engine.tra Files for Hosts with Multiple NIC Cards	107
TIBCO BusinessEvents DataGrid WKA Discovery	108
Configuring the Project's CDD for Cluster Management	108
MM Console Properties Reference	108
MM Metrics and Features Configuration	110
Configuring Alerts	110
Alert Configuration Reference	111
Path to an Alert Metric Value (and a Reference Value)	112
Specifying the Alert Message	114
Pane Types Reference for Alert Configuration	114
Health Metric Rules	116
Cluster Member Paths	116
Two Types of Thresholds	117
Health Metric Rule Examples	118
Examples Using Alerts	118
Health Metric Rules Configuration	119
Setting Up the Health Metric Rule	119
Configuring a Health Metric Rule with the Child Member Health status	120
Configuring a Health Metric Using Cluster Member Alerts	120
Health Metric Rule Configuration Reference	121
Action Configuration	123
Configuring an Action	124
Action Configuration Reference	124
Deployment and Management of Engines with MM	126
Starting the MM Server	126
Logging On to MM Console	126
Setting Global Variables in MM	127
Engines with MM	127
Deploying Cluster Engines in MM Console	127
Hot Deployment for Engines with MM	128

Remote Engines (PUs) and the MM-tools Utility	128
Configuring the mm-tools.tra File	129
Using Public Private Key Authentication with mm-tools	129
Example Commands for Authentication with mm-tools	130
Deploying Starting or Stopping a Remote Engine	130
mm-tools Utility Options Reference	130
Monitoring and Management Component (MM) for TIBCO BusinessEvents Cluster	132
Cluster Explorer Nodes	133
Members of the TIBCO BusinessEvents Cluster	133
Cluster Explorer	134
Managing Engines	135
Purge Inactive Unpredefined Processes	135
Viewing Monitored Objects	135
Executing a Method	135
Thread Analyzer Reports	136
Generating Thread Analyzer Reports	136
Panels and Panes	136
MM Metric Panes	137
Cluster Overview	138
Cluster Level Metrics	138
System Alerts Pane	139
Machine Overview	139
Machine Level Metrics	140
Process Overview	140
Process Level Metrics	141
Agent Overview	142
Agent Reference	142
Inference Agent Overview	144
Inference Agent Reference	144
Query Agent Overview	145
Query Agent Reference	145
Ontology (Cache Objects) Overview	146
Ontology Reference	147
MM Process Methods	147
MM Inference Agent Methods	150
MM Query Agent Methods	151
Enterprise Archive (EAR) Files	152
Building an EAR File in TIBCO BusinessEvents Studio	152
Enterprise Archive Reference	153

Building an EAR File at the Command Line	154
Options for Building an EAR File	154
Engine Management at the Command Line	156
Command Line Startup Option Reference	156
Supplementary Property Files	157
Setting up TIBCO BusinessEvents Engine as a Windows NT Service	157
Deployment with TIBCO Administrator	160
Deploying a Project in a TIBCO Administrator Domain	160
Other Deployment Tasks	161
Overriding of Global Variables in TIBCO Administrator	161
Project Deployment	162
Deploying a Project EAR in a TIBCO Administrator Domain	163
Deploying a Project EAR for the First Time	163
Deploying a Project EAR for an Existing Application	164
Deploying on a Service Level	164
Deploying on an Instance Level	165
Hot Deployment	167
Modifications Allowed in Hot Deployment	167
Enabling Hot Deployment	168
Hot Deployment in a TIBCO Administrator Domain	169
Performing Hot Deployment in a TIBCO Administrator Domain	169
Performing Hot Deployment Outside a TIBCO Administrator Domain	170
User Authentication	171
Authentication Options	171
Authentication Configuration	172
Enabling Authentication and Selecting Authentication Type	172
Configuring File-Based Authentication	173
Authentication Property Reference for the TRA File	173
Common Authentication Properties for the CDD File	174
LDAP Authentication Properties for the CDD File	174
Access Control Configuration	177
Guidelines for Configuring Access Control	177
Structure of the Access Control File	178
Permissions—ALLOW and DENY	178
Access Control Files	179
Specification and Grouping of Project Resources	179
Permissions Definition	180
Resource Types and Corresponding Action Types	180
Dockerize TIBCO BusinessEvents	183

Key Docker Concepts	183
Running TIBCO BusinessEvents in Docker	183
Dockerfile for TIBCO BusinessEvents	184
Building a Docker Image for TIBCO BusinessEvents®	186
(Linux Only) Building a Lightweight Docker Image for TIBCO BusinessEvents®	187
Generating BusinessEvents Application Dockerfile	188
BusinessEvents Docker Utility Features	189
Building BusinessEvents Application Docker Image	190
Running TIBCO BusinessEvents® Application in Docker	191
Running BusinessEvents Rule Management Server (RMS) in Docker	193
Setting up BusinessEvents Multihost Clustering on Amazon EC2 Instances Using Docker	194
Setting up Standalone Amazon EC2 Instances	194
Configuring Amazon RDS for Shared All Persistence	196
Configuring Amazon EFS for Shared Nothing Persistence	197
Running TIBCO BusinessEvents® on AWS Based Kubernetes Cluster	198
Setting up a Kubernetes Cluster on AWS	199
Deploying TIBCO BusinessEvents® Cluster for No Backing Store on AWS	200
Deploying TIBCO BusinessEvents® Cluster for Shared Nothing Storage on AWS	201
Deploying BusinessEvents Cluster for Shared All Storage on AWS	204
Sample Kubernetes Resource Files for Shared All Storage	205
Running RMS Applications in Kubernetes	206
Sample Kubernetes Resource Files for RMS	209
TIBCO Hawk Microagent Methods	211
Enabling the TIBCO Hawk Microagent	212
activateRule()	212
deactivateRule()	213
execute()	213
getChannels()	214
getCacheRecoveryInfo()	214
getDestinations()	215
getEvent()	216
GetExecInfo()	216
getHostInformation()	217
getInstance()	217
getJoinTable	218
GetLoggerNamesWithLevels()	218
getMemoryUsage()	218
getNumberOfEvents()	219
getNumberOfInstances()	219

getOMInfo()	220
getRule()	220
getRules()	221
getScorecard()	221
getScorecards()	222
getSessionInputDestinations()	222
getSessions()	223
getStatus()	223
getTotalNumberRulesFired()	224
getTraceSinks()	224
reconnectChannels()	225
resetTotalNumberRulesFired()	225
resumeChannels()	226
resumeDestinations()	226
resumeRuleServiceProvider()	226
setLogLevel()	227
SetLogLevel(Stringnameorpattern String Level)	227
startFileBasedProfiler()	227
stopApplicationInstance()	228
stopFileBasedProfiler()	228
suspendChannels()	228
suspendDestinations()	228
suspendRuleServiceProvider ()	229

Figures

TIBCO BusinessEvents and TIBCO Enterprise Administrator Integration Architecture	22
TIBCO Enterprise Administrator Landing Page	25
BusinessEvents Enterprise Administrator Agent Start Page	25
BusinessEvents Enterprise Administrator Agent Machine Management Page	26
All Instances View	38
Monitoring Charts for the Instance	55
MM Runtime Architecture	85
Summary of Site Topology Configuration	90
Locations for the CDD and EAR Files	91
Active and Inactive Nodes in Cluster Explorer	133
Cluster Overview Panel	137
Metric Gallery	137
Restore Metric	138
Cluster Overview Pane	138
Metric Gallery	139

TIBCO Documentation and Support Services

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website, mainly in HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product. To access the latest documentation, visit <https://docs.tibco.com>.

Product-Specific Documentation

Documentation for TIBCO products is not bundled with the software. Instead, it is available on the TIBCO Documentation site. To directly access documentation for this product, double-click the following file:

`TIBCO_HOME/release_notes/TIB_businessesvents-standard_version_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`.

The following documents for this product can be found in the TIBCO Documentation site:

- *TIBCO BusinessEvents Installation*
- *TIBCO BusinessEvents Getting Started*
- *TIBCO BusinessEvents Architect's Guide*
- *TIBCO BusinessEvents Developer's Guide*
- *TIBCO BusinessEvents Configuration Guide*
- *TIBCO BusinessEvents WebStudio User's Guide*
- *TIBCO BusinessEvents Administration*
- Online References:
 - *TIBCO BusinessEvents Java API Reference*
 - *TIBCO BusinessEvents Functions Reference*
- *TIBCO BusinessEvents Release Notes*

How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

- For an overview of TIBCO Support, visit <http://www.tibco.com/services/support>.
- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support portal at <https://support.tibco.com>.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to <https://support.tibco.com>. If you do not have a user name, 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 <https://community.tibco.com>.

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's 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](#)
 - If you use the TIBCO BusinessEvents Monitoring and Management (MM) component, first configure it to work with your cluster. Two kinds of configuration are documented:
- Basic configuration is about connecting with that cluster, including defining the site topology file for cluster to be monitored. The Site Topology file configures the processing units and agents for deployment in Deployment Units (DUs) to hosts. See [Basic MM Configuration](#).
- Operational configuration is performed to suit your needs, for example to set up health level metric thresholds, alerts, and actions. See [MM Metrics and Features 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 Rendezvous, 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](#).
- Using TIBCO BusinessEvents Monitoring and Management options. See [Deployment and Management of Engines with MM](#).
- At the command-line. See [Building an EAR File at the Command Line](#).
- To a TIBCO Administrator domain. See [Deploy a Project in a TIBCO Administrator Domain](#).



For details about deploying TIBCO BusinessEvents Decision Manager classes (implemented virtual rule functions) see *TIBCO BusinessEvents Decision Manager User's 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's 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](#).

TIBCO BusinessEvents Monitoring and Management also allows hot deployment. See [Hot Deployment for Engines with MM](#) for information pertinent to Monitoring and Management (MM).

Management and Monitoring

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

- [Monitoring and Management Component \(MM\) for TIBCO BusinessEvents Cluster](#).
- Certain topics in [Project Deployment in a TIBCO Administrator Domain](#), and [TIBCO Hawk Microagent Methods](#).

Authentication and Authorization

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

- [User Authentication](#)
- [Configuring Access Control for a Project](#)

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.



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 Cache Plus Memory objects if any. 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 up for outbound traffic. Inbound listeners do not start yet.
5. Scorecards are created.
6. Startup functions execute (for example, they initialize values of scorecards).
7. The first RTC cycle occurs and all rule actions that are eligible to execute now execute. (Scorecards and startup rule functions can cause rules to be eligible to execute. Depending on the state of entities recovered from the backing store, the RTC will take more or less time.) See *TIBCO BusinessEvents Architect's 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 executes, 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.



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 execute.
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).



Global variables set in the CDD file are ignored if you deploy using TIBCO Administrator. They are overridden by variables set in TIBCO BusinessEvents Monitoring and Management.

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.

- For deployment using MM, the name specified in the **Processing Unit Configuration Name** field in the site topology file. See [Site Topology](#).
- 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's 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`.

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, TIBCO Rendezvous, 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's Guide*.

For additional information about system configuration, see the following:

- For TIBCO Enterprise Message Service and TIBCO Rendezvous Channels:

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.



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`.

- Local installation is the only option for TIBCO Rendezvous, which is not a pure Java API.

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's Guide*.

Setting Property for Cache Based Object Management on AIX

A specific property must be added to all TRA files when TIBCO BusinessEvents is installed on AIX and uses cache-based object management.

Procedure

1. Add the following property: `java.net.preferIPv4Stack`

2. Set the property value to `true`:

```
java.net.preferIPv4Stack=true
```

3. If you do not add the property, you see the following exception:

```
java.net.SocketException: The socket name is not available on this systemI
```



Remember to set this property on all internal TIBCO BusinessEvents engines TRA files too, such as in `be-mm.tra` for the TIBCO BusinessEvents Monitoring and Management (MM) server and the MM broker properties set in the MM CDD file. Add-on products also have engine TRA files you must update.

Setting JMX Properties

JMX properties are set for various purposes.

Procedure

1. Set JMX properties for the MM component to monitor the cluster.

Configure various JMX settings as described in [JMX Properties and To-Be-Monitored Engine TRA Files](#).

2. Set JMX properties for other purposes.

Using a JMX-compliant monitoring tool such as JConsole can be useful for other purposes. For example, MBeans enables you to see cache details if you are using Coherence as the cache provider.

3. 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
```

4. You can also set the JMX connector port for deployment with TIBCO Administrator using this CDD property:

```
be.engine.jmx.connector.port
```

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 shipped with TIBCO BusinessEvents Enterprise Administrator Agent that can be used to administer, manage, and monitor BusinessEvents applications.

TIBCO BusinessEvents Enterprise Administrator Agent is the back-end 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).

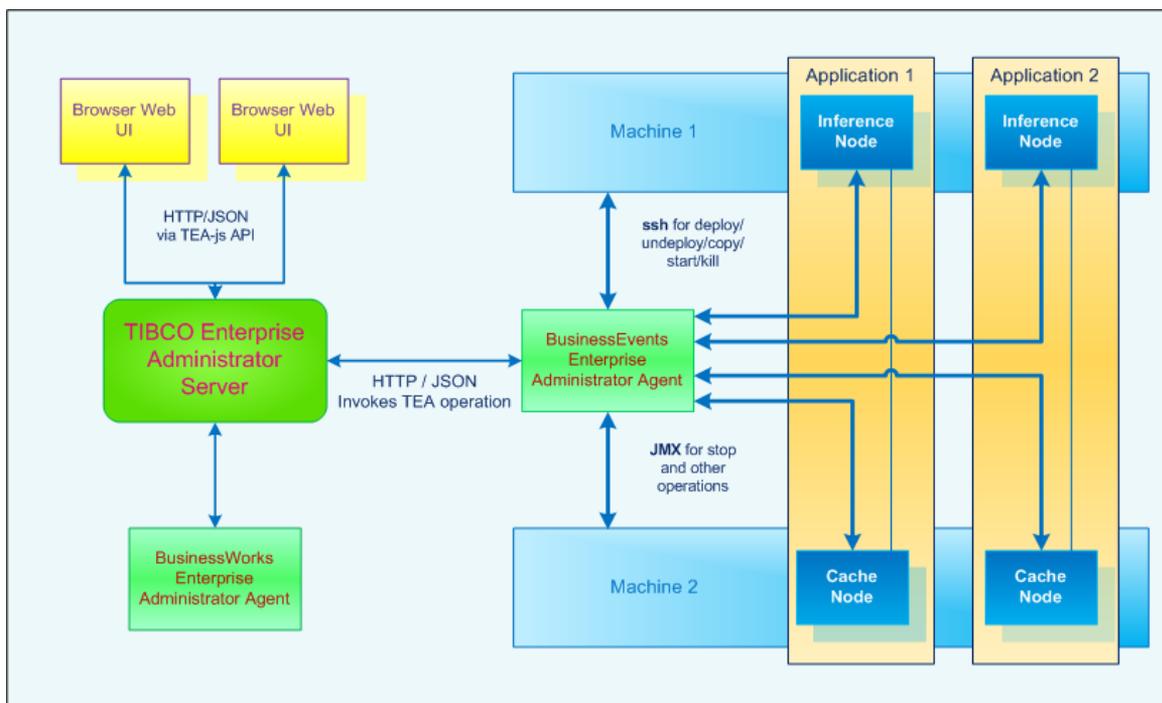
In BusinessEvents Monitoring and Management you use different components for BusinessEvents application deployment management. While using BusinessEvents Enterprise Administrator Agent, you can monitor and manage the BusinessEvents application deployment using a single dashboard. Thus BusinessEvents Enterprise Administrator Agent provides you 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 provides product specific functionality, such as, deployments, 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, thereby providing a common dashboard across products to end users. The following diagram shows the architecture for BusinessEvents integration with TIBCO Enterprise Administrator.

TIBCO BusinessEvents and TIBCO Enterprise Administrator Integration Architecture



Starting BusinessEvents Enterprise Administrator Agent

Start the BusinessEvents Enterprise Administrator Agent to use the monitoring and management capabilities of TIBCO Enterprise Administrator.

Prerequisites

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

Procedure

1. Navigate to `BE_HOME\teagent\bin`.
2. Run `be-teagent.exe` to start the BusinessEvents Enterprise Administrator Agent.



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`.

On Windows, you can also start the BusinessEvents Enterprise Administrator Agent by using the Start menu:

Start > All Programs > TIBCO > <TIBCO_HOME> > TIBCO BusinessEvents <version> > Start Enterprise Administrator Agent

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

Signing in to the TIBCO Enterprise Administrator Server

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

Prerequisites

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.



The default port number and other settings can be changed by modifying the settings in `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 user name 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.



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.

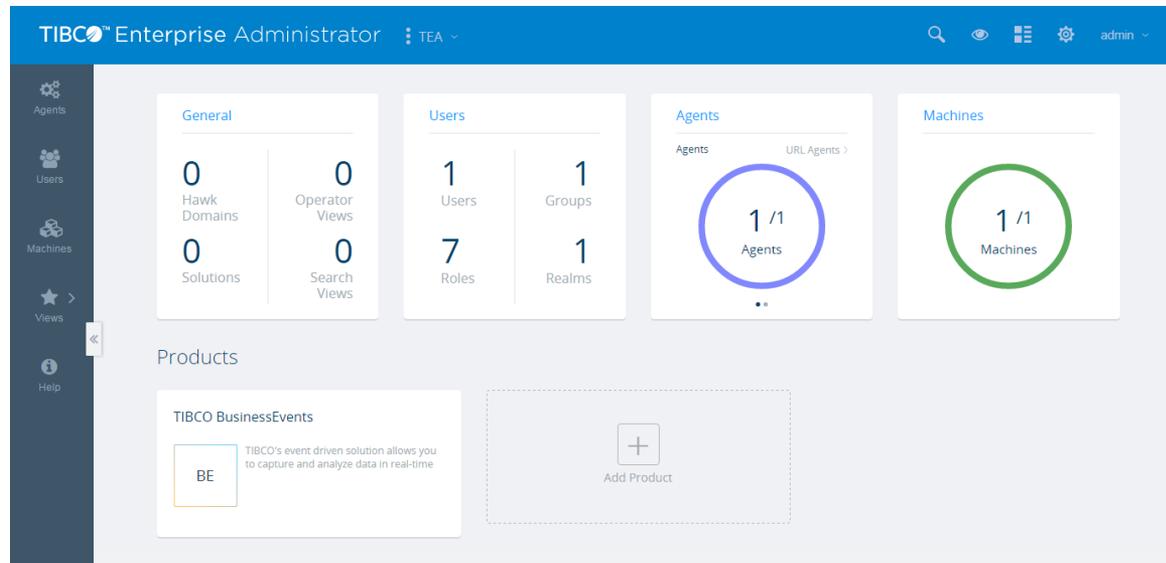
TIBCO Enterprise Administrator Landing Page

On successful authentication, the landing page is displayed. The user name 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.



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](#).

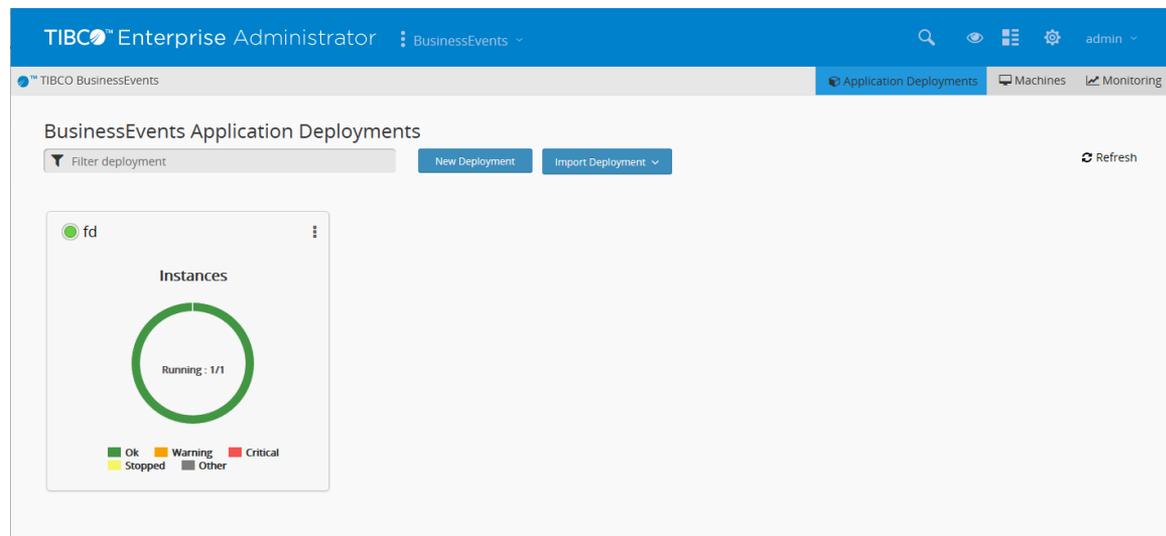
TIBCO Enterprise Administrator Landing Page



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.

BusinessEvents Enterprise Administrator Agent Start Page



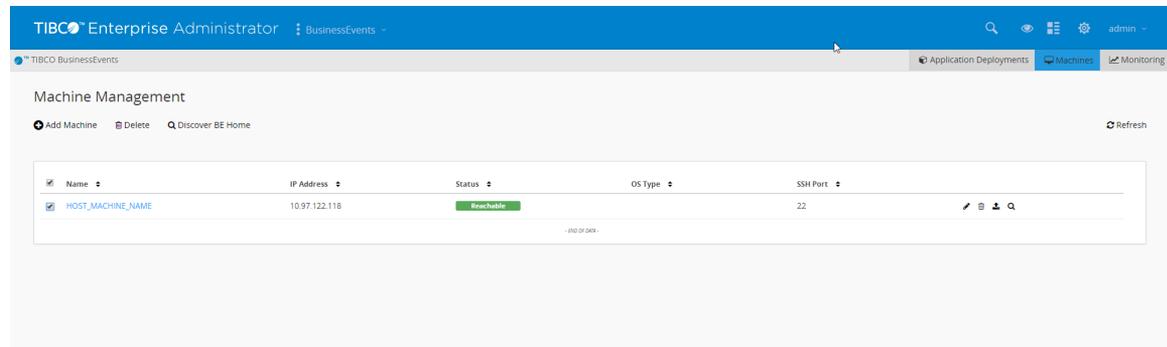
Machine Management

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 machines added after importing BusinessEvents Monitoring and Management site topology (.st) files or those which are detected by the TIBCO Enterprise Administrator server, are available for different application deployments.

BusinessEvents Enterprise Administrator Agent Machine Management Page



In the Machine Management page, you can upload an external JAR file to a machine if required for your project, see [Uploading External JAR Files](#) on page 29. 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](#) on page 30. 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](#) on page 30.

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

- 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

Field	Description
Add BE Home	<p>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:</p> <ul style="list-style-type: none"> • BE HOME: The path to the BusinessEvents installation (for example, C:\tibco\be\<i>version</i>) • 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\<i>version</i>\bin\be-engine.tra • BE Version: The version of the TIBCO BusinessEvents software installed at the specified BE HOME. <p>You can also remove a BusinessEvents installation using Remove icon, if that BE HOME is not used in any instance.</p>
Discover BE Installations	<p>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.</p>

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 [Machine Properties](#) on page 27 for details on the machine properties.
The updated machine details are displayed in the machine lists on the Machine Management page.

Uploading External JAR Files

Your application might require some external JAR files to run successfully. In BusinessEvents Enterprise Administrator agent you can upload the external JAR files 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 JAR files.
 - Click the machine name to open the Machine Details page. In the Machine Details page, click **Upload External Jars**.
3. Select the **BE Home** to which you want to upload the external JAR files.
4. In the **JAR File/s** field, click **Browse** to browse the required JAR file and click **Open**.



For uploading the multiple JAR files, compress those JAR files into an archive and then upload that archive.

5. Click **Finish** to upload the JAR file to the selected **BE Home**.
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 check boxes 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](#) on page 30.

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 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 list of instances grouped by the application deployments on the machine. You can expand a application deployment to view all its instances. Click on a instance name to see details of that instance. You can also select check boxes of multiple instances and **Start**, **Stop**, or **Kill** those instances.
 - **BE Homes** - The **BE Homes** tab displays 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.

Importing Site Topology File

Using the import option, you can import application and instances details along with the associated machine details using an existing Site Topology (.st) file used by BusinessEvents Monitoring and Management.

The site topology file contains information about the processing unit instances and associated machines that can be imported in the BusinessEvents Enterprise Administrator Agent. If the machine, specified in the ST file, is already present in the machine repository (matching all machine parameters), no new machine entry is created. Otherwise, the machine is added to the machine repository. If a machine already exists in the machine repository with the same name as mentioned in the site topology file, but other details are different, the agent auto-generates a unique name.

Prerequisites

BusinessEvents application site topology file which contains information about processing unit instances and associated machine 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 MM Deployment**.
The Import Deployment page is displayed to enter the deployment details.
3. Enter the details in the Import Deployment page and click **Save**.

Field	Description
Site Topology	Click Browse and select the site topology file for the BusinessEvents application. For example, C:\tibco\be\5.3\examples\standard\FraudDetectionCache\FraudDetectionCache\FraudDetectionCache.st for the FraudDetectionCache application.
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 name specified in the ST file. For example, FraudDetection for the FraudDetectionCache example.

A new card for the BusinessEvents application (for example, `FraudDetection`) is displayed on the BusinessEvents Application Deployments page. The import confirmation message is displayed:
`<Application deployment> application deployment import successful.`

Result

The import option has now added all the processing unit instances mentioned in the site topology file to the BusinessEvents Enterprise Administrator Agent. The machine details from the site topology file are added to the machine repository. The imported machine entry is also bounded to the respective processing unit instances.



The concept of "Deployment Unit" of BusinessEvents Monitoring and Management is not used in BusinessEvents Enterprise Administrator Agent.

Exporting Application from TIBCO Administrator

For migration from TIBCO Administrator to TIBCO Enterprise Administrator, the application needs to be exported from TIBCO Administrator in a 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, 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 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: 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 user name
- 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.



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 a archive file.

Importing TIBCO Administrator Applications

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

Prerequisites

Ensure that the deployment that is exported from TIBCO Administrator is in an archive file. See [Exporting Application from TIBCO Administrator](#) on page 33 for more details.



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 check box 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.

Prerequisites

The application archive, with the application CDD file, EAR file, and configuration XML file, exported from the BusinessEvents Enterprise Administrator agent. See [Exporting Application from TIBCO BusinessEvents Enterprise Administrator Agent](#) on page 37 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. If 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.

Prerequisites

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 a 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 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.

All Instances View

The screenshot shows the 'All Instances' view in the TIBCO Enterprise Administrator. The top navigation bar includes 'TIBCO Enterprise Administrator', 'BusinessEvents', and user 'admin'. The breadcrumb trail is 'TIBCO BusinessEvents > FraudDetection > Application Deployments > Machines > Monitoring'. The 'All Instances' view is active, showing a list of instances. The instance details for 'CacheServer' and 'InferenceAgent' are visible. The 'CacheServer' instance has a status of 'Needs Deployment' and is running on a machine with JMX Port 5501. The 'InferenceAgent' instance also has a status of 'Needs Deployment' and is running on a machine with JMX Port 5500.

Name	Status	UpTime	Machine	JMX Port	Processing Unit	Process Id	Deployment Status	CPU Usage	Memory
CacheServer	Needs Deployment		HOST_MACHINE_NAME	5501	cache		Needs Deployment		
InferenceAgent	Needs Deployment		HOST_MACHINE_NAME	5500	default		Needs Deployment		

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.

Prerequisites

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 a new 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 drop-down is populated with the processing units defined in the application CDD file.
Machine Name	Select the machine on which you want to deploy. The drop-down lists all the machines registered in the machine repository. See Machine Management .
BE Home	Select the <i>BE_HOME</i> from list of all the <i>BE_HOME</i> for the selected Machine Name . The list contains all <i>BE_HOME</i> of the BusinessEvents version which matches with the application BusinessEvents version. The BusinessEvents engine is started from the selected <i>BE_HOME</i> .

Field	Description
JMX Port	<p>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 new port used is 5507. You can change the port number if you don't want to use the suggested port.</p>
JMX User Name	<p>Optional. Specify the JMX username for user authentication for the JMX connections from the BusinessEvents Enterprise Administrator Agent to the instance.</p> <p>Additional configuration are required to activate the JMX authentication. See Authentication and SSL Configurations for more details.</p>
JMX Password	<p>Optional. Specify the password for the JMX username specified in the JMX User Name.</p> <p>Additional configuration are required to activate the JMX authentication. See Authentication and SSL Configurations for more details.</p>
Deployment Path	<p>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 batch files to start the instance.</p> <p>The default path is the Default Deployment Path from the associated machine.</p>

The instance is 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.

Prerequisites

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.



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.



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.

Prerequisites

Navigate to any of the deployment views for instance management. See [Viewing the Instance Management Page](#) for more details.

Procedure

1. Select the check box 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.

Prerequisites

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.



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, 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.

Prerequisites

Navigate to any of the deployment views for instance management. See [Viewing the Instance Management Page](#) for more details.

Procedure

- Select the check boxes for all the instances you want to deploy and 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, 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.

Prerequisites

Navigate to any of the deployment views for instance management. See [Viewing the Instance Management Page](#) for more details.

Procedure

- Select the check boxes for all the instances you want to undeploy and 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.

Prerequisites

Navigate to any of the deployment views for instance management. See [Viewing the Instance Management Page](#) for more details.

Procedure

- Select the check boxes for all the deployed instances you want to start and 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.

Prerequisites

Navigate to any of the deployment views for instance management. See [Viewing the Instance Management Page](#) for more details.

Procedure

- Select the check boxes for all the running instances you want to stop and 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 kill processing unit instances that were previously running (uses kill -9 to kill instances on Unix).

Prerequisites

Navigate to any of the deployment views for instance management. See [Viewing the Instance Management Page](#) for more details.

Procedure

- Select the check boxes for all the running instances you want to forcefully stop and 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.

Prerequisites

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.

Prerequisites

Navigate to any of the deployment views for instance management. See [Viewing the Instance Management Page](#) for more details.

Procedure

1. Select the check boxes 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.

Prerequisites

Navigate to any of the deployment views for instance management. See [Viewing the Instance Management Page](#) for more details.

Procedure

1. Select the check boxes for instances for which you want to download log files.
2. Click **BE Logs** or **AS Logs** to download BusinessEvents logs and 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 an instance.

Prerequisites

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 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.

Prerequisites

Navigate to any of the deployment views for instance management. See [Viewing the Instance Management Page](#) for more details.

Procedure

1. Select the check box for the instance 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.

Prerequisites

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.



These options are enabled only if the `be.engine.cluster.externalClasses.path` property (for decision table) and the `be.cluster.ruletemplateinstances.deploy.dir` property (for 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.



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.

Prerequisites

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](#).
- Remove a previously added system property. See [Removing a System Property](#).

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.

Prerequisites

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.

Prerequisites

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

- 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.

Prerequisites

Navigate to any of the deployment views for instance management. See [Viewing the Instance Management Page](#) for more details.

Procedure

- 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.
- In the **Configuration** tab, select **System Properties**.
All the system properties for the instance are listed in the page with their default values.
- Click **Add New** to add a new property.
A new entry is added to the list with empty Name and Value.
- 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.

Prerequisites

Navigate to any of the deployment views for instance management. See [Viewing the Instance Management Page](#) for more details.

Procedure

- 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.
- In the **Configuration** tab, select **System Properties**.
All the system properties for the instance are listed in the page with their default value.
- 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.
- 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.

Prerequisites

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.

Prerequisites

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.

Prerequisites

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.

Prerequisites

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 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.

Prerequisites

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.

Prerequisites

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.

Prerequisites

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 there 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** check box, 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.

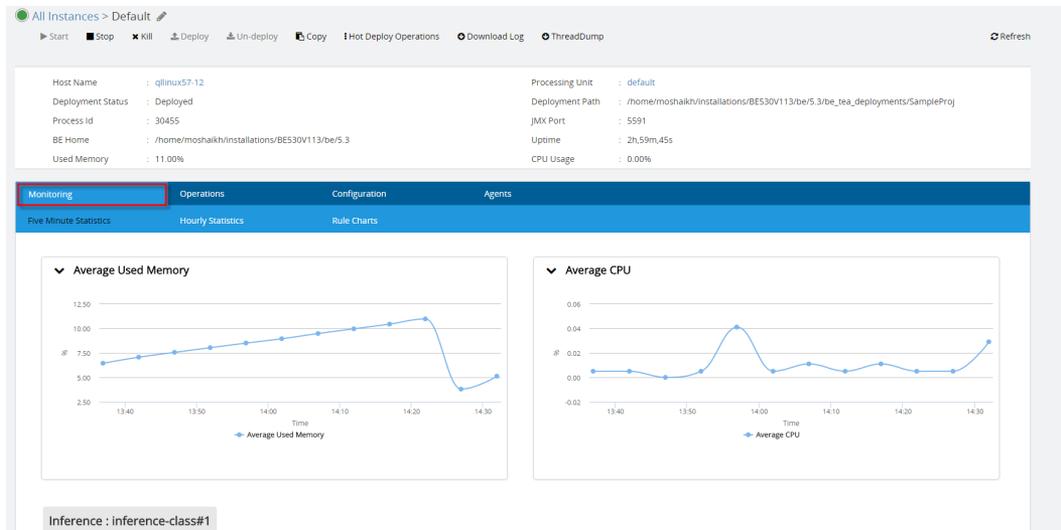
Instance Monitoring Charts

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

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 local, total, 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.

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>
    <.....>
  </chart>
</section>
```



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](#) 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="DESTINATION_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.



- 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 a 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 some action if the condition is met. However, you must specify a clear condition and associated clear action as well so that you can undo the earlier action.

Procedure

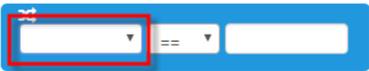
1. In the Rules view, click **Create New Rule**.
The Create Rule wizard is displayed.

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

Field	Description
Name	The unique name of the rule. The agent displays an error if the rule with the same name already exists for the application.
Monitored Entity	The entities for which metrics are computed. The values are: <ul style="list-style-type: none"> • 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 drop-down 	The metrics of the monitored entity selected earlier. See Monitored Entities Reference for more details on the metrics for all monitored entities.
Center drop-down 	The comparison operators for the condition. The values are: <ul style="list-style-type: none"> • == • >= • > • <= • <
Right text box 	The value of the metrics for creating the set condition.
Expression 	The option to add more condition 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, click **Save**.

Action Type	Description
Set-Health-Action	<p>Sets the health of the instance if the condition is satisfied. You need special privilege to create health action, see Rules and Alerts for more details. The fields for the health action are:</p> <ul style="list-style-type: none"> • Alert Level - The level of the alert. The values are: <ul style="list-style-type: none"> – High – Medium – Normal – Low • HealthValue - The health status of the instance to be set if the condition is satisfied. The values are: <ul style="list-style-type: none"> – Critical – Warning – Normal • Alert Text - The text which 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	<p>Sends 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:</p> <ul style="list-style-type: none"> • Alert Level - The level of the alert. The values are: <ul style="list-style-type: none"> – High – Medium – Normal – Low • To - The email address of the primary recipient of the email • Cc - The email address of recipient to which email is copied • Bcc - The email address of recipient to which email is blind copied • Subject - The subject of the email • Body - The main body of the email
Log-Action	<p>Logs the alert text to the logs, if the set condition is satisfied. The fields of the log action are:</p> <ul style="list-style-type: none"> • Alert Level - Level of the alert. The values are: <ul style="list-style-type: none"> – High – Medium – Normal

Action Type	Description
	<ul style="list-style-type: none"> - Low • Alert Text - The text which is displayed in the alert. You can use alert tokens insert dynamic values in alert text. See Alert Tokens Reference for list of alert tokens available in the BusinessEvents Enterprise Administrator Agent.
Alert-Only	<p>Generates only the alert and takes no other action. The fields are:</p> <ul style="list-style-type: none"> • Alert Level - Level of the alert. The values are: <ul style="list-style-type: none"> - 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 action 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 step 5](#)), 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 have 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 which 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 have 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 running or stopped state.
Thread Count	Count of the running threads grouped in five minute.
Deadlocked Thread Count	Count of the deadlocked threads grouped in five minute.
Processing Unit	The associated processing unit name.
Processing Unit Name	The associated instance name.

RTC Transactions

The monitored entity have the metrics for the RTC transactions executed in the inference engine.

RTC Transactions Metrics

Metric	Description
Pending Locks	The count of pending locks to release grouped in five minutes
RTC Transaction Count	The number of RTC Transactions grouped in five minute
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 executed in the inference engine.

Event Throughput Metrics

Metric	Description
Event Throughput	Total number of events asserted in the inference engine in five minute interval.
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 have the metrics for the BusinessEvents rules executed 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
<code>\${agent.name}</code>	The name of the agent
<code>\${alert.priority}</code>	The priority of the alert created by the associated action
<code>\${alert.timestamp}</code>	The time at which the action was triggered
<code>\${alert.type}</code>	The type of the condition (set or clear) which triggered the action
<code>\${application.name}</code>	The application deployment name
<code>\${average.cpu.usage}</code>	The average CPU usage percentage, averaged over five minutes
<code>\${average.memory.usage}</code>	The average memory usage percentage that is evaluated over max allocated memory, and averaged over five minutes.
<code>\${berule.exec.time}</code>	The average rule execution time in milliseconds
<code>\${berule.name}</code>	The BusinessEvents rule URI
<code>\${deadlocked.threadcount}</code>	The count of the deadlocked threads grouped in five minute
<code>\${destination}</code>	The URI of the destination where the event was received
<code>\${entity.health}</code>	The health of the rule entity at the set or clear action of the rule
<code>\${event.throughput}</code>	The value of the event throughput in milliseconds.
<code>\${instance.name}</code>	The associated instance name
<code>\${isrunning}</code>	If the instance is in running or stopped state
<code>\${pending.locks}</code>	The count of pending locks to release, grouped in five minutes.
<code>\${percent.critical}</code>	The percentage of processing units in the <code>Critical</code> state of the total deployed processing units in the cluster

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

Deployment Profiles

If you want to deploy an application in different environments using different set of properties, you can use deployment profile. A deployment profile is a collection of global variables, system properties, and BusinessEvents properties. You can associate a deployment profile to an application. An application can have multiple deployment profiles based on environment but only one can be selected as active profile.

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

Changes made in profiles are not reflected for global variables, system properties, and BusinessEvents properties under configuration tab or vice versa.

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 **Profiles** view, click **Add Profile**.
The Add Application Profile wizard is displayed.
2. 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.
3. 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](#) on page 46.

4. 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 **Profiles** view, click the **Edit** icon  for the profile which you want to edit.
The Edit Application Profile wizard is displayed.
2. 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.
3. 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 **Profiles** view, click the **Delete** icon  for the profile which you want to delete.
The Delete Application Profile confirmation dialog is displayed.
2. 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.

Prerequisites

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

Procedure

1. In the Profiles view, select the deployment profile from where you want to copy the variables and click **Copy Profile**.
2. 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](#) on page 65 for more details.

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's 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 [Default Roles and Permissions of BusinessEvents Enterprise Administrator Agent](#) 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's Guide* for adding roles and permissions.

Default Roles and Permissions of BusinessEvents Enterprise Administrator Agent

Permissions	APP_ MANAGE R	DEPLOYE R	OPERAT OR	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 Delete a host	Yes	Yes	No	No	No	No
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

Permissions	APP_ MANAGE R	DEPLOYE R	OPERAT OR	RULE_ AUTHOR	RULE_ AUTHOR - ADMIN	VIEW_ ALL
DELETE_INSTANCE_ PERMISSION Delete a processing unit instance	Yes	Yes	No	No	No	No
START_PU_INSTANC E_PERMISSION Start a processing unit instance	Yes	Yes	Yes	No	No	No
STOP_PU_INSTANCE _PERMISSION Stop a processing unit instance	Yes	Yes	Yes	No	No	No
KILL_INSTANCE_PE RMISSION Kill a processing unit instance	Yes	Yes	Yes	No	No	No
HOT_DEPLOY_PERMI SSION Hot Deploy	Yes	Yes	No	No	No	No
COPY_INSTANCE_PE RMISSION Copy a processing unit instance	Yes	Yes	No	No	No	No
UPDATE_GV_VAR_PE RMISSION Update Global Variables	Yes	Yes	No	No	No	No
UPDATE_SYSTEM_PR OPS_PERMISSION Update System Properties	Yes	Yes	No	No	No	No
UPDATE_JVM_PROPS _PERMISSION Update JVM Property	Yes	Yes	No	No	No	No

Permissions	APP_ MANAGE R	DEPLOYE R	OPERAT OR	RULE_ AUTHOR	RULE_ AUTHOR - ADMIN	VIEW_ ALL
UPDATE_LOG_LEVEL _PERMISSION Update Log Level	Yes	Yes	No	No	No	No
SUSPEND_AGENT_PE RMISSION Suspend running BusinessEvents agent	Yes	Yes	Yes	No	No	No
RESUME_AGENT_PER MISSION 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_INSTANC E_PERMISSION Un-deploy a processing unit instance	Yes	Yes	No	No	No	No
UPLOAD_TRA_PERMI SSION Upload the TRA file	Yes	Yes	No	No	No	No
DEPLOY_CLASSES_P ERMISSION Deploy the classes	Yes	Yes	No	No	No	No
UPLOAD_CLASSES_P ERMISSION Upload the classes	Yes	Yes	No	No	No	No
DEPLOY_RULE_TEMP LATE_PERMISSION Deploy rule template instance	Yes	Yes	No	No	No	No

Permissions	APP_ MANAGE R	DEPLOYE R	OPERAT OR	RULE_ AUTHOR	RULE_ AUTHOR - ADMIN	VIEW_ ALL
CREATE_RULE_PERM SSION Create an alert rule	No	No	No	Yes	Yes	No
UPDATE_RULE_PERM SSION Update an alert rule	No	No	No	Yes	Yes	No
DELETE_RULE_PERM SSION Delete an alert rule	No	No	No	Yes	Yes	No
GET_RULES_PERMIS SION View all alert rules	No	No	No	Yes	Yes	No
RULE_ADMIN_PERMI SSION Create health rules	No	No	No	No	Yes	No
CLEAR_ALERTS_PER MISSION Clear alerts	No	No	No	Yes	Yes	No
GET_ALERTS_PERMI SSION View alerts	No	No	No	Yes	Yes	No
read Read permission	Yes	Yes	Yes	Yes	Yes	Yes
UPDATE_BE_PROPS_ PERMISSION Update BusinessEvents property	Yes	Yes	No	No	No	No
ADD_PROFILE_PERM SSION Add a new deployment profile	Yes	Yes	No	No	No	No

Permissions	APP_ MANAGE R	DEPLOYE R	OPERAT OR	RULE_ AUTHOR	RULE_ AUTHOR ADMIN	VIEW_ ALL
UPDATE_PROFILE_P ERMISSION Update the existing deployment profile	Yes	Yes	No	No	No	No
DELETE_PROFILE_P ERMISSION Delete the existing deployment profile	Yes	Yes	No	No	No	No
SET_DEFAULT_PROF ILE_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`.

General Configurations

General Configuration Properties

Property	Description
<code>be.tea.server.url</code>	Specifies TIBCO Enterprise Administrator server URL The default value is <code>http://localhost:8777/tea</code> .
<code>be.tea.agent.port</code>	Specifies TIBCO BusinessEvents Enterprise Administrator Agent listening port The default value is 9777.
<code>be.tea.agent.retry.interval</code>	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.

Property	Description
be.tea.agent.resource.base	<p>Specifies path to the directory that contains TIBCO BusinessEvents Enterprise Administrator Agent web resources</p> <p>The default value is <i>BE_HOME/teagent</i>.</p>
be.tea.agent.application.datastore	<p>Specifies path to the directory that contains the TIBCO BusinessEvents Enterprise Administrator Agent configuration datastore</p> <p>The default value is <i>BE_HOME/teagent/config/repo</i>.</p>
be.tea.agent.rta.config.plugin.dir	<p>Do not change. Specifies path to the XML file which contains monitoring metric collector plugins configurations</p> <p>The default value is <i>BE_HOME/teagent/config/plugins</i>.</p>
be.tea.agent.metrics.view.config.file	<p>Do not change. Specifies path to the XML file that contains monitoring views configurations</p> <p>The default value is <i>BE_HOME/teagent/config/EntityMetricViewConfig.xml</i>.</p>
be.tea.agent.view.beentity.config.file	<p>Do not change. Specifies path to the XML file that contains BusinessEvents entity monitoring configurations</p> <p>The default value is <i>BE_HOME/teagent/config/BeEntityMap.xml</i>.</p>
be.tea.agent.metrics.rules.attr.map.file	<p>Do not change. Specifies path to the XML file that contains rules attribute and entity mapping configurations</p> <p>The default value is <i>BE_HOME/teagent/config/RuleEntityAttrMap.xml</i>.</p>
be.tea.agent.schema.store	<p>Do not change. Specifies path to the directory from where the server loads the schema files</p> <p>The default value is <i>BE_HOME/teagent/config/schema</i>.</p>
be.tea.agent.jmx.port	<p>Specifies the port for JMX connection. The default value is 5566.</p>
be.tea.agent.jmx.usesingleport	<p>Optional. Specifies whether single port is used for the JMX connection. The default value is <i>true</i>.</p>
be.tea.agent.jmx.usessl	<p>Optional. Specifies whether SSL is enabled for the JMX connection. The default value is <i>false</i>.</p>

Property	Description
<code>be.tea.agent.poller.delay</code>	Optional. Specifies the application instances status poller interval (in milliseconds) The default value is 30000.
<code>be.tea.agent.enable.gc.charts</code>	Specifies whether to enable additional views (Garbage collector and Memory Pool chart) The default value is <code>false</code> .
<code>be.tea.agent.worker.thread.count</code>	Optional. Specifies the maximum number of worker threads for thread pools The default value is 16.
<code>be.tea.agent.worker.thread.count.min</code>	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 <code>be.tea.agent.worker.thread.count</code> . The default value is 1.
<code>be.tea.agent.worker.thread.idle.timeout</code>	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 <code>be.tea.agent.worker.thread.count.min</code> . The default value is 300.
<code>be.tea.agent.worker.queue.size</code>	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.
<code>be.tea.agent.metric.compute.thread.count</code>	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.
<code>be.tea.agent.metric.compute.queue.size</code>	Optional. Specifies the job queue size for the metric computation jobs thread pool. Ensure that you set the properties value twice the value of <code>be.tea.agent.metric.worker.thread.count</code> . The default value is 64.

Property	Description
<code>be.tea.agent.rules.actions.scan.frequency</code>	Optional. Specifies the interval (in milliseconds) before scanning for rule actions The default value is 5000.

Log Action Configurations

Log Action Configuration Properties

Property	Description
<code>be.tea.agent.log.alert.format</code>	Specifies the format (XML or TEXT) in which the alert text is logged for the rule. The default value is TEXT

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

SSH Configurations

You can provide the SSH connection-related configuration.

SSH Configuration Properties

Property	Description
<code>be.tea.agent.ssh.privatekey.file</code>	Specifies the path to the private key file for password-less SSH authentication
<code>be.tea.agent.ssh.privatekey.passphrase</code>	Specifies passphrase to the private key file (if required) The default value is 3.
<code>be.tea.agent.ssh.connection.timeout</code>	Specifies the SSH connection timeout (in milliseconds) The default value is 30000.

Localization Configurations

You can specify configurations related to your local language.

Localization Configuration Properties

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

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 using the BusinessEvents Enterprise Administrator Agent message file `messages_en_US.properties`. The message file contains BusinessEvents Enterprise Administrator agent success and error messages. Messages are specified for each identifier separated by equal sign (=). You can replace these messages with messages in your language.

BusinessEvents support translations for the following languages:

- Arabic
- French
- German
- Italian
- Korean
- Simplified Chinese

You can also download the language packs for BusinessEvents from <https://edelivery.tibco.com>. These language pack contains message files of the supported language.

Procedure

1. Create a copy of the existing message file `BE_HOME\teagent\config\messages\messages_en_US.properties` and rename the new file as `messages_<language-code>_<country-code>.properties`.
The `<language-code>` is a two-digit code defined in ISO 639-1 and `<country-code>` a two-digit code defined in ISO 3166-1. The `<country-code>` is optional.
2. Open the newly created file `messages_<language-code>_<country-code>.properties` for editing.
3. In the `messages_<language-code>_<country-code>.properties` file, replace the english messages with their translation in the target language, and save the file.
4. Open `BE_HOME\teagent\config\be-teagent.props` file for editing and specify the newly created message file path as the value of the `be.tea.agent.message.file` property.
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
```
5. 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 pickle` 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 execute the configuration and application management commands for BusinessEvents Enterprise Administrator Agent in the Python-based command-line interface. The python script files are located at `BE_HOME\teagent\cli\python`.

Configuration Management Commands (`configurationMgmt.py`)

Without SSL

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

```
python configurationMgmt.py -t SERVERURL -u USERNAME -p USERPWD commandname
commandparameters
```

With SSL

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

```
python configurationMgmt.py -ssl true -t SERVERURL -u USERNAME -p USERPWD -sc
SERVER_CERTIFICATE_PATH -cc CLIENT_CERTIFICATE_PATH commandname commandparameters
```

editMachine

Command to edit the machine details:

```
editmachine -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 -m MACHINENAME
```

discoverBEHomes

Command to discover BusinessEvents installation on the machine:

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

uploadExternalJars

Command to upload external JAR file:

```
uploadexternaljars -m MACHINENAME -b BEHOME -z JARFILES
```

editDeployment

Command to edit an application deployment:

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

editInstance

Command to edit an application instance:

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

saveGlobalVariable

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

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

saveSystemProperty

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

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

saveJVMProperty

Command to update JVM property for one or multiple instances:

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

saveBEProperty

Command to update BusinessEvents property for one or multiple instances:

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

Application Management Commands (`applicationsMgmt.py`)

Without SSL

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

```
python applicationsMgmt.py -t SERVERURL -u USERNAME -p USERPWD commandname commandparameters
```

With SSL

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

```
python applicationsMgmt.py -ssl true -t SERVERURL -u USERNAME -p USERPWD -sc SERVER_CERTIFICATE_PATH -cc CLIENT_CERTIFICATE_PATH commandname commandparameters
```

addMachine

Command to add a new machine:

```
addmachine -m MACHINENAME -i IPADDRESS -o {"windows", "unix", "os-x"} -b BEHOME -t BETRA -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 -d APPLICATIONNAME -c CDDFILE -e EARFILE
```

importDeployment

Command to import an application using the specified CDD, EAR, and site topology files:

```
importdeployment -d APPLICATIONNAME -c CDDFILE -e EARFILE -s=STFILE
```

deleteDeployment

Command to delete the application:

```
deletedeployment -d APPLICATIONNAME
```

createInstance

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

```
createinstance -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 policy in the CDD file:

```
copyinstance -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 -d APPLICATIONNAME -i INSTANCENAME
```

deploy

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

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

undeploy

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

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

start

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

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

hotdeploy

Command to hotdeploy an application provided by the EAR file:

```
hotdeploy -d APPLICATIONNAME -e EARFILE
```

downloadLogs

Command to download logs for multiple instances:

```
downloadlogs -d APPLICATIONNAME -l DOWNLOADLOCATION -lt LOGTYPE(BE | AS | TD) [-i [INSTANCES [INSTANCES ...]]]
```

exportTeaDeployment

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

```
exportteadeployment -d APPLICATIONNAME -l DOWNLOADLOCATION
```

importTeaDeployment

Command to import a application deployment:

```
importteadeployment -z ZIPFILE
```

hotDeployDtRt

Command to hot deploy decision table and business rule:

```
hotdeploydtrt -d APPLICATIONNAME -i INSTANCENAME -o DEPLOYTYPE (DT | RT) -z ZIPFILE
```

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

Without SSL

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

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

With SSL

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

```
python adminToAgentMigration.py -ssl true -t SERVERURL -u USERNAME -p USERPWD -sc
SERVER_CERTIFICATE_PATH -cc CLIENT_CERTIFICATE_PATH commandname commandparameters
```

migrateapplications

Command to migrate application, that is exported from TIBCO Administrator, to TIBCO BusinessEvents Enterprise Administrator agent. See [Exporting Application from TIBCO Administrator](#) on page 33 for more details.

```
migrateapplications -z EXPORTED_ZIP_FILE_PATH
```

migrateuserandroles

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

```
migrateuserandroles -x EXPORTED_XML_FILE_PATH
```

Arguments for Commands

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

Parameters	Description
[]	Identifies an optional parameter
-ssl	(Optional) Enable SSL for the python script execution. The values are: <ul style="list-style-type: none"> true false The default value is false.
-sc SERVER_CERTIFICATE_PATH	Path of the server SSL certificate.
-cc CLIENT_CERTIFICATE_PATH	Path of the client SSL certificate.
-t SERVERURL	The TIBCO Enterprise Administrator server URL If no value is provided then default value (<code>http://localhost:8777</code>) is used.
-u USERNAME	TIBCO Enterprise Administrator server user name

Parameters	Description
-p USERPWD	TIBCO Enterprise Administrator server password
-m MACHINENAME	The current machine name
-n NEWMACHINENAME	The new machine name after changing 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.
-p PWD	The password for the user name 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
-i INSTANCENAME	The application instance name
[-i [INSTANCES [INSTANCES ...]]]	Identifies multiple instance names. For example, -i instance1 ins2 ins3.
-u PU	Processing unit name
-p JMXPORT	JMX port number
-ju JMXUSER	JMX user name
-jp JMXPASS	JMX password
-n VARNAME	Global variable name.
-v VARVALUE	Global variable value
-n PROPNAME	System, BusinessEvents, or JVM property name
-v PROPVALUE	System, BusinessEvents, or JVM property value

Parameters	Description
-x EXPORTED_XML_FILE_PATH	The location of the XML file of the application exported from TIBCO Administrator.
-z EXPORTED_ZIP_FILE_PATH	The location of 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 DEPLOYTYPE	Type of artifact to be hot deployed. The values are: <ul style="list-style-type: none"> • DT - Decision table • RT - Business rule (rule template instance)
-l DOWNLOADLOCATION	Download location of log files.
-lt LOGTYPE	Type of the logs to be downloaded. The values are: <ul style="list-style-type: none"> • BE - BusinessEvents logs • AS - ActiveSpaces logs • TD - Thread dumps

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 *TIBCO Enterprise Administrator User's Guide* for more details on configuration for 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 *TIBCO Enterprise Administrator User's Guide* for more details on configuration for SSL authentication between the TIBCO Enterprise Administrator server and TIBCO BusinessEvents Enterprise Administrator Agent.

Add the BusinessEvents Enterprise Administrator Agent side properties to `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 BusinessEvents Enterprise Administrator Agent and BusinessEvents 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 of 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 user name 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>`



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 of 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.

Basic MM Configuration

TIBCO BusinessEvents Monitoring and Management (MM) component has to be configured for use with a deployed TIBCO BusinessEvents cluster.

The tasks are arranged in a reasonable order, but the specified order is not required for many of them:

- The MM server cluster uses cache-based object management, but MM can monitor TIBCO BusinessEvents engines running in In-Memory mode too.
- TIBCO BusinessEvents MM allows you to monitor multiple clusters. Each cluster is configured using its own site topology file. The cluster names must be unique.
- For monitoring machine-level metrics, TIBCO Hawk is required. Other metrics are available without use of TIBCO Hawk. The version of TIBCO Hawk provided with TIBCO Runtime Agent is sufficient.

Before you begin, the following monitored cluster project files must be correctly configured and available on the MM server machine.

- The CDD files: Cache OM and In-Memory are supported. However, monitoring In-Memory and Cache OM simultaneously is not supported.
- The EAR files: The EAR files containing the compiled TIBCO BusinessEvents Studio projects.

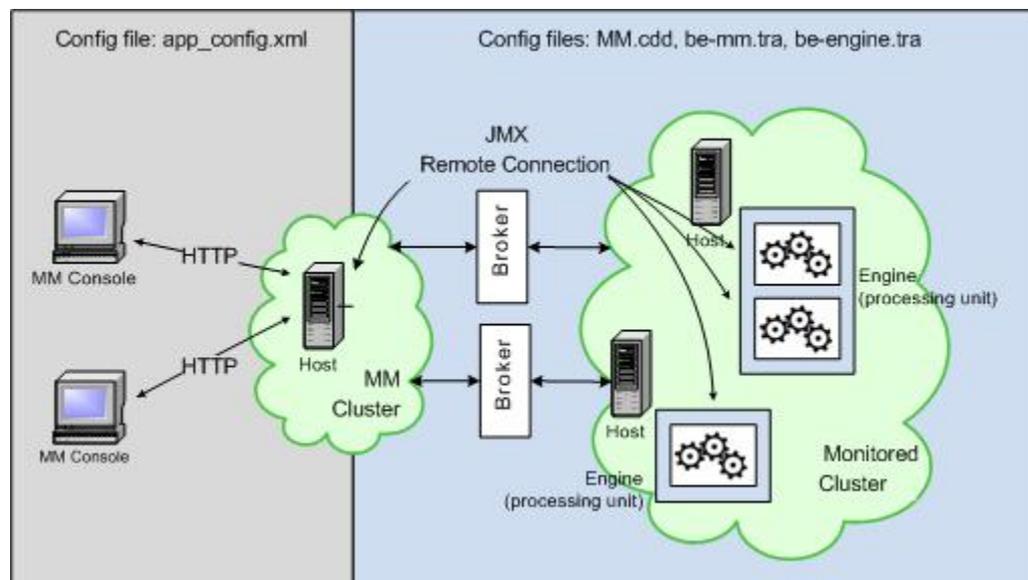
See *TIBCO BusinessEvents Developer's Guide* for details on maintaining these resources.

MM Runtime Architecture

TIBCO BusinessEvents Monitoring and Management (MM) consists of MM consoles and a MM cluster.

The conceptual diagram shows the MM cluster in the center, the web-based MM Console on the left, and one instance of the monitored cluster on the right. The monitored cluster is connected to the MM cluster through JMX, and through a Java process that uses RMI to connect the two clusters. This process is known as a broker. Note that the broker is used only with the Coherence cache provider and is not required for the TIBCO BusinessEvents DataGrid cache provider.

MM Runtime Architecture



Remote JMX connections enable MM to connect to MBeans exposed in the monitored cluster's engines. These MBeans allow the user to invoke remote operations from MM Console to gather performance metrics. Additionally, software utilities are used for remote start and deployment, and TIBCO Hawk is used for machine level metrics.

Files used to configure the console and the connection to the monitored cluster are shown along the top of the diagram.

Software for Remote Start and Deployment

Several software utilities are available to perform remote operations.

For more details, refer to the software utility's documentation.

Software Options for Deployment, Remote Start, and Remote Method Invocation

Software	Deployment	Remote Start, Stop	Remote Method Invocation
SSH	Yes	Yes	
PsTools (Windows)		Yes	
TIBCO Hawk		Yes	Yes
JMX (required)			Yes

Deployment using MM requires SSH

Although it is possible to use more than one utility for the machines in the cluster, for best results use only one across all the machines. Ensure that the software is installed and running on all relevant machines.



The software you use on each machine in the monitored cluster is specified in the cluster's site topology file. See [Start PU Method Setting](#).

SSH

Only SSH software enables MM to deploy TIBCO BusinessEvents software to the predefined hosts, that is, those configured in the monitored cluster's site topology file. SSH can also be used to start remote engines.

The SSH utility is available on UNIX machines by default and no action is required. On Windows machines, you must install an SSH server.

The user name and password that you use to stop the engine are the same user name and password that you use to log into the BEMM UI. They are not the credentials of the remote machine that you used to start the engine.



These credentials are different because to start the engine you use SSH, and to stop the engine you use one of the MBeans methods, which use the MM user credentials.

Therefore, SSH certificates cannot be used with the stop operation.

Installing and Configuring an SSH Server

If you want to use SSH on Windows machines, you must download the software and install it. Many SSH servers are available. For Windows, OpenSSH and Copssh are supported. See the product readme file for specific versions that are supported.

TIBCO has tested with OpenSSH software. See the product readme file for specific versions that are supported. If you use the OpenSSH server, note the following when installing OpenSSH:

- The OpenSSH package is not a part of the default Cygwin installation. During its installation, ensure that you select the OpenSSH package. Also, select the option `Select required packages (RECOMMENDED)` to install all the required packages to satisfy the dependencies.
- Accept the default username suggested when configuring the OpenSSH server and provide a password for the username.
- For deployment and starting PUs, it is best to use the user login that was used to install and configure OpenSSH. The credentials of the user can be specified in the host settings of the site topology file, **Host Settings User** and **Password** fields.

However, if you choose to use a different user, ensure that the user is added to the SSH server.

TIBCO Hawk

To use TIBCO Hawk to start remote engines, install and run it on the MM server machine and on all the client machines that use MM with TIBCO Hawk.

All machines in the same cluster must use the same TIBCO Hawk domain and the same transport definitions (server, network, daemon port).

TIBCO Hawk is also used for machine-level monitoring. See [TIBCO Hawk Configuration for Machine Level Metrics](#).

The same properties are used for both purposes. TIBCO Hawk is used for remote start only if specified in the cluster's site topology file. See [Start PU Method Setting](#).

PsTools

PsTools is an open Windows utility that enables execution of processes on remote machines. You can use PsTools to start remote engines only when both MM and the target host run on Windows.

Installing and Configuring PsTools: Accepting the Certificate

To use PsTools, download it to every target machine and save it to `BE-HOME/mm/bin/pstools`.

The first time PsTools is run on a client machine, a pop-up window appears. Accept the certificate so that PsTools becomes fully functional. Do this once on each host machine.

TIBCO Hawk Configuration for Machine Level Metrics

To monitor machine level statistics, use TIBCO Hawk software as well as TIBCO Rendezvous as the transport.

The version of TIBCO Hawk provided by TIBCO Runtime Agent is sufficient for this functionality.

Using TIBCO Hawk allows you to visualize machine-level metrics in the MM UI. These machine-level metrics are made available by TIBCO Hawk micro agents. If you do not configure the TIBCO Hawk domain, the enterprise monitor will not show the machine-level metrics, but it will show all other levels of metrics.

TIBCO Hawk can also be used for starting remote engines. See [Software for Remote Start and Deployment](#) for other options.

Configuring TIBCO Hawk

The configuration shown here is used both for configuring machine level metrics and for remote engine startup.

Procedure

1. Install TIBCO Hawk and TIBCO Rendezvous software on the MM server machine, and on all client machines in the TIBCO BusinessEvents cluster whose machine-level metrics you want to monitor.
2. Configure a Hawk domain. A Hawk domain specifies a group of TIBCO Hawk agents that acts as a monitoring set. Each machine has a Hawk agent and various micro agents (HMA) that provide useful machine-level metrics to the enterprise monitor.

Use the same Hawk domain name and Rendezvous transport for all the monitored engines and for the emonitor application.

3. Import the `BE_HOME/MM/project/emonitor` project into your workspace and edit the `MM.cdd`. If you copy files into the workspace, remember to copy the `MM.cdd` file to the above location. In the `mm-class` agent properties list, add the following property to specify the Hawk domain:

```
tibco.clientVar.Domain=TIBCO Hawk Domain
```

4. If you use non-default values for the Hawk transport properties, specify them in the `mm-class` agent properties list in the `MM.cdd` also. The properties are as follows:

```
tibco.clientVar.TIBHawkDaemon=Rendezvous daemon used by Hawk
tibco.clientVar.TIBHawkNetWork=Rendezvous network used by Hawk
tibco.clientVar.TIBHawkService=Rendezvous service used by Hawk
```

To use the default client socket, omit the daemon argument. Default service is 7474, and daemon is `tcp:7474`. See Hawk documentation for additional information.

5. Add the same properties you added to the `MM.cdd` file to the monitored project's CDD file, in the Cluster tab properties sheet.
6. In the `BE_HOME/mm/bin/be-mm.tra` file, set the `tibco.env.HAWK_HOME` property and the `tibco.env.RV_HOME` to point to the TIBCO Hawk and TIBCO Rendezvous installation root directories.
7. In the TRA files of all monitored cluster engines, set the properties `tibco.env.HAWK_HOME` and `tibco.env.RV_HOME`.

JMX Properties and To-Be-Monitored Engine TRA Files

After the TIBCO BusinessEvents cluster engines are started, they use JMX MBeans to expose monitoring and management information to the MM server, and to allow remote method invocation.

The JMX port number must be specified before the engine's JVM starts. A variable for the port number is provided in the TRA file so that the actual value can be specified before the engine starts.

Note that in the current release, JMX with SSL is not supported.

JMX Properties Configuration

JMX properties are provided in the shipped `BE_HOME/bin/be-engine.tra` file.

The following properties have to be commented:

- `#java.property.be.engine.jmx.connector.port=%jmx_port%`
- `#java.property.be.engine.jmx.connector.authenticate=false`

These properties have to be commented for all TRA files for all monitored TIBCO BusinessEvents engines as needed.

Enabling Monitoring and Management

JMX for monitoring and management has to be exposed without authentication.

Procedure

1. Uncomment the following property:
`java.property.be.engine.jmx.connector.port=%jmx_port%`
2. Ensure that the value of the port property is set to this literal value: %jmx_port%.
 The actual value is substituted at run time.
3. See [JMX Remote Port Number Setup at Runtime](#).



When more than one PU (engine) is deployed on the same host, ensure that a different JMX port is used for each of the PUs, in the site topology file.

Enabling JMX MBeans Authentication

To enable authentication follow these steps:

Procedure

1. Set the following property:
`java.property.be.engine.jmx.connector.authenticate=true`
2. Configure the authentication technology you want to use in the emonitor project as explained in [User Authentication](#).

JMX Remote Port Number Setup at Run time

When you use the MM UI to start TIBCO BusinessEvents engines remotely, MM reads the port number from the PU configuration setting in the site topology file.

MM passes this value to the TRA file's `jmx_port` variable, in the command line it composes when starting a TIBCO BusinessEvents engine:

```
-propVar jmx_port=portnum
```



If you start an engine manually from the command line, provide the port number in the same way, that is, using the option `-propVar jmx_port= portnum`. If the port number is not specified, the default port 5555 is used.

Use the same JMX port number as specified in the PUC so that MM treats the engine as a predefined engine. If you use a different number, the engine starts as an un-predefined engine.

User Authorization for Administrator and User Roles

MM authorization uses two preconfigured roles.

These roles are specified in the provided passwords file that is used for file-based authentication:

```
BE_HOME/mm/config/users.pwd
```

The file as shipped contains the following entries:

```
jdoe:A31405D272B94E5D12E9A52A665D3BFE:MM_ADMINISTRATOR;
mm_user:11b2016b63c99ef7ab6d6d716be7b78e:MM_USER;
admin:21232f297a57a5a743894a0e4a801fc3:MM_ADMINISTRATOR;
```

If you add more users, ensure that they have the appropriate role. Note that role names are case sensitive:

- *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



To use LDAP authentication, add these roles in the LDAP directory for the relevant users.

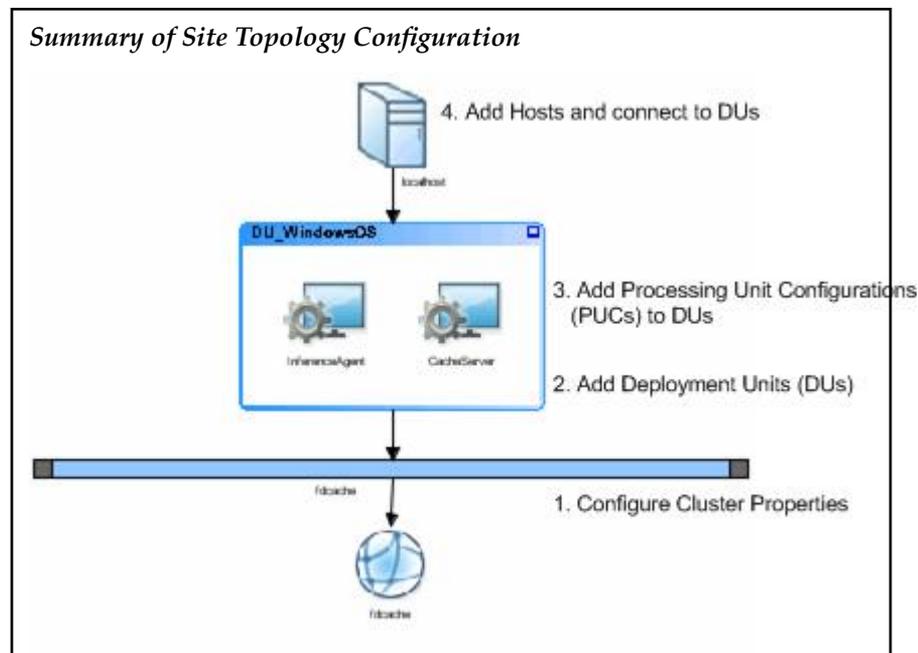
See [User Authentication](#) for more on authentication topics, and configuring the password file.

Site Topology

The site topology file contains deploy-time information such as what processing units to deploy to specific machines in your environment.

You need to know information about the machines that will host the agents you plan to deploy, for example information about the operating system and IP address of the machines.

You also need to know what remote invocation software you will use to start remote processes on these machines: TIBCO Hawk, PSTools, or SSH.



- Changes to the EAR file do not affect the topology configuration. However, if the cluster, processing unit, or agent definitions in the CDD file change, you must recreate the site topology file using the updated CDD.
- If you change the site topology, you must restart the MM server.

When possible, use the graphical site topology file editor in TIBCO BusinessEvents Studio. It provides validation and structure that are helpful.

When working on runtime installations, however, it may not be possible to use TIBCO BusinessEvents Studio. An annotated site topology file template is available, so you can edit the XML-based topology file in a text editor. You can access it at the following location:

```
BE_HOME/mm/config/site_topology_template.st
```

If you are editing the file directly, adapt the GUI-based instructions accordingly. The configuration requirements are the same in both cases.

Using the canvas-based editor in TIBCO BusinessEvents Studio, you can create a visual representation of the desired site topology. Using the tabs that show the properties, you can configure each item

represented by the topology diagram icons: the cluster, hosts, deployment units, and processing unit configurations.

The output of this activity is an XML file used in MM. A summary of the steps is shown below. For detailed steps, see [Site Topology in TIBCO BusinessEvents Studio](#).

Configuring the Site Topology

The site topology is best configured using the graphical site topology file editor in TIBCO BusinessEvents Studio.

Procedure

1. Configure Cluster Properties

In the Cluster Properties tab, reference the fully configured CDD and EAR files for your project. See [Project Master and Deployed Locations of CDD and EAR Files](#) for more on the use of these files.

2. Add Deployment Units (DUs) to the canvas as needed.

For each DU, specify the following:

- The location of the CDD and EAR files. MM copies the files to the specified location at deploy time.



See the note in [Project Master and Deployed Locations of CDD and EAR Files](#) for an important limitation when deploying multiple DUs on one machine.

- One or more *processing unit configurations* (PUCs). You will configure the PUCs in the next step.

3. Add Processing Unit Configurations (PUCs) to DUs.

For each PUC, select one processing unit (PU) from the list of PUs defined in the CDD file. Set deploy time properties such as the JMX ports used by MM to communicate with the deployed engine.

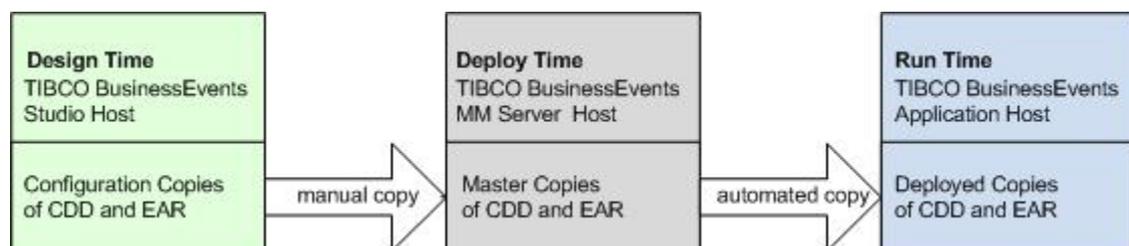
4. Add Hosts.

Specify the machine configuration here, including the software used on the remote machines to start processes remotely. To deploy a DU to a host, connect the DU to that host in the canvas editor. Multiple hosts can use the same deployment unit if you want to reuse an identical configuration on more than one machine.

Project, Master and Deployed Locations of CDD and EAR Files

In the topology file, reference three locations for the CDD and EAR files.

Locations for the CDD and EAR Files



The files in each location must be exact copies:

- Project CDD file: In the **Cluster Configuration** tab, specify a locally available copy of the project CDD, used only at design-time for configuring the topology file in TIBCO BusinessEvents Studio.
- Master CDD and EAR files: In the **Cluster Configuration** tab, specify the location of the master CDD and EAR files. These copies must be manually copied to the specified location on the MM server, for use in deployment.

- **Deployed CDD and EAR files:** In the Deployment Unit settings, specify where MM will place the CDD and EAR files when it performs deployment.

The project and master CDD can be in the same location if you use one machine to configure the topology file and to run MM server. These two sets of fields are available in case you configure the topology on a different machine from the MM server machine.



- All locations specified must already exist. The software does not create directories.
- Use the correct path delimiter for the operating system of the host machines.

Deployment-Specific Processing Units and Global Variables

In general, you can reference one processing unit multiple times to create different processing unit configurations (PUCs). However processing units that have deployment-specific settings cannot be used in this flexible manner.

- **Agent-Instance-Specific Properties:** If a processing unit contains agent-instance-specific properties such as agent key and priority settings, use it in only one PUC, which is used in only one DU that is itself used only once in the deployment.
- **Host-Specific Processing Units:** Processing units can host-specific settings. If a deployment unit contains a PUC that references such a processing unit, link it only to the appropriate host for deployment. For example, the Coherence cache provider property `tangosol.coherence.localhost` property is a host-specific setting, and so is the TIBCO BusinessEvents DataGrid property `be.engine.cluster.as.listen.url`.
- **Global Variables:** Global variable overrides (if any) set in the master CDD are used for command-line deployment. They can be overridden by TIBCO BusinessEvents Monitoring and Management settings. If you plan to deploy using MM, override global variables using MM, instead of in the CDD file.

Site Topology in TIBCO BusinessEvents Studio

Before you begin, ensure that you have a valid CDD file. The processing units that you deploy to the various hosts are defined in the CDD.

Adding a Site Topology Diagram



Disable Animation to Avoid Display Issues. Animation can cause display problems. To disable animation, go to **Windows > Preferences > TIBCO BusinessEvents > Diagram**. In the Animation section, clear the **Allow** check box.

Procedure

1. Open the project in TIBCO BusinessEvents Studio. Select the project root, right-click and select **New > Other > TIBCO BusinessEvents > Site Topology**.
2. At the New Site Topology Wizard, enter a unique Site Topology name and optional description.
3. Select the Cluster Deployment Descriptor (CDD) that contains the PU definitions and other details you want to use. Only CDD files within the studio project you configure are available for selection.
4. Click **Finish**. The site topology editor is now ready to build the site topology diagram.

Configuring the Site Topology

Procedure

1. On the canvas, click the site icon (the globe). In the **Site Properties** tab you can change the site name and description as desired. Other fields are view-only.
2. Click the blue bar, which represents the cluster. In the **Cluster Properties** tab, specify the following:
 - The location of the Project CDD, which must be available to the TIBCO BusinessEvents Studio Explorer. The CDD you have selected with the wizard.
 - The location of the Master CDD and EAR on the MM server, which reads these files and copies them to the remote deployment locations specified in the DUs.



If the MM server is on a different machine from the machine where you are running TIBCO BusinessEvents Studio, copy the master files to the specified location so they are available for use by MM.

See [Project Master and Deployed Locations of CDD and EAR Files](#) for details.

3. Add one or more deployment units: In the Site Topology section of the palette to the right of the canvas, click the deployment unit icon and then click the canvas. A DU icon appears on the canvas. Click again to add more DUs. Right-click the canvas to stop adding units. (If the palette is not visible, click **Window > Show View > Palette** or **Window > Reset > Perspective**).
A connection arrow appears automatically, connecting each deployment unit to the cluster.
4. Click each DU in turn and configure the **Deployment Unit Properties** tab settings.
 - In the Deployed CDD and Deployed EAR fields, specify the directory where MM will put the files when it deploys this DU to the host machine.
 - Click **Add** and add one or more processing unit configurations (PUCs) to the deployment unit.
5. Configure processing unit configurations (PUCs): In the DU property sheet, double-click one of the listed PUCs (or click the PUC icon shown in the diagram). The Processing Unit Configuration properties appear. Configure the PUC as follows (and configure the rest of the PUCs in a similar way):
 - Replace the default PUC name with a name of your choice.
 - Select the option to use the PUC name as the engine name.
 - Select the processing unit to use for this configuration. The list displays the PUs defined in the CDD. You can use one PU in multiple DUs, as appropriate. When you select a PU, the number of agents defined for it is displayed. (No agent level configuration is done in the site topology editor.)
 - Set the JMX port for MM to perform monitoring and management. When multiple PUs are running on one host, each PU must have a different JMX port. You can reuse ports on different hosts.
6. Add one or more hosts. In the Site Topology section of the palette, click the **Host** icon, and then click the canvas. A **Host** icon appears on the canvas. Click again to add more hosts. Right-click in the canvas area to stop adding hosts.
7. Click each **Host** icon in turn and configure the **Properties** tab.
 - In the **General** tab, configure the host name (including the domain extension), IP, and as needed, the user name and password, and operating system.
 - In the **Installation** tab, ensure the TIBCO BusinessEvents Home and TRA file locations are correctly specified.

- In the Start-PU-Method tab, select an option to use for MM to start a processing unit on this host.
8. Connect each host to one or more deployment units:
- In the **Links** section of the palette, click the **Connect** icon.
 - Click a host and then the title bar of the deployment unit you want to deploy on that host.
- Right-click to stop connecting.
- To remove a connection, right-click to stop connecting, then right-click a connection arrow and click **Delete**.
9. Click **Save**.



The canvas has a property sheet too: click an empty area of the canvas to see the number of deployment units and number of processing units in the site topology.

Site Topology Files for the MM Server

The MM server parses and loads all the site topology files, except the template file `site_topology_template.st`, located at `BE_HOME\mm\config`.



For backward compatibility, MM server loads the site topology file specified using the property `be.mm.topology.file` in the `MM.cdd` file. However, this property is deprecated in the release 5.1. TIBCO recommends that you put the site topology file to the `BE_HOME\mm\config` location.

If the location `BE_HOME\mm\config` does not contain any site topology file, the MM server fails to start with an appropriate message in the log file.

If a site topology file (with the same name) is present under `BE_HOME\mm\config` and is also specified using the property `be.mm.topology.file` in the `MM.cdd` file, only the site topology file specified in the property will be parsed and loaded by the MM server.

In a multi-cluster configuration, all the Master CDD files defined on every site topology file (STF) loaded by the MM server must have the same Cache Object Model. You cannot load one STF pointing to a Master CDD using Coherence Cache/DataGrid and another STF pointing to a Master CDD using TIBCO Cache/DataGrid.

Site Topology Reference

Site Topology – Site Settings

Property	Notes
Site Name	Default value is the name of the site topology file. In a multi-cluster configuration, the name must be the same for every cluster (in each site topology file).
Description	Description of the site.

Property	Notes
Number of Clusters	<p>Read-only field. Only one cluster per site is supported in this release. (Not present in the site topology XML file.)</p> <p>In a multi-cluster configuration, each cluster is configured in a different site topology file. Hence, the number of clusters in the site topology file will always be one.</p>
Number of Hosts	Read-only field displaying the number of hosts in this topology. (Not present in the site topology XML file.)

Site Topology – Cluster Settings

Property	Notes
Cluster Name	<p>Read-only field displaying the cluster name specified in the CDD. This name is set in the Cluster Name field of the CDD editor.</p> <p>In a multi cluster configuration, the cluster name must be unique for each cluster.</p>
TIBCO BusinessEvents Version	<p>Read-only field displaying the version of TIBCO BusinessEvents. Must match the Host Settings field of the same name.</p> <p>Note</p> <p>If you copy a site topology from a prior release into a project and open it in the current release, this field value displays the prior release version number. The field is red, and it is editable. Update the field value to the current version. It again becomes a read-only field. Also change the host-level TIBCO BusinessEvents Version field in a similar way: both must match. If you migrate the project, this issue does not occur. See the section "Migration" from 4.x to 5.1, in <i>TIBCO BusinessEvents Installation</i>.</p>
Project CDD	Location and name of project CDD. This is the location used by TIBCO BusinessEvents Studio for configuration of the site topology. See Project Master and Deployed Locations of CDD and EAR Files .
Master CDD	Location and name of the master CDD. This is the location used by the MM server. See Project Master and Deployed Locations of CDD and EAR Files .
Master EAR	Location and name of the master EAR. This is the location used by the MM server. See Project Master and Deployed Locations of CDD and EAR Files .

Site Topology – Deployment Unit Settings

Property	Notes
Deployment Unit Name	<p>Name of the deployment unit. It can be helpful to include the operating system of the host to which you will deploy this DU in the DU name. If a DU contains any host-specific settings, it is also a good idea to put the host name in the DU name.</p> <p>In a multi cluster configuration, the deployment unit name must be unique.</p> <p>Note</p> <p>Paths in different operating systems are specified using different tokens. Even if the DUs are identical in all other settings, you must create different DUs for different operating systems.</p> <p>Default value is DU_n where n is a number that increments each time you add a DU to the diagram.</p>
Deployed CDD	<p>Absolute file path to the location where the MM server will deploy the copy of the master CDD. See Project Master and Deployed Locations of CDD and EAR Files .</p>
Deployed EAR	<p>Absolute file path to the location where the MM server will deploy the copy of the master EAR. See Project Master and Deployed Locations of CDD and EAR Files .</p>
Processing Unit Configurations	<p>Displays a list of processing unit configurations. Adding and configuring PUCs is explained in Site Topology in TIBCO BusinessEvents Studio .</p>

Site Topology – Processing Unit Settings

Property	Notes
Processing Unit Configuration Name	<p>The name that identifies this configuration of the processing unit, as specified in the Processing Unit setting (see below). The processing unit name must be unique across the deployment unit.</p> <p>The processing units settings are configured in the CDD. For more information, see <i>TIBCO Business Events Developer's Guide</i>.</p>

Property	Notes
Use As Engine Name	<p>Check this checkbox to use the value of the Processing Unit Configuration Name field as the engine name.</p> <p>For best results, ensure that you use the same choice across all processing units in the cluster.</p>
Processing Unit	<p>Select the processing unit that you want to use. Only processing units configured in the CDD selected as the Project CDD appear in the list. The same processing unit can be used in multiple PUCs.</p>
Number of Agents	<p>Displays the number of agents in the selected processing unit. (Not present in the site topology XML file.)</p>
JMX Port	<p>JMX port used by MM to perform monitoring and management. Required.</p> <p>When more than one PU is deployed on the same host (in one DU or multiple DUs), you must ensure the JMX port in each of these PUs is different.</p> <p>See JMX Remote Port Number Setup at Runtime for more details.</p>

Site Topology – Host Settings

Property	Notes
General Settings	
Host Name	<p>Name of the machine hosting the mapped DUs (including the domain extension). Used for remote access. Used to identify the host in the MM user interface. Required.</p> <p>To validate the hostname, ping the host using this name from the MM server machine.</p> <p>Note</p> <p>Specify the exact name of the host. Errors in the host name result in the host appearing in the MM Console UI as an unpredefined machine. Do not, for example, use localhost.</p>
IP	<p>IP address of the host machine. Used for remote access. Required.</p>

Property	Notes
User Name	<p>User name to log onto the host machine.</p> <p>The user credentials are used for remote deployment and execution, including starting a process unit.</p> <p>At run time, a dialog box pops up to authenticate the user, for example when deploying a PU. If you provide a user name and password here, then the dialog is prepopulated with these values. You can enter different values as needed.</p> <p>If you do not provide the credentials here, then you must provide them at the pop-up dialog.</p> <p>You can specify a local user or a domain user.</p> <p>Enter details for the user you specified for the remote connection utility you are using. For example, if you use PsTools, specify <i>domain\user</i> for domain users.</p> <p>Note</p> <p>If you use a SSH server, you must specify the same user credentials here that you used to install the SSH server, or the credentials of a user who is enabled to login to the SSH server.</p>
Password	<p>Password of the user referenced in the User Name field. The password is encrypted.</p> <p>See notes in User Name section.</p>
Operating System	<p>Operating system of the host machine. See the product readme for a list of supported platforms.</p>
Installation Settings	
TIBCO BusinessEvents Version	<p>Read-only field displaying the version of TIBCO BusinessEvents. Must match the Cluster Settings field of the same name.</p> <p>Note</p> <p>If you copy a site topology from a prior release into a project and open it in the current release, this field value displays the prior release version number, the field is red and it is editable. Update the field value to the current version. It again becomes a read-only field. Also change the cluster level TIBCO BusinessEvents Version field in a similar way: both must match.</p>
TIBCO BusinessEvents Home	<p>Install location of TIBCO BusinessEvents on the host machine, for example:</p> <p><code>c:/tibco/be/<version></code></p>

Property	Notes
TRA File	Location of the be-engine.tra file, for example: <code>c:/tibco/be/<version>/bin/be-engine.tra</code>
Start PU Method Setting	
Start-PU-Method	<p>Choose the method that MM will use to start this processing unit on remote machines:</p> <ul style="list-style-type: none"> • Use Hawk • Use PsTools • Use SSH. If you choose Use SSH, and do not want to use the default port number of 22, then also enter the port. The host must accept a secure connection through this port. Using the default port is generally recommended because it is also the default port used by most Linux SSH servers. <p>Note that a user name and password for the remote machines are required for MM to connect (see notes for User Name and Password fields).</p> <p>See Software for Remote Start and Deployment for details on each option.</p> <p>Default is SSH. Default SSH port number is 22.</p>

Basic MM Settings in MM.cdd

The MM server uses the `MM.cdd` file to perform basic configuration and also to configure alerts, health metric rules, and actions.

Configurations for alerts, health metric rules, and actions are uniform across all the clusters monitored by the MM server. These configurations cannot be customized for each cluster individually.

Expert users can also edit the XML file using a text editor, but using the UI protects you from editing errors. Back up the file before editing it.

To run the MM server with the installation defaults, the `MM.cdd` file must remain in the installed location.

Within the CDD file change only the settings for the `mm-class` agent. The `mm-class` agent is defined using an internal type of agent class (Monitoring & Management) used only in the `MM.cdd` file.

Whenever you change the `MM.cdd` file, restart the BEMM server so that it uses the updated values.

Importing the emonitor Project for CDD Editing

The emonitor file cannot be edited in TIBCO BusinessEvents Studio outside of its project context.

To edit the CDD in TIBCO BusinessEvents Studio, import the project into your workspace.



As with any procedure that changes files, make sure that the emonitor project is backed up before you edit its CDD file.

Procedure

1. In TIBCO BusinessEvents Studio, choose **File > Import Existing Projects into Workspace** and select the following project:
`BE_HOME/mm/project/emonitor`
2. To edit the CDD file in its original location, clear the **Copy the projects into workspace** check box. (If you do copy the emonitor project into your workspace, remember to copy the edited CDD file to its original location.)
3. In Studio Explorer, double-click **MM.cdd** to open it in the CDD editor.

Configuring the Basic Settings in the MM.cdd File

To edit the CDD using the CDD editor, import the eMonitor project into TIBCO BusinessEvents Studio.

Procedure

1. Import the emonitor project into your workspace and open the CDD file for editing. See [Import the emonitor Project for CDD Editing](#) for details.
2. In the CDD editor, click the **Agent Classes** tab and select `mm-class` agent.
3. In the properties sheet, complete the values as explained in [MM Agent Basic Configuration Reference](#).

To monitor engines running in memory mode, add the property, `be.mm.monitor.in.mem`, to the properties and set its value to true.

To monitor engines running in cluster mode, set the value of the property to false. Alternately, you can remove the property from the properties sheet.

Specify the host address in the `tibco.clientVar.HTTPHost` property, and specify the port as needed.



If you are running on AIX you must add an additional property to the mm-agent class CDD properties. See [Setting Property for Cache Based Object Management on AIX](#) for details.

4. The monitored cluster's topology file must be located on the MM server under `BE_HOME/mm/config`. (see [Site Topology](#)). Copy the file from its design-time location as needed. To monitor and manage other projects, ensure that the property `be.mm.topology.file` is either deleted from the MM.cdd file, or the property is set to ""(empty string).
5. Save the CDD. If you imported it to your workspace, copy it to `BE_HOME/mm/project/emonitor` and to `BE_HOME/mm/bin`.



Whenever you change the `MM.cdd` file restart the BEMM server so that it uses the updated values.

MM Agent Basic Configuration Reference

This section provides a reference to the properties to be set in the Master CDD file.

Master CDD mm-class Agent Class Properties

Property	Notes
<code>be.mm.monitor.in.mem</code>	

Property	Notes
	<p>Enables the MM server to monitor engines running in memory mode when this property is set to true.</p> <p>You can either delete this property, or set the property to false to indicate monitoring engines running in cluster mode.</p>
<code>be.mm.topology.file</code>	
	<p>This property is deprecated in 5.1. The MM server locates and picks up all the site topology files present under <code>BE_HOME\mm\config</code>. To monitor and manage other projects, ensure that this property is either deleted from the file, or the property is set to ""(empty string).</p> <p>Fully qualified path to the site topology file for the cluster to be monitored as shipped, points to the example project Fraud Detection Cache in TIBCO BusinessEvents®. See Site Topology and the sections following for more details.</p> <p>Note</p> <p>The topology file must be located on the same machine as the machine where the MM server is running.</p>
	<p>Authentication Properties (auth section)</p> <p>Authentication is optional. To avoid conflicts between authentication for MM and for add-on products that also use authentication, MM has its own set of property names. Authentication topics are documented in User Authentication.</p>
	<p>HTTP Properties</p> <p>Do not change unless advised by TIBCO.</p>
<code>be.channel.http.propertiesAsXML</code>	
	<p>Do not change unless advised by TIBCO.</p> <p>Default is true.</p>
	<p>Email Properties for Actions Feature</p> <p><code>actions_email</code></p> <p>Configure the properties needed to send emails in response to alerts or health levels. Required only if you will use the Actions feature. Replace default values for host, email addresses and passwords.</p>
<code>be.mm.email.protocol</code>	
	<p>Email protocol. Supported protocols are smtp and smtps.</p> <p>Default value is smtp.</p>
<code>be.mm.email.authentication</code>	

Property	Notes
	<p>Specifies whether the user must authenticate to the email server.</p> <p>Possible values are true and false. If set to true then also configure the username and password properties.</p> <p>Default is false.</p>
<code>be.mm.email.host</code>	
	Specifies the email host.
<code>be.mm.email.from</code>	
	Specifies the From address.
<code>be.mm.email.username</code>	
	<p>Specifies the username used to authenticate to the email server.</p> <p>Used only if <code>be.mm.email.authentication</code> is set to true.</p>
<code>be.mm.email.password</code>	
	<p>Specifies the password used to authenticate to the email server.</p> <p>Used only if <code>be.mm.email.authentication</code> is set to true.</p>
Global Variable Overrides – HTTP	
<code>global_variable_overwrite > http</code>	
	Overrides for the project global variables.
<code>tibco.clientVar.HTTPHost</code>	
	<p>Used by the emonitor project HTTP channel. Set to the IP address of the machine hosting the MM server.</p> <p>Default value is localhost</p>
<code>tibco.clientVar.HTTPPort</code>	
	<p>Used by the emonitor project HTTP channel. Specify as needed to avoid port conflicts.</p> <p>Default is 9000.</p>
<code>tibco.clientVar.BEMMDocRoot</code>	
	<p>The directory from which static HTML content is served.</p> <p>Do not change unless advised by TIBCO.</p> <p>Default is ../web-root.</p>
<code>tibco.clientVar.BEMMDocPage</code>	

Property	Notes
	<p>The name of the default static HTML file stored in the document root. Do not change unless advised by TIBCO.</p> <p>Default is <code>\index.html</code>.</p>
	<p>Global Variable Overrides – Frequencies</p> <p><code>global_variable_override > frequencies</code></p> <p>These properties define how frequently certain checks and updates are done. Modify as needed to ensure best performance.</p> <p>Overrides for the project global variables.</p>
	<p><code>tibco.clientVar.TopologyUpdateFreq</code></p>
	<p>Specifies the time interval between two consecutive calls to the monitored cluster to fetch the latest (current) cluster topology (in milliseconds). The purpose of this check is to ensure that the Cluster Explorer topology matches the actual cluster topology.</p> <p>Default is 30000.</p>
	<p><code>tibco.clientVar.JMXUpdateFreq</code></p>
	<p>Specifies the time interval between two consecutive calls to get metrics of each monitored entity in the cluster (in milliseconds). The purpose of this check is to ensure that metrics at all monitored entity levels are updated regularly.</p> <p>A monitored entity in a cluster can be a cluster, a machine, a process unit or an agent.</p> <p>Adjust as needed. For example, if the requests are affecting performance, increase the time interval. If you want more immediate notifications, decrease the time interval.</p> <p>Default is 30000.</p>
	<p><code>tibco.clientVar.HealthCheckFreq</code></p>
	<p>Specifies the time interval between two consecutive health pings to each machine and process in the monitored cluster (in milliseconds).</p> <p>This property is used to determine which monitored entities are inactive. Inactive items are indicated in Cluster Explorer. See Inactive Members for more details.</p> <p>Default is 30000.</p>
	<p><code>tibco.clientVar.SweepFreq</code></p>

Property	Notes
	<p>This setting applies only to unpredefined cluster members. Predefined cluster members (those defined in the topology file) are never purged.</p> <p>The time interval between two sweep checks to physically delete inactive purged cluster members in the discovered topology (in milliseconds).</p> <p>The <code>SweepFreq</code> property determines how often the system checks for inactive unpredefined cluster members to be purged, and the <code>SweepThreshold</code> property determines for how long an item must remain inactive before it is eligible for purging.</p> <p>If a user is viewing an inactive cluster member that another user has purged, the cluster member remains visible until the user has finished viewing the details.</p> <p>Note</p> <p>Do not change unless advised by TIBCO.</p> <p>Default is 300000 (that is, five minutes)</p>
<code>tibco.clientVar.SweepThreshold</code>	
	<p>The length of inactivity for a purged cluster member to be qualified for physical deletion (in milliseconds).</p> <p>Also see notes for <code>Sweepfreq</code>.</p> <p>Note</p> <p>Do not change unless advised by TIBCO.</p> <p>Default is 600000 (that is, ten minutes)</p>
Global Variable Overrides – Hawk	
<code>global_variable_overwrite > hawk</code>	
	<p>Overrides for the project global variables. If you use non-default values, specify them here.</p>
<code>tibco.clientVar.Domain</code>	
	<p>Name of the TIBCO Hawk domain.</p>
<code>tibco.clientVar.TIBHawkService</code>	
	<p>Rendezvous service used by TIBCO Hawk.</p>
<code>tibco.clientVar.TIBHawkDaemon</code>	
	<p>Rendezvous daemon used by TIBCO Hawk.</p>
<code>tibco.clientVar.TIBHawkNetWork</code>	
	<p>Rendezvous network used by TIBCO Hawk.</p>

Broker Properties for Working with Coherence Cache Provider

A broker process enables the MM cluster to communicate and retrieve information from the target cluster.

A multi-cluster configuration must have one broker per cluster, with each broker specifying its own set of properties. Each broker must have its unique RMI port. The RMI port is specified using the property `be.mm.broker.rmi.port`. The value of this property must be an integer and the default value is 11200.



The broker properties are not required for TIBCO BusinessEvents DataGrid clusters.

The broker properties *must* be specified in the Master CDD file at the cluster level. In the Master CDD file, only the properties present at the cluster level with the prefix `tangosol.coherence` or `be.mm.broker` are passed to the broker process.

For backward compatibility, the broker properties defined in the MM.cdd file with the prefix `be.metric.cluster.property` are still loaded. However, if a property with the same name exists in the MM.cdd file and the Master CDD file, the value specified in the Master CDD file takes precedence.

Broker Properties Reference

Broker properties reference contains properties for working with a Coherence cache provider.

Master CDD: Broker Properties for Working With Coherence Cache Provider

Property	Notes
<code>be.mm.broker.killoldbroker</code>	<p>A broker process enables the MM cluster to retrieve information from the target cluster. If MM stops, the broker terminates after about three minutes.</p> <p>By default, if MM restarts before the previous broker terminates, it uses that broker. If the previous broker has terminated, however, then MM creates a new broker process, using the target cluster properties in the master CDD that is specified in the topology file.</p> <p>When set to true, the existing broker process is never reused on startup. For example, if you want to monitor a different cluster when you restart MM (by specifying a different topology file), then set this property to true.</p> <p>Default is <code>false</code>.</p>
<code>be.mm.broker.log.file</code>	<p>Location of the log file for the broker relative to the working directory. The broker connects to the cluster that is to be monitored. This file relates to cluster activities.</p> <p>Default is <code>logs/mm-broker.log</code>.</p>
<code>be.mm.broker.rmi.port</code>	

Property	Notes
	<p>Port used to establish the RMI (Remote Method Invocation) communication between the MM cluster and the cluster to be monitored.</p> <p>Specify a valid port number for this property. The value must be an integer that corresponds to an open port on the machine hosting the MM server. The default value of the port is 11200.</p> <p>Specify a unique RMI port for each Master CDD corresponding to the cluster that is to be monitored. The property must have a different value for each Master CDD file.</p>
<code>be.mm.broker.tangosol.coherence.localhost</code>	
	IP address of the server machine. Required only if you are using a non-default IP interface.
<code>be.mm.broker.tangosol.coherence.localport</code>	
	<p>Specifies the port that the socket will listen to or publish on.</p> <p>If more than one cluster is running on the same subnet, then even though they have different cluster names and cluster addresses, you may need to specify this property to avoid conflict.</p> <p>Also used for Coherence WKA configuration. See TIBCO BusinessEvents DataGrid WKA Discovery.</p> <p>Possible values are 1 to 65535.</p> <p>Default is 9000.</p>
<code>be.mm.broker.java.net.preferIPv4Stack</code>	
	<p>If the server is hosted on an AIX machine, set this property to <code>True</code>. Remember to also add a similar property to the <code>be-mm.tra</code> file. See Setting Property for Cache Based Object Management on AIX for general details (note that the property name is different here than in the to-be-monitored engine TRA file).</p> <p>Default is <code>false</code>.</p>

Coherence WKA Cluster Discovery

If you are using the Oracle Coherence cache provider and you have configured the monitored project to discover cluster members using well-known addresses (WKA), then make some additional changes to the project CDD so that MM can monitor and manage the cluster.

Configure the `mm.cdd` to work with the cluster to be monitored, and keep in mind that the MM cluster itself does not use the WKA discovery

For more details about WKA discovery, see CDD Configuration in *TIBCO BusinessEvents Configuration Guide*.



Use unique ports including for MM Server engine. If any of the monitored cluster engines are configured to run on the same machine as the MM server, ensure that the values for `localport` are unique across the MM server and the monitored clusters on that machine.

Configuring the Project's CDD to Communicate with the Cluster

This section assumes that the CDD for the cluster to be monitored has already been configured for WKA cluster discovery, so that the additional configuration enables MM to communicate with the cluster.

Procedure

1. Open the project to be monitored in TIBCO BusinessEvents Studio and open its CDD file in the CDD editor.

This CDD must be copied for use as the master CDD. See [Project Master and Deployed Locations of CDD and EAR Files](#) to understand the purpose of these copies of the CDD.

2. For each PU you will deploy to a WKA machine, add the `be.engine.hostaddress` property (in addition to the `tangosol.coherence.localhost` and `tangosol.coherence.localport` properties as needed for WKA configuration). Set it to the same value as the `localhost` property:

```
tangosol.coherence.localhost HostIP
tangosol.coherence.localport Hostport
be.engine.hostaddress HostIP
```

For better organization, put these properties into a property group, named as desired.

3. Add the following WKA properties to the cluster level properties:

```
tangosol.coherence.wka = IP_of_machine_hosting_MMServer
tangosol.coherence.wka.port = Unused_port_on_machine_hosting_MMserver
```



If the CDD has already been configured for WKA cluster discovery and either one or both of the properties, `tangosol.coherence.wka` and `tangosol.coherence.wka.port`, already exist in the CDD under the cluster level properties, rename such a pair of properties to `tangosol.coherence.wkan` and `tangosol.coherence.wkan.port`, where *n* is the first unused integer in the WKA list. All the other WKA properties can remain as they are.

Cluster configuration is documented in *TIBCO BusinessEvents Configuration Guide*.

Configuring the MM.CDD File

The `MM.cdd` file is used to perform basic configuration as well as alerts, health metric rules, and actions.

Procedure

1. Import the `emonitor` project into your workspace and open the CDD file for editing. See [Importing the emonitor Project for CDD Editing](#) for details.
2. If the MM server runs on the same host as any monitored cluster engine, specify the following properties in the **Cluster > Properties** sheet:

```
tangosol.coherence.localhost
tangosol.coherence.localport
```

3. Set `localhost` to the IP of the host where the MM server will run and set the `localport` property to a different port than any used by any monitored cluster engines on this host machine.

Configuring the be-engine.tra Files for Hosts with Multiple NIC Cards

When you execute the `be-engine`, it searches for a property file `be-engine.tra` in the working directory. This configuration applies to host machines with multiple network cards (NIC).

Procedure

1. Add the following property to the `be-engine.tra` file on each host:

```
java.property.java.rmi.server.hostname
```

The default value is `localhost`.

2. Set this property to the IP address of the desired NIC.

This IP address must match the value specified in other properties where the host IP is specified.

This property is required so that the engine is discovered by the MM cluster and appears as an active predefined engine. It is also required so that remote users can connect to any agents running on this host using a JMX client like JConsole.

TIBCO BusinessEvents DataGrid WKA Discovery

If you use the TIBCO BusinessEvents DataGrid cache provider, and you have configured the monitored project to discover cluster members using well-known addresses (WKA) then you must make some additional changes to the monitored project CDD so that MM can monitor and manage the cluster.

For more details about WKA discovery in a TIBCO BusinessEvents DataGrid cluster see the sections "Datagrid Discover URL" and "DataGrid Listen URL" in *TIBCO BusinessEvents Configuration Guide*.

Configuring the Project's CDD for Cluster Management

If the monitored project is configured to discover cluster members using well-known addresses (WKA), make some additional changes to the monitored project CDD so that MM can monitor and manage the cluster.

Procedure

1. Open the project to be monitored in TIBCO BusinessEvents Studio and open its CDD file in the CDD editor.

This CDD must be copied for use as the master CDD. See [Project Master and Deployed Locations of CDD and EAR Files](#) to understand the purpose of these copies of the CDD.

2. Add the following property to the cluster properties sheet.

```
be.mm.cluster.as.listen.url MMHostIP:Port
```

Specify the IP of the computer hosting the MM server, and an unused port.

3. Add the value of the `be.mm.cluster.as.listen.url` property to the list of addresses in the `be.engine.cluster.as.discover.url` property. The discovery property should be set at the cluster level (so the value is identical for all potential cluster members).

The discovery URL for well-known address configuration uses the following format:

```
tcp://ip:port[;ip:port]*
```

4. Click **Save**.

MM Console Properties Reference

Values for the MM console properties are configured in the file located at `BE_HOME/MM/web-root/app_config.xml`.

MM Console Configuration Properties

Property	Notes
<code>debugMode</code>	Set to <code>true</code> to enable more detailed error messages. Default is <code>false</code> .

Property	Notes
Demo Mode	<p>If demo mode is enabled, chart updates are made with fake random values based on the most recent value.</p> <p>Default is <code>false</code>.</p>
updateInterval	<p>Defines the time interval (in seconds) between two consecutive calls from MM Console to the MM server. The UI is refreshed after each update interval: the panes and tables with statistics are populated with the newly received data, and the topology tree is updated with the last state of the cluster.</p> <p>Default is 5.</p>
failedPaneThreshold	<p>Maximum ratio of failed pane updates to number of displayed panes, before a system crash is assumed. If the number of failed panes exceeds the threshold, an error displays in the console: <code>Lost connection to data server</code>. Click OK to log out.</p> <p>Default is 0.2.</p>
logoURL	<p>Path to the image file for the company logo (or other image as desired). The image file must be stored within the <code>BE_HOME/MM/web-root/</code> folder. The <code>logoURL</code> value is the relative location of the image file within the <code>web-root</code> folder. For example, if the image is in this location: <code>web-root/images/logo.jpg</code>, then the value of <code>logoURL</code> would be <code>images/logo.jpg</code>.</p> <p>The image displays in the upper left corner.</p> <p>The images size must be no more than 32 by 32 pixels.</p>
chartStyles	<p>You can configure preferences such as colors used for various chart elements. Follow the documentation in the file for each element.</p>

MM Metrics and Features Configuration

The MM component provides a console that enables you to monitor the status of deployed TIBCO BusinessEvents engines and perform management tasks.

You can configure thresholds and other settings for the various health metrics and alerts to suit your needs. You can also configure actions to take based on alerts or health level status values. These configuration tasks are done in the CDD file for MM, `MM.cdd`.

See the following sections for details:

- Configure alerts:
 - [Configuring Alerts](#)
 - [Alert Configuration Reference](#)
 - [Path to an Alert Metric Value \(and a Reference Value\)](#)
 - [Pane Types Reference for Alert Configuration](#)
- Configure health metric rules:
 - [Health Metric Rules](#)
 - [Health Metric Rules Configuration](#)
 - [Health Metric Rule Configuration Reference](#)
- Configure actions to take upon triggering of an alert of change in health level:
 - [Action Configuration](#)
 - [Action Configuration Reference](#)

Alert conditions are met by comparing a specified metric value with a reference value. The reference value can be a constant, or it can be another value in the same pane. As an example of a constant reference, you can configure an alert to trigger when the number of deadlocked threads exceeds a certain number. As an example of a reference that is another value in a pane, you could configure an alert to trigger when Used memory exceeds 95% of Max memory.

Each alert has an alert level (`critical`, `warning`, and `normal`), and a configurable message. Messages use the Java Message Format syntax. You can set up three alerts of different severity for the same metric, using different threshold values.

Severity is defined in terms of the following three levels:

- Critical (red bulb icon)
- Warning (yellow bulb icon)
- Normal (green bulb icon)

Alerts are viewable in the System Alerts pane of the Cluster Overview. See [Cluster Overview](#) for an example.

The presence of alerts can also be used to define the health level value for any monitored entity. See [Health Metric Rules Configuration](#).

Configuring Alerts

Alert configuration tasks are performed in the `MM.cdd` file.

See [Alert Configuration Reference](#) for more details on the settings referenced in this procedure.



- Ensure that condition settings in different alerts do not overlap with each other, so that it is clear which alert to use in all cases.
- As with any procedure that changes files, ensure the emonitor project is backed up before you edit its CDD file.

Procedure

1. Import the emonitor project into your workspace and open the CDD file for editing.
2. Open the MM.cdd in the Cluster Deployment Descriptor editor and select the **Agent Classes** tab.
3. Select **mm-class (Monitoring & Management) > Alert Configurations**.
4. Click **Add** or select an existing alert configuration.
5. Configure the fields as explained in [Alert Configuration Reference](#)
6. Click **Save**.
7. If you have finished configuration, start the emonitor project using the modified CDD file. This starts the MM server.

Alert Configuration Reference

Use this reference to configure alerts.

MM CDD mm-class Agent Class Alert Configuration

Property	Notes
Alert ID	
	An ID for this alert.
Condition Settings	
	Ensure that condition settings in different alerts do not overlap with each other, so that it is clear which alert to use in all cases.
Path	
	Enter the cluster path that defines the scope of this alert. Scope is defined in terms of cluster levels. See Cluster Member Paths for details on specifying this value.
Alert Value	
	A structured path which points to the metric value used for the alert. It can be a series in a chart pane, or a column in a table pane. Elements of the path are as follows: <i>PaneType/SeriesName/CategoryValue/ValueIndex</i> See Path to an Alert Metric Value (and a Reference Value) for details on this setting.
Reference Value	

Property	Notes
	<p>A constant value, or a partial structured path which points to a metric value in the same pane as the metric value specified in the Name field.</p> <p>The partial path is specified as:</p> <p><i>SeriesName/CategoryValue/ValueIndex</i></p> <p>It is appended to <i>PaneType</i> part of the path provided in the Name field.</p> <p>For example, if Name specifies <i>/memory/used</i>, then Reference might specify <i>max</i>, which is another series name in the pane type <i>memory</i>.</p>
Threshold	
	<p>Expressed as a percentage. Enter a value between 0 and 100 as desired.</p> <p>When the value of <i>Name</i> meets or exceeds the <i>Threshold</i> percentage of the value of <i>Reference</i>, the alert is triggered.</p>
Projection Properties	
Severity	
	The severity of the alert. Possible values are: <i>critical</i> , <i>warning</i> , and <i>normal</i> .
Message	
	<p>Message to display when this alert is triggered. The message string can optionally contain variables.</p> <p>For example:</p> <p>"{0}'s garbage collection time {2} for {1} has crossed 10% of {3} up time"</p> <p>See Specifying the Alert Message for more details.</p>

Path to an Alert Metric Value (and a Reference Value)

When you configure an alert, specify two values to compare, and specify a threshold.

The Alert Value

This field provides the path to the metric you are interested in being notified about. The path is structured as follows:

PaneType/SeriesName/CategoryValue/ValueIndex

The Reference Value

This field provides a comparison value. It can be a constant (such as a threshold number) or a different metric value on the same pane. Because it is on the same pane, you do not have to provide the full path. The path for reference is structured as follows

SeriesName/CategoryValue/ValueIndex

Elements Used to Specify the Path to an Alert Metric Value

Path element	Notes
Pane Type	<p><i>PaneType</i>/. . .</p> <p>The specific pane type for the selected element type.</p>
Series Name	<p><i>PaneType/Series Name</i>/. . .</p> <p>To set an alert on a value in one series of a multi-series chart, specify the series name that appears in the chart in the <i>SeriesName</i> position of the path.</p> <p>In a table pane or a single series chart pane, use the value <code>\$default</code>. For example: <code>gc/\$default/* /2</code></p>
Category Value	<p><i>PaneType/SeriesNameCategoryValue</i>/. . .</p> <p>To set an alert on a specific category of information in a chart or table, specify its name in the <i>CategoryValue</i> position of the path.</p> <p>In a chart</p> <p>Each chart element, such as a bar or a line, represents a category of information. For example, in the Thread Pool Best Performers chart each thread is a category (shown as a green bar). Use the name that appears for the chart element, or in the tooltip if the full name does not display.</p> <p>In a table</p> <p>Each row represents a category of information. The value in the first (left-most) column of the row is the name of the category. Use the name of the first column as the category value.</p> <p>Not used for time-based panes.</p> <p>To use all category values, use an asterisk (*) as a wild card character in the <i>CategoryValue</i> position. For example, <code>"gc/\$default/* /2"</code></p> <p>If the category value is not found or is defined as "all," then all the categories in the specified series are considered for condition checking.</p>
Value Index	<p><i>ElementTypePath/PaneType/SeriesName/CategoryValue/ValueIndex</i></p> <p>A specific item of information in a category.</p> <p>Only used for tables (not for charts).</p> <p>For tables, use the index of the column from which the value is taken. The first column is ignored. (It is used as the category value). Indexing begins with the second column from the left. The second column index is 0, the third column index is 1, and so on.</p>

Specifying the Alert Message

An alert message can be a simple text string, or it can use parameters.

Below are two examples showing parameter-based messages:

```
"{0}'s garbage collection time {2} for {1} has crossed 10% of {3} up time"
```

```
"{0}'s used memory {2} has crossed 95% of {3} max memory at {1,date,short}
{1,time,short}"
```

The message supports the following substitutions (using Java Message Format syntax).

Alert Message Parameters

Parameter	Value
{0}	The name of the cluster member that the alert is about. Specified by the <code>condition/getproperty@path</code> attribute.
{1}	The category that the alert is about. Specified by the <code>PaneType/SeriesName/CategoryValue/...</code> part of the <code>condition/getproperty@name</code> path.
{2}	The actual value that is causing the alert to be triggered, as specified in the <code>condition/getproperty@name</code> .
{3}	The reference value, used to calculate whether a metric value is of concern (and the alert is therefore triggered). Specified by the <code>condition/getproperty@reference</code> attribute.

Pane Types Reference for Alert Configuration

This reference shows the element types.

Pane Type Details

Element type	Pane Type Title	Table or Graph	Pane Type ID
Cluster	Cluster Overview	Table	<code>cstats</code>
	System Alerts	Table	<code>sysalerts</code>
Machine	CPU Usage	Chart (time)	<code>cpustats</code>
	Memory Usage	Chart (time)	<code>memory</code>
	Swap File Usage	Chart (time)	<code>swap</code>
Process	CPU Usage	Chart (time)	<code>cpustats</code>

Element type	Pane Type Title	Table or Graph	Pane Type ID
	Memory Usage	Chart (time)	memory
	Running Threads	Chart (time)	rthreads
	Deadlocked Threads	Chart (time)	dthreads
	Garbage Collection	Table	gc
All Agents	Running Threads	Chart (time)	rthreads
	Deadlocked Threads	Chart (time)	dthreads
	Garbage Collection	Table	gc
	Thread Pool Best Performers	Chart	bestpool
	Thread Pool Worst Performers	Chart	worstpool
	Thread Pool Usage	Chart (time)	tpool
	Job Queue Best Performers	Chart	bestjqueue
	Job Queue Worst Performers	Chart	worstjqueue
	Job Queue Usage	Chart (time)	jqueue
Inference Agent	Locks Held	Chart (time)	locks
	RTC Statistics	Chart (time)	rtcstats
	Worst Rule Performers	Chart	worstrules
	Best Rule Performers	Chart	bestrules
Query Agent	Entity Count	Chart (time)	entitystats

Element type	Pane Type Title	Table or Graph	Pane Type ID
	Snapshot Query Execution Query Name, Pending, Accumulated	Table	ssqestats
	Continuous Query Execution Query Name, Pending, Accumulated	Table	cqestats

Health Metric Rules

Health metrics are available for cluster members at each level: cluster, machine, process, and agent. The Cluster Overview panel in MM provides information about the overall health of the cluster, and of each of the cluster members.

Health of the cluster is defined in terms of the following three levels:

- Critical (red bulb icon)
- Warning (yellow bulb icon)
- Normal (green bulb icon)

The colored bulb icons are currently used only in the overall cluster health metric and in alerts. The use of icons is not configurable.

Health metric thresholds are set in the `MM.cdd` file, which you edit in TIBCO BusinessEvents Studio.

Note the following main points:

- You can configure health metrics for none, some, or all types of cluster members as desired
- When configuring health level thresholds, you do not have to set thresholds for all three health levels (critical, warning, and normal). Use only the ones that are useful to you.
- When configuring rules for more than one health level for a single member, ensure that the conditions have no overlap so that it is clear which condition sets the appropriate health level.

Cluster Member Paths

The cluster metrics scope is defined using a cluster path: `site/cluster/machine/process/inference`.

A cluster member in this context is a type of cluster node. The path is a hierarchy with specified names for types of the cluster node: `site`, `cluster`, `machine`, `process`. Below the process level, you can specify types of agents.

In the path specifying a type of cluster member, the members are specified as follows:

```
site/cluster
site/cluster/machine
site/cluster/machine/process
site/cluster/machine/process/inference
site/cluster/machine/process/query
site/cluster/machine/process/cache
site/cluster/machine/process/dashboard
site/cluster/machine/process/inference/*
```

Child Cluster Member Paths

In addition to the above values, when you are constructing a Child Cluster Member path in the Health Metric Rule Configuration panel, use a wild card character (*). Specific agent instances cannot be specified.

To reference all agents in the system, use the wildcard character after the process level: *site/cluster/machine/process/**

To reference all agents of a particular type, add the type and then specify the wildcard character: *site/cluster/machine/process/inference/**

Two Types of Thresholds

You can compute health for a cluster member using either of two methods: Health of Child Cluster Members or Number of Alerts.

Health of Child Cluster Members

One method computes the threshold as a percentage of active (or inactive) specified child cluster members. You can optionally specify that only those child cluster members that are themselves at a certain health level are used when computing the threshold. For example, you could set up a threshold such that the overall cluster health level is set to warning when fifty percent or more agents of any type are at health level "Warning."

Number of Alerts

The other method computes the threshold as a number of alerts of a given severity for the cluster member, during a given time period.

The following guidelines are used to decide which method to implement for different cluster members:

- Use Health of Child Members to compute overall cluster health and machine level health.
- Use Number and Frequency of Alerts to compute the health of processes and agents.

Health of Child Members

Thresholds based on the health of child members can use child member health levels or child member activity status (active or inactive), or both. You can also set a threshold value such that the health level of the parent is set only if a minimum percentage of child members satisfies the specified condition.

For example, if you are setting up thresholds for *site/cluster/machine*, you might select *site/cluster/machine/process* as the child member type. You might specify that the health level should be set to warning on the machine level if any process unit on that machine has a health level of warning. Or you might set the health level of a machine to critical if any of its process units is inactive.

You can also use different child members when configuring each health level for a parent member, depending on your need.

Number and Frequency of Alerts

To define the threshold for a cluster member's health level using alerts, you define which alert severity level to use, and the frequency of alerts received during a specified time period.

All alerts of a specified severity defined for the cluster member are counted.

MM begins a count after it receives the first alert for the specified cluster member. After the time specified in Range has elapsed, the application counts the number of alerts of the specified severity were received during this period. If the count meets or exceeds the threshold, the health indicator is changed to the specified health level for this rule.

Health Metric Rule Examples

Rules can be configured to display a health level indicator on a cluster member based on the health levels of its child members.

These rules can be set on any parent cluster member of the specified child members. The parent member is not shown in the examples. The scope of the rule is wider for parent members higher in the cluster member hierarchy.

To set the health level to critical if a single inference agent is inactive, follow these steps:

- Set Health Level to `critical`
- Set Path to `site/cluster/machine/process/inference`
- Set Threshold to 0
- Add a property called `active` whose value is `false`

To set the health level to critical if all agents are inactive

- Set Health Level to `critical`
- Set Path to `site/cluster/machine/process/*`
- Set Threshold to 100
- Add a property called `active` whose value is `false`

To set the health level to warning if fifty percent of agents are inactive, follow these steps:

- Set Health Level to `warning`.
- Set Path to `site/cluster/machine/process/*`
- Set Threshold to 50
- Add a property called `active` whose value is `false`.

To set the health level to critical if all agents are inactive, follow these steps:

- Set Health Level to `normal`
- Set Path to `site/cluster/machine/process/*`
- Set Threshold to 100
- Add a property called `active` whose value is `true`

To set the health level to warning if thirty percent of inference agents have a health level of warning, follow these steps:

- Set Health Level to `warning`
- Set Path to `site/cluster/machine/process/inference`
- Set Threshold to 30
- Add a property called `healthLevel` whose value is `warning`

Examples Using Alerts

Rules can be configured to display a health level indicator for a cluster member based on the number of alerts received in a time window.

In these examples (unlike the child cluster member examples) the cluster member path is shown. The cluster member path is used in both types of rules but is more relevant to display here.

To set the health level to warning if one critical alert is received for a cluster, follow these steps:

- Set Cluster Member Path to `site/cluster`
- Set Health Level to `warning`.
- Set Threshold to `1`
- Do not set Range.
- Add a property called `severity` whose value is `critical`

To set the health level to `warning` if five or more critical alerts are received within a window of 5 minutes, for a query agent, follow these steps:

- Set Cluster Member Path to `site/cluster/process/query`
- Set Health Level to `warning`.
- Set Threshold to `5`
- Set Range to `300000`
- Add a property called `severity` whose value is `critical`

Health Metric Rules Configuration

You can configure health metrics for none, some, or all types of cluster members.

First, specify the cluster member for which a set of rules will apply. Then configure the individual rules. You can use either of the following as the basis of the rule:

- A characteristic of the specified member's child cluster members: either the number that is active or inactive, or their health level.
- The specified cluster member's number and frequency of alerts at a certain severity.

When setting up a health metric rule, put the most severe health level first. Within each `clustermember` element, the MM server examines the `setproperty` element that is closest to the top of the file first. When a health metric threshold for a cluster member is met, the application stops and does not process additional thresholds for that member.

If you configure all three levels, put `critical` first, then `warning`, and finally `normal`.

See [Health Metric Rule Configuration Reference](#) for more details on the settings referenced in this procedure.



As with any procedure that changes files, ensure the emonitor project is backed up before you edit its CDD file.

Setting Up the Health Metric Rule

When setting up a health metric rule, put the most severe health level first.

Procedure

1. Import the emonitor project into your workspace and open the CDD file for editing. See [Importing the emonitor Project for CDD Editing](#) for details.
2. Open the `MM.cdd` in the Cluster Deployment Descriptor editor and select the **Agent Classes** tab.
3. Select **mm-class (Monitoring & Management) > Health Metric Rule Configurations**.
4. Click **Add**. The configuration panel is displayed.
5. In the **Cluster Member ID** field enter a descriptive name to identify this cluster member.
6. In the **Path** field, enter a path to identify the cluster member. For example, `site/cluster`.

7. Click **Add**. Fields to define a health metric rule for this cluster member are displayed.
8. You can configure health metric rules in two ways. See [Two Types of Thresholds](#) for advice on which method to use. Go to one of the following procedures to continue, depending on how you want to configure this rule:
 - [Configuring a Health Metric Rule with the Child Member Health Status](#)
 - [Configuring a Health Metric Using Cluster Member Alerts](#)

Configuring a Health Metric Rule with the Child Member Health status

Use either the number of active or inactive child members or their health level to configure a health metric rule.

Procedure

1. In the **Health Metric Rule ID** field, enter a descriptive name to identify the rule.
2. In the **Health Level** field, select the health level that the rule will indicate.
3. In the **Condition Type** field, select **Child Cluster Member**.
4. In the **Path** field, enter the cluster path of the child cluster member you want to use to compute this health level metric. For example, in computing cluster health you might specify `cluster/machine`, and for machine health, you might specify `cluster/machine/process`.
5. In the **Threshold** field enter the threshold percentage.
If the percentage of child cluster members that match the criteria specified meets or exceeds this threshold, then the health level of the parent cluster member is set to the **Health Level** field value.
6. In the Properties sheet add one or both of the following properties to set the criteria for counting child members:
 - To use the number of active or inactive child members, add a property called `active`. Set the value to `false` to count only inactive members. Set the value to `true` to count only active members.
 - To use the number of child members at a certain health level, add a property called `healthLevel` and set the value to one of `critical`, `warning`, or `normal`.
7. Repeat this procedure to configure this cluster member's thresholds for remaining health levels (critical, warning, or normal).
8. Click **Save**.
9. If you have finished configuration, start the emonitor project using the modified CDD file.
This starts the MM server.

Configuring a Health Metric Using Cluster Member Alerts

Use cluster member's number and frequency of alerts at a certain severity to configure a health metric rule.

Procedure

1. In the **Health Metric Rule ID** field, enter a descriptive name to identify this rule.
2. In the **Health Level** field, select the health level that this rule will indicate.
3. In the **Condition Type** field, select **Notification**.
4. In the **Range** field enter a time period in milliseconds. Or enter 0 to specify no time period.

If the Threshold number of alerts (of the `severity` value) is received in the Range period, the health metric is set to the Health Level value. If you specify zero (0) then the health level is changed after receiving the Threshold number without regard to any time window.

5. In the **Threshold** field enter the threshold number of alerts.
6. In the Properties sheet add a property called `severity`.
As the value specify one of `critical`, `warning`, or `normal`.
7. Click **Save**.
8. If you have finished configuration, start the emonitor project using the modified CDD file. .
This starts the MM server.

Health Metric Rule Configuration Reference

Use the health metric rule configuration reference to configure cluster member settings.

MM CDD mm-class Agent Class Health Metric Rule Configuration

Property	Notes
Cluster Member Settings	
A cluster member is a level in the cluster member hierarchy. It can be set to the entire cluster, a machine, or a process.	
Cluster Member ID	
	Provide an ID for this cluster member.
Path	
	Path that defines the scope of this metric. Scope is defined in terms of cluster levels. See Cluster Member Paths for details on specifying this value. Specify a cluster member only once, then specify all the health metric rules for that member in one set.
Health Metric Rule Configuration Settings	
For each cluster member you add, you define one or more health metric rules.	
Health Metric Rule ID	
	ID for this health metric rule configuration.
Health Level	
	From the drop-down list, select one of the following health levels: <code>critical</code> , <code>warning</code> , or <code>normal</code> . For each cluster member, you can define up to three rules, one for each health level.
Condition Type	

Property	Notes
	<p>Select the condition type used to compute the cluster member's health value:</p> <p>Child Cluster Member Computes the threshold using the health level of specified child cluster members.</p> <p>Notification Computes the threshold as a number of alerts of a specified severity for the cluster member, during a given time period.</p> <p>See Two Types of Thresholds for more details on this choice.</p>
Path	<p>This field appears if you choose Child Cluster Member in the Condition Type field. Enter the cluster path that defines which child cluster members to use in computing this health metric. For example, in computing cluster health you might specify <code>cluster/machine</code>, and for machine health, you might specify <code>cluster/machine/process</code>.</p> <p>See Cluster Member Paths for more on defining paths.</p> <p>You can use any child of the cluster member specified in the Cluster Member ID field of the cluster member node. It doesn't have to be an immediate child, and it doesn't have to be the same child member in rules you configure for the other health levels for this cluster member.</p> <p>In addition, you must do the following to complete configuration:</p> <ul style="list-style-type: none"> • In the Properties sheet add a property called <code>active</code> and set it to true or false. This defines whether the active or inactive state of the cluster members specified in this path is used in computing the status. • Optionally, add a property called <code>healthLevel</code> and set it to <code>normal</code>, <code>warning</code>, or <code>critical</code>. When you do this, the Threshold percentage applies only to those specified child members whose health level matches this setting (and that are active or inactive as specified in the active property). • In the Threshold field specify a percentage. The health level rule is defined as a percentage (as defined in the Threshold field) of all child cluster members of the specified type that are active or inactive (as specified in the <code>active</code> property) and optionally: that are at the specified health level.
Range	
	<p>This field appears if you choose Notification in the Condition Type field. Enter the number of milliseconds to be used as a range. The Threshold count is reset at the end of each range period.</p>
Threshold	

Property	Notes
	<p>When used for a Child Cluster Members Condition Type</p> <p>Defines a percentage. Enter a value between 0 and 100 as desired. When the number of child cluster members that satisfy the rule criteria meets or exceeds the percentage, the health indicator specified for the health level specified in this rule displays.</p> <p>When used for a Notification Condition Type</p> <p>Defines a number of alerts. When the number of alerts <code>notification/property@severity</code> alerts for the enclosing <code>clustermember</code> meets or exceeds this value, within the time period specified in the range setting, the health indicator specified for the health level specified in this rule displays. If either the Threshold or Range settings is not defined, then a single alert for the cluster member causes the health indicator to change.</p>
	<p>Health Metric Rule Configuration Properties</p> <p>The <code>active</code> and health level properties are used only when computing the health level using child cluster members. You can use both <code>active</code> and <code>healthLevel</code> properties in one rule, although there may be few use cases for using both properties.</p>
<code>active</code>	
	<p>Used only when the Condition Type is set to Child Cluster Members.</p> <p>If set to true, then the health metric calculations use only the specified child members that are active.</p> <p>If set to false, then the health metric calculations use only the specified child members that are inactive.</p>
<code>healthLevel</code>	
	<p>Used only when the Condition Type is set to Child Cluster Members .</p> <p>If set, then the health metric calculations use only the specified child members whose health level is as specified in this property.</p>
<code>severity</code>	
	<p>Used only when the Condition Type is set to Notification.</p> <p>Set to the alert severity that you want to use for the health metric rule calculation.</p>

Action Configuration

The MM server can perform actions when alerts are triggered or when health level indicators change. An action can be execution of a command, or sending of an email.

You must configure email settings in order to use the email feature (see [Basic MM Settings in MM.cdd](#) for details).

Commands are executed on the machine or machines where the trigger condition occurs.

Configuring an Action

For the MM server to execute an action, it has to be configured.



As with any procedure that changes files, ensure the emonitor project is backed up before you edit its CDD file.

See [Action Configuration Reference](#) for more details on the settings referenced in this procedure.

Procedure

1. Import the emonitor project into your workspace and open the CDD file for editing.
See [Importing the emonitor Project for CDD Editing](#) for details.
2. Open the MM.cdd in the Cluster Deployment Descriptor editor and select the **Agent Classes** tab.
3. Select **mm-class (Monitoring & Management) > Action Configurations**.
4. Click **Add** or select an existing action configuration.
5. Configure the fields as explained in [Action Configuration Reference](#)
6. Click **Save**.
7. If you have finished configuration, start the emonitor project using the modified CDD file.
This starts the MM server.

Action Configuration Reference

The action configuration reference supplies properties to configure the mm-class (Monitoring and Management).

MM CDD mm-class Agent Class Action Configuration

Property	Notes
Action ID	
	An ID for this action.
Trigger Condition	
Trigger Condition	
	Select the type of condition that triggers this action: Health Level A specified health level of the specified cluster members. Alert An alert of a specified severity raised on any of the specified cluster members.
Path	
	Enter a cluster path. This path defines the cluster members whose health level or alerts trigger this action. See Cluster Member Paths for details on specifying this value.
Severity or Health Level	

Property	Notes
	<p>If you choose Alert as the trigger condition, in this field specify the alert severity that will trigger the action.</p> <p>If you choose Health Level as the trigger condition, in this field specify the health level that will trigger the action.</p> <p>In both cases the possible values are: <code>critical</code>, <code>warning</code>, and <code>normal</code>.</p>
Action Settings	
Action	
	<p>Choose a value from the drop-down list:</p> <p>Execute Command If you choose this option, enter the command in the Command field.</p> <p>Send Email If you choose this option configure the email message in the fields that appear.</p> <p>Note If you choose Send Email then you must also configure the email properties in the mm-class agent properties. See Basic MM Settings in MM.cdd .</p>
Command	
	<p>If you choose Execute Command in the Action field, enter a command that is executed when the action is triggered. You can specify a shell script or batch.</p> <p>In this release, commands are executed on the BEMM server machine.</p>
To, Cc, Subject, Message fields	
	<p>Complete the email fields to define the message that is sent when the action is triggered. See the note in the Action field.</p>

Deployment and Management of Engines with MM

After you have configured your project for deployment, and Monitoring and Management (MM) are connected to the deployment, you are ready to deploy the project and manage which engines are running in it.

After you have completed all the configuration steps explained in [Basic MM Configuration](#), you are ready to use MM.

Starting the MM Server

Start the MM server after the configuration is done.



The machine running MM must be able to access the monitored cluster through the network. It must be in the same network or have access to the network using multicast.

Procedure

1. At a command prompt, navigate to `BE_HOME/mm/bin` and type:
`be-mm.exe -c MM.cdd -u default -n mm MM.ear`
2. Type `be-mm.exe /help` to view usage information.

On Windows, select the following:

Start > All Programs > TIBCO > TIBCOEnv > TIBCO BusinessEvents <version_number> > Start Monitoring and Management Server.

Logging On to MM Console

After the Monitoring and Management Server has started, you can log on to MM Console.

Procedure

1. In a web browser, enter the URL for the console.
The hostname and port are configured in the `MM.cdd`.
By default the URL is: `http://localhost:9000/index.html`.
2. Log on using the user credentials that were configured in the password file or other authentication mechanism you configured for TIBCO BusinessEvents. As shipped, the default credentials are `admin/admin`.

See [User Authentication](#).

Only users with the role `ADMIN` see [User Authorization for Administrator and User Roles](#).

Result

You see Cluster Explorer in the left panel, and the Cluster Overview on the right.

See [Cluster Explorer Nodes](#) for an introduction to the MM console user interface.

When the connection to the MM server is lost, all panel contents are dimmed and an error message displays: `ERROR Lost connection to data server`. Once the server has come back online you may login again.



See [MM Console Properties Reference](#) for details about the property `failedPaneThreshold`. It determines the maximum ratio of failed pane updates to number of displayed panes before a system failure is assumed.

Setting Global Variables in MM

Global variables are defined and set in the TIBCO BusinessEvents Studio project.

Global variables can also be set in the CDD file. If they are defined as deployment settable and service settable, they can be set in MM at deploy time as well. You can set values at the machine level (but not at the engine level).

The global variable settings are appended to the CDD file that is deployed to a machine.



Multiple users can open an MM console on their machines and work with global variables. All users see the global variable overrides that have been saved by any user. However, if user A has the global variable editor open while user B saves a change, user A will not see that change, until he or she clicks **Refresh** (or starts a new console session).

Procedure

1. Log on to MM Console. See [Logging On to MM Console](#).
2. In the Cluster Explorer, select the host node and click **Deploy**.
3. Enter the login credentials that you configured for Openssh and click **Next**.
The global variable names and their default values are displayed.
4. In the Current Value column, replace the current value with the desired override value.



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`.

5. Click **Save**.
6. Click **Refresh** to ensure that your value was the last entered.
If another user enters an override just after you do, their value overrides your value.

Engines with MM

You can deploy engines using MM Console or using the command-line utility, MM-tools.

- For details about deploying in MM Console, see [Deploying Cluster Engines in MM Console](#).
- For information about hot deployment, see [Hot Deployment for Engines with MM](#).

Before deployment, be sure to copy custom function and third-party jars.

If your project has JAR files for custom functions or third-party software, you must 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, update the classpath in the TRA file to include the location.

Deploying Cluster Engines in MM Console

Deploy the engines configured to run on a predefined machine node.

Procedure

1. Log on to MM Console. See [Logging On to MM Console](#).
2. From the Cluster Explorer, select the machine node you want to deploy.
3. Select the icon of the host machine where you want to deploy and click **Deploy**.
The deployment unit that you configured to deploy on that machine in the site topology file deploys.

4. If you want to override any global variables, see [Setting Global Variables in MM](#) for details on how to do it.
5. Verify the login details or provide them (see for details).
6. Click **OK**. The engine or engines configured to deploy to that machine deploy.



To see if an engine or agent is deployed, move the mouse pointer over its name in the explorer panel. A tooltip shows if it is deployed or undeployed.



Deployment time information is saved to a file located under `BE_HOME/mm/deploym` and the last deployment time is displayed in the UI.

Hot Deployment for Engines with MM

You can hot deploy to a running engine deployed by MM.



See [Hot Deployment](#) for more details.

The following prerequisites have to exist in order to hot deploy a running engine deployed by MM:

- You have modified the TIBCO BusinessEvents Studio project and built the EAR file, following the limitations shown in [Modifications Allowed in Hot Deployment](#).
- The deployed processing units that you want to hot deploy to were enabled for hot deployment before they were deployed. See [Enabling Hot Deployment](#).
- The new EAR file has the same name as the existing one.

To hot deploy in MM, replace the master EAR file with the updated one, and then deploy the engines again using MM Console. There is no need to restart the engines.

Remote Engines (PUs) and the MM-tools Utility

You can deploy engines and start and stop remote engines (PUs) at the command line using the MM command line utility named `mm-tools`.

All engines have to be predefined in the site topology file.

Before you can use the `mm-tools` utility you must configure it. Optionally, you can authenticate a user using certificates when performing `deploy` and `remote start` and `stop` operations, which requires additional configuration.

Before you can use the `mm-tools` utility, you must configure the `mm-tools.tra` file to reference the cluster's site topology file.

Prerequisite Configuration

- For deployment, SSH must be running on the remote machine.
- For remote start, the software that MM uses to start a processing unit on remote machines must be running: TIBCO Hawk, PsTools or SSH. See [Software for Remote Start and Deployment](#) for details.
- The site topology file (and its prerequisites) must also be configured correctly. See [Site Topology](#) and sections following for details.

In addition, the `mm-tools.tra` file must be configured as explained in this section.

Configuring the mm-tools.tra File

The mm-tools.tra file is used to reference the cluster's site topology file.

Procedure

1. Open the following file for editing:
BE_HOME/mm/bin/mm-tools.tra
2. In the following property, specify the path to the site topology file used for cluster deployment:
`be.mm.topology.file <path to the site topology file>`
3. For remote start, if the `start-pu-method` in the site topology file is Hawk, and you use non-default values, uncomment and specify the following Hawk properties. Use the values that are configured for the Hawk agent running on the remote machine:

```
#be.mm.tools.Domain=TIBCO Hawk Domain
#be.mm.tools.TIBHawkService=Rendezvous daemon used by Hawk
#be.mm.tools.TIBHawkNetwork=Rendezvous network used by Hawk
#be.mm.tools.TIBHawkDaemon=Rendezvous service used by Hawk
```

If TIBCO Hawk is also used for machine level metrics, the values specified would be the same as those specified in the MM.cdd.

See [TIBCO Hawk Configuration for Machine Level Metrics](#).

4. Save the file.

Using Public Private Key Authentication with mm-tools

Users are authenticated using certificates when performing `deploy` and `remote start` and `stop` operations.

Procedure

1. On the computer hosting the MM server, use a utility to create a public/private key pair. The `ssh-keygen` utility is widely available. Two files are generated. They are referred to as follows:

PK_FILE_NAME: The file containing the private key

PK_FILE_NAME.pub: The file containing the public key

Optionally, you can specify a passphrase.

Place both generated files in the same directory on the computer hosting the MM server. For example, put them in *BE_HOME/mm/certificates*.

2. On computers hosting the to-be-monitored cluster PUs that you want to remotely deploy, start, or stop, copy the contents of the file *PK_FILE_NAME.pub* to the file containing the list of authorized keys for the user who will be logging in remotely.

For example, for SSH using certificates for authentication, the authorized keys file is called 'authorized_keys' and it is stored in the `.ssh` directory of the user who will be logging in remotely, that is, in `~/ .ssh/authorized_keys`.

3. When executing a command with mm-tools, use these new options:

`-pkf` The fully qualified path to the *PK_FILE_NAME* file, that is, the file with the private key.

`-pph` The passphrase, if you specified one when creating the public/private key pair. (One example below shows the `-pph` option.)

Example Commands for Authentication with mm-tools

These commands are used for remote deploy, start, and stop.

Remote Deploy:

```
mm-tools --propFile mm-tools.tra -op deploy -m 100.100.100.101 -pkf BE_HOME/mm/certificates/PK_FILE_NAME
```

Remote Deploy, with Passphrase:

```
mm-tools --propFile mm-tools.tra -op deploy -m 100.100.100.101 -pkf BE_HOME/mm/certificates/PK_FILE_NAME -pph passphrase
```

Remote Start:

```
mm-tools --propFile mm-tools.tra -op start -puc CS -m 100.100.100.101 -pkf BE_HOME/mm/certificates/PK_FILE_NAME
```

Remote Stop:

```
mm-tools --propFile mm-tools.tra -op stop -puc CS -m 100.100.100.101 -pkf BE_HOME/mm/certificates/PK_FILE_NAME
```

Deploying Starting or Stopping a Remote Engine

When deployment is done through SSH, remote start is done using the method configured in the site topology file start-pu-method setting.

```
BE_HOME/mm/bin/mm-tools [-h] [--propFile StartupProperty File] -op [deploy | start | stop] -m MachineHostName [-puc ProcessingUnitConfig] [-user UserName] [-pwd Password>]
```

Deployment example:

```
mm-tools -propFile mm-tools.tra -op deploy -m Acme-PC
```

Remote start example:

```
mm-tools -propFile mm-tools.tra -op start -m Acme-PC -puc CacheServer
```

Remote stop example:

```
mm-tools -propFile mm-tools.tra -op stop -m Acme-PC -puc CacheServer
```

mm-tools Utility Options Reference

mm-tools Utility Options

Option	Description
-help	Displays this help.

Option	Description
<code>-propFile</code>	<p>When you execute <code>mm-tools</code>, 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.</p> <p>You can specify the path and filename of a startup property file explicitly using the <code>-propFile</code> parameter.</p> <p>For example, if you execute the command from a directory other than <code>BE_HOME/mm/bin</code>, then you would generally use <code>-propFile</code> to specify <code>BE_HOME/mm/bin/mm-tools.tra</code>.</p>
<code>-op</code>	<p>Specifies the operation. Two operations are available:</p> <p><code>deploy</code>: The deploy operation is done through SSH.</p> <p><code>start</code>: The start operation is done using the mechanism defined in the site topology file <code>start-pu-method</code> setting.</p> <p><code>stop</code>: The stop operation is done using the mechanism defined in the site topology file <code>start-pu-method</code> setting.</p>
<code>-m</code>	<p>The hostname of the machine where you want to start or deploy an engine.</p> <p>Only hostnames defined in the site topology file can be used.</p>
<code>-puc</code>	<p>Specify the ID of the processing unit configuration (PUC) you want to use for this engine.</p> <p>Only IDs defined in the site topology file can be used.</p>
<code>-user</code>	<p>Optional. If not specified, the user name in the site topology file is used.</p> <p>For the <code>deploy</code> option: The user name used by SSH on the remote machine.</p> <p>For the <code>start</code> option: The user name used to log on to the remote machine for remote start.</p>
<code>-pwd</code>	<p>Optional. If not specified, the password in the site topology file is used.</p> <p>For the <code>deploy</code> option: The password for the user name used by SSH on the remote machine.</p> <p>For the <code>start</code> option: The password for the user name used to log on to the remote machine for remote start.</p>

Monitoring and Management Component (MM) for TIBCO BusinessEvents Cluster

You can monitor the health of a TIBCO BusinessEvents deployment using the TIBCO BusinessEvents Monitoring and Management (MM) component and manage the deployment accordingly.



You can monitor TIBCO BusinessEvents Views Dashboard agents, but not otherwise manage them.

All TIBCO BusinessEvents cluster members: agents, JVMs (processing units) and machines, generate system metrics. They are made available over the network for use by the Monitoring and Management (MM) web-based user interface.

Using MM Console, you can examine all aspects of the cluster, including objects in the cache. At each level of the cluster hierarchy, various panes display metrics relating to that level graphically. You can rearrange and remove panes, and promote panes of special interest to the cluster overview to create a custom dashboard.

Before you can use the Monitoring and Management component, you must do some configuration:

- For information on configuring MM see [Basic MM Configuration](#).
- For information on configuring the metrics used by the charts and tables, see [MM Metrics and Features Configuration](#)

For each level of the cluster hierarchy, various metrics are provided using charts and tables. Later panels in this chapter provide a reference to the metrics available at each level. The charts and tables appear in different panes on the panel or panels (tabs) relating to one level of the cluster hierarchy.

The agent level can have two panels. One shows the same information for all types of agents. For inference and query agents, a second panel shows information specific to the agent type. Cache agents do not have a second panel.



- All charts display data for the previous ten minutes (or less if, for example MM or a JVM has been running less than ten minutes).
- Monitoring and Management polls for updates and refreshes the display periodically. The interval is configurable. See the `updateInterval` property in [Basic MM Settings in MM.cdd](#).

Health Indicators and Alerts

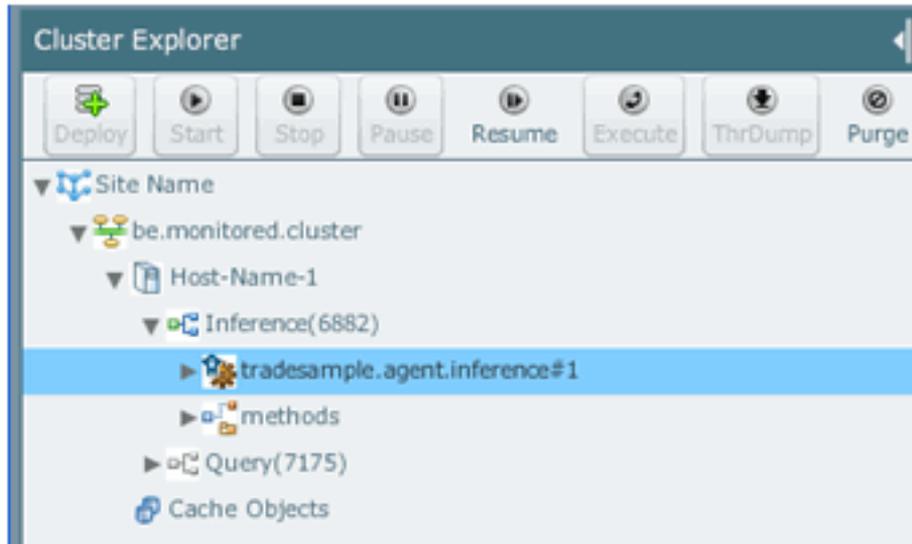
You can configure health indicator thresholds that define whether a value indicates normal functioning, a potential problem (warning), or a critical situation. You can also configure alerts to bring specific problem situations to the attention of system users. These health indicators and alerts are shown in Cluster Overview, providing a dashboard where you can read the health of the entire cluster at a glance.

See [MM Metrics and Features Configuration](#) for configuration details.

Cluster Explorer Nodes

Active and inactive nodes are shown in Cluster Explorer for a quick view of system health.

Active and Inactive Nodes in Cluster Explorer



The Cluster Explorer figure shows the hierarchy of cluster members. Inactive agents (which could be standby agents or failed agents) are dimmed.

The structure of the cluster member hierarchy is as follows:

```

Site
  Cluster
    Machine (host name)
      Process (Processing Unit or Deployment Unit or JVM process ID)
        Agent (inference agent, query agent, or cache agent, dashboard agent,
              or mm agent)
    Cache Objects
  
```

Where:

- Site is the root and has no other purpose in this release.
- Cluster shows the name of the cluster being monitored.
- Machine shows one or more machines within the cluster. They run the cluster processes (process units or engines).
- Process shows each of the JVM processes (TIBCO BusinessEvents engines) running on a machine. The label for a process that was predefined in the topology file is the process unit ID assigned in the file, concatenated with the process ID enclosed in parentheses. The label for an unpredefined process is the JVM process ID.
- Agent lists all agents of each type running in the JVM process.
- The Cache Objects panel shows all the objects stored in the cache, regardless to their physical location in the TIBCO BusinessEvents cluster.

Members of the TIBCO BusinessEvents Cluster

Machines, TIBCO BusinessEvents engines, and agents are all *members* of the TIBCO BusinessEvents cluster.

Predefined and Unpredefined Members

Engines that are not defined in the site topology file are known as unpredefined engines. There are some differences between predefined and unpredefined engines.

- You cannot start or deploy an unpredefined engine using MM.
- You can “Purge Inactive” members that are unpredefined to remove them from the display. Predefined members always remain in the cluster explorer UI.
- The label for a process that was predefined in the topology file is the process unit ID assigned in the file, concatenated with the process ID enclosed in parentheses. (The label for an unpredefined process is the JVM process ID.)

Note that if you start a predefined TIBCO BusinessEvents engine at the command line (outside of MM) and you use a different JMX port from the one specified in the topology file, the engine starts as an *unpredefined* engine.

Inactive Members

Part of cluster health is checking to see that all members are running. When a member becomes inactive, Cluster Explorer and other parts of the MM Console displays a visual indicator. Standby agents in a fault tolerant group display as inactive, as well as cluster members that have stopped operating.

How Inactive Members Display

In Cluster Explorer, the icons for inactive members display in a dimmed state.

If a machine is inactive, processes and agents on that machine are also marked as inactive. Similarly, if a process is inactive, agents running in that process are also marked as inactive.

When a cluster member is inactive, you can still view the last available data in the panel for that member, but overlaid with a gray panel with the label `Entity Inactive`.

When a cluster member is inactive, and a pane relating to that member has been promoted to the Cluster Overview panel, the pane displays in gray with a message:

`Entity Inactive`

The difference in the display inactive member’s promoted pane alerts you to the fact that the member is inactive.

How Inactivity is Determined

Inactivity is determined by the unsuccessful return of a health ping. Health pings are set up for machines and processes only. Process pings use JMX. Machine pings use TIBCO Hawk. If TIBCO Hawk is not available, cluster health status is determined using the health status of the processes (TIBCO BusinessEvents engines) running on each machine.

The property that controls the frequency of the health check ping is

`tibco.clientVar.healthCheckFreq`. See [MM Agent Basic Configuration Reference](#) for details.



You can also configure health thresholds and alerts at any level of the cluster hierarchy. See [MM Metrics and Features Configuration](#) for all configuration options

Cluster Explorer

Using Cluster Explorer you can use node functionality and view information about the node level.

You can use functionality available at various nodes on the left, and you can view information about that node level on the right.

- Expand Cluster Explorer and select the member you want to work with or whose metrics you want to see. Metrics display on the right.
- Click an inactive cluster member to display the last available health metrics for that member.
- Click the minimize button in the Cluster Explorer title bar to minimize the explorer pane.

You can also use the MM-tools utility to start and stop engines at the command line.

Managing Engines

You can start, stop, pause, or resume an engine.

Procedure

1. From Cluster Explorer, select the engine you want to start, stop, pause, or resume. (You resume a paused engine.) You cannot start an unpredefined engine.
2. Click the appropriate icon: **Start**, **Stop**, **Pause**, or **Resume**.
3. Verify the login details and click **OK**.



Stop Cache Nodes Last.

When you stop a cache node, all running inference nodes also appear as inactive in MM and cannot be stopped using MM. This is the expected behavior.

When you restart all cache nodes, the running inference nodes will appear as active again after some time. You may have to re-log on to MM Console before the display is correct.

When you stop all engines in a cluster, make sure you stop the cache nodes after stopping all other types of nodes.

Purge Inactive Unpredefined Processes

You can purge inactive processes.

Procedure

- Click the **Purge** icon.
The Cluster Explorer view is cleaned.

For configuration related to the purge feature see [MM Metrics and Features Configuration](#).

Viewing Monitored Objects

You can view the monitored object details.

Procedure

1. Select the Monitored Objects node from the Cluster Explorer.
The Cache Overview page is displayed in the right pane.
2. The details of various processes are displayed.
For example, name of the entity, count, gets, puts, and so on.

Executing a Method

You can execute methods from the Cluster Explorer.

Procedure

1. Expand the **methods** node to the level at which you want to execute the method, cluster, process or agent.
You will see the method group nodes.
2. Expand the desired method group node, select the method you want to execute on the running engine or engines, and click the **Execute** icon.
3. Specify the argument or arguments according to the dialog that appears, and click **Execute**.
Tooltips explain the values required for the arguments.

Thread Analyzer Reports

Thread analyzer connects to hosts through JMX ports and collects the thread dump for each host.

Thread Analyzer analyzes the thread dump to provide information such as the following:

Thread dump compression

Threads with same stack trace are compressed into one to provide a compressed thread dump report.

Deadlock analysis

A resource allocation graph identifies the deadlocks in each of the thread dumps. Thread analyzer creates a wait-for graph for a given set of stack traces and analyzes the graph to identify Circular Wait Conditions (CWC).

Thread Dump Summarization

Provides a detailed call flow summarization of the thread dump.

The name of the thread analyzer report uses the format: *IPAddress_Port_x.y.log*. The y element is the number of the file, 0-9. You can generate up to ten log files for each set of reports for a machine, and the default size of each file is 10MB.

The x element is used if you stop and start again. It is a number used to distinguish each set of reports, when multiple sets of reports are generated for the same host.

Generating Thread Analyzer Reports

You can generate thread analyzer reports.

Procedure

1. From the Cluster Explorer, select the process whose threads you want to analyze and click the **ThrDump** icon.
2. Enter the details as shown below:

Option	Description
Host name and IP	Displays the name and address of the machine whose thread dump you want to get and analyze. (Thread Analyzer is always started on the server. Only a remote JMX connection is established with this host to obtain the thread dump). When not provided, it is assumed the host is localhost. MM connects to the JMX port configured in the site topology file.
User name and password	Enter the credentials (if any are required) used to connect to the JMX server running on the target machine. These are neither the JMX credentials nor those used to log on to the machine.
Report Directory	Specify where on the MM server to save the Thread Analyzer report. If blank or incorrect, the thread report is saved here by default: <i>BE_HOME/mm/logs/thread-analyzer/</i> .
Time Interval	The interval in seconds between thread dumps. For example, if you enter 10, a thread dump is obtained every 10 seconds.

3. Click OK.

Panels and Panes

The Cluster Overview panel (also called a tab) consists of three panes promoted from a lower level display.

One of the promoted panes indicates an agent is inactive.

Cluster Overview Panel

The screenshot displays the Cluster Overview Panel. On the left is the Node Explorer showing a hierarchy: Site Name > BE Application Cluster 1 > Machine_1 > 6422 > Inference Agent > Query Agent. The main area contains a Cluster Overview table, System Alerts, and two metric panes.

Type	Total	Active	Inactive	Normal	Warning	Critical
machine	1	1	0	1	0	0
process	2	2	0	2	0	0
inference	1	1	0	1	0	0
query	1	1	0	1	0	0
cache	0	0	0	0	0	0

Cluster Health: ●

Severity	Node Name	Node Type	Message	Received
●	nodeX	type1	System out of Memory	Tue Mar 10 2009 03:54:08 PM
●	nodeY	type2	Message Timeout	Tue Mar 10 2009 03:55:08 PM
●	nodeZ	type3	Message Timeout	Tue Mar 10 2009 03:56:08 PM

RTC Statistics - Inference Agen... (Line graph showing Latency and Throughput over time from 15:55 to 16:03)

Running Threads - Inference Age... (Line graph showing Threads over time from 15:55 to 16:03)

MM Metric Panes

When you navigate to different levels of the cluster hierarchy using **Cluster Explorer**, appropriate sets of panes display.

This section explains how to use the controls in a pane's button bar. This information applies to panes in general, not just those that display in the cluster overview.

- To enlarge a pane, click the **Expand**  button in the title bar of the pane.
- To promote a pane to the **Cluster Overview** panel (tab), click the **Promote**  button in the title bar of the pane.

In this way you can create a custom dashboard. The pane continues to display in its original location also.)

- To remove a promoted pane from the Cluster Overview tab, click the **Demote**  button.
- To remove a pane from a tab, click the **Remove**  button in the title bar of the pane.

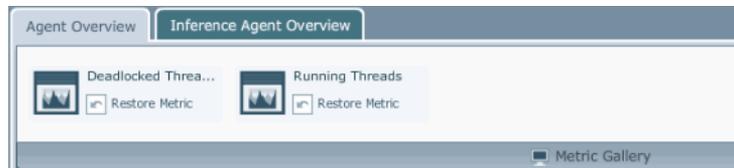
The pane is hidden in the Metric Gallery. The Metric Gallery appears as a bar you can click to open.

Metric Gallery



- To restore a removed pane, click the **Metric Gallery** bar in the appropriate tab and click **Restore Metric**. It is not available on the cluster overview tab.

Restore Metric



Cluster Overview

The Cluster Overview displays summary information about the health of the cluster.

Cluster Overview Pane

Type	Total	Active	Inactive	Normal	Warning	Critical
machine	1	1	0	1	0	0
process	2	2	0	2	0	0
inference	1	1	0	1	0	0
query	1	1	0	1	0	0
cache	0	0	0	0	0	0

Cluster Health: ●

Severity	Node Name	Node Type	Message	Received
●	nodeX	type1	System out of Memory	Tue Mar 10 2009 03:54:08 PM
●	nodeY	type2	Message Timeout	Tue Mar 10 2009 03:55:08 PM
●	nodeZ	type3	Message Timeout	Tue Mar 10 2009 03:56:08 PM

The overall cluster health is shown in one indicator using the red, yellow, or green icon.

Various other metrics display for each type of item in the cluster: machines, processes, and each type of agent.

See [Cluster Level Metrics](#).

Thresholds for normal, warning, and critical health metrics are configurable.

See [Health Metric Rules Configuration](#).

Cluster Level Metrics

Metrics for cluster items: machines, processes, and agent types.

Cluster-Level Metrics

Metric	Notes
Total	Total number of cluster members of this type (machines, processes, agents).
Active	Total number of active cluster members.
Inactive	Total number of inactive cluster members.
Normal	Total number of cluster members whose health is below the threshold set for Warning.

Metric	Notes
Warning	Total number of cluster members whose health is above the threshold set for Warning and below the threshold for Critical.
Critical	Total number of cluster members whose health is above the threshold set for Critical.

System Alerts Pane

The System Alerts pane displays only if any system alerts have been triggered. It shows (up to) the last 25 alerts generated by MM while monitoring the cluster.

The colored icons in the Severity column indicate the severity level of that particular alert.

- **Member Name** displays the name of the specific cluster hierarchy element that triggered the alert, for example, a process ID for a JVM or an agent name and ID.
- **Member Type** displays the type of element, for example, inference agent, process, machine, and so on.

You can define and configure the alerts you are interested in. For each alert, specify the metric value of interest, the threshold that triggers the alert, the severity level, and a message.

See [Configuring Alerts](#).

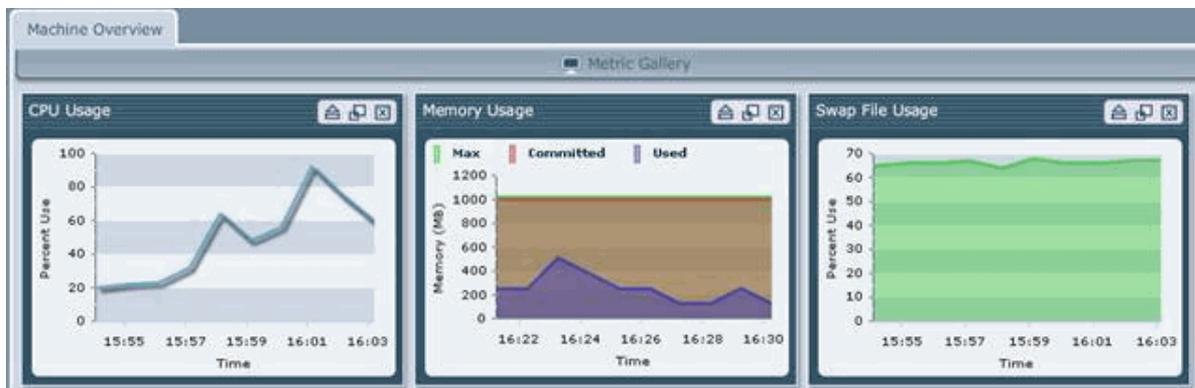
Machine Overview

Machine level metrics are visible only if a TIBCO Hawk domain has been configured.

The Metric Gallery (shown as a gray bar above the panes) holds any panes you remove from the panel to keep your display uncluttered.

See [Panels and Panes](#) for details.

Metric Gallery



See [Machine Level Metrics](#) for more details.

Machine Level Metrics

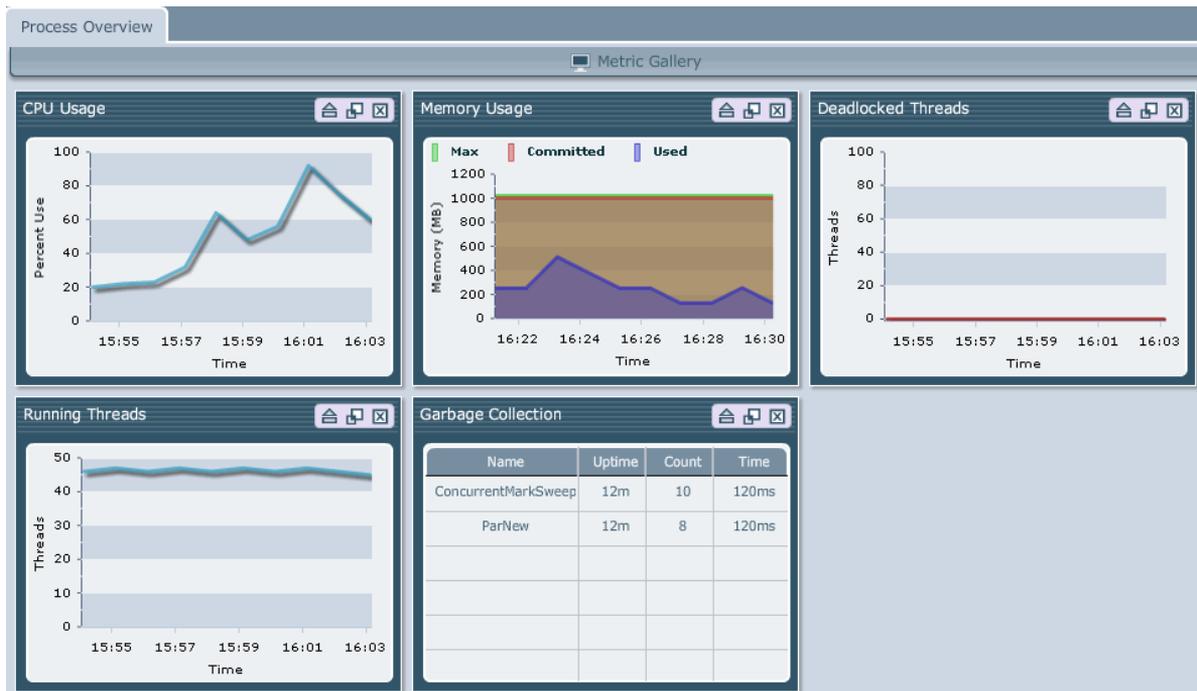
This reference supplies the metrics for a machine.

Machine-Level Metrics

Metric	Notes
CPU Usage	CPU usage, as a percentage, over time.
Memory Usage	Available memory, in megabytes, over time.
Swap File Usage	Available swap file (page file) usage, in megabytes, over time.

Process Overview

You can monitor TIBCO BusinessEvents processes using the TIBCO BusinessEvents Monitoring and Management (MM) component.



The IBM JRE does not provide information on process level CPU usage, resulting in the following limitations relating to the AIX operating system:

When MM server runs on AIX

No process level CPU usage metrics display for any process (no matter what operating system is running on the host machine).

When discovered processes run on AIX

Even when MM server is not running on AIX, no process level CPU usage metrics display for those processes.

Process-level metrics show information at the JVM level, that is at the TIBCO BusinessEvents engine level. See [Process Level Metrics](#) for more details.

Process Level Metrics

Process level metrics show information at the JVM level.

Process Level Metrics

Metric	Notes
CPU Usage	Percentage of CPU used by this process.
Memory Usage	Max, Committed, and Used memory, in megabytes.
Deadlocked Threads	The number of deadlocked threads in the process.
Running Threads	The number of threads in the process, including system threads.
Garbage Collection	<p>For each garbage collector running in the process, metrics shown are:</p> <ul style="list-style-type: none"> • The total up time of the process • The number of objects garbage collected • The cumulative time spent in garbage collection overall.

Agent Overview

You can monitor TIBCO BusinessEvents agents using the TIBCO BusinessEvents Monitoring and Management (MM) component.



Running Threads, Deadlocked Threads, and Garbage Collection are process-level metrics.

Agent Reference

Agent overview is common for all agent types.

Agent Overview (Common) Metrics

Metric	Notes
Running Threads	The number of threads in the process, including system threads.
Deadlocked Threads	The number of deadlocked threads in the process.

Metric	Notes
Garbage Collection	<p>For each garbage collector running in the process, the metrics shown are:</p> <ul style="list-style-type: none"> • The total up time of the process • The number of objects garbage collected • The cumulative time spent in garbage collection overall
Thread Pool Best Performers	<p>The TIBCO BusinessEvents-specific thread pools in the agent, sorted by number of active threads. The fewer running threads in a pool, the better its performance.</p> <p>(If there are only a few thread pools running in an agent, the best and worst performer charts are a mirror image of each other.)</p>
Thread Pool Worst Performers	<p>The TIBCO BusinessEvents-specific thread pools in the agent, sorted by number of active threads. The more running threads in a pool, the worse its performance.</p> <p>(If there are only a few thread pools running in an agent, the best and worst performer charts are a mirror image of each other.)</p>
Thread Pool Usage	<p>Each line represents one thread pool. Usage is in terms of the number of threads in use in each pool. The pane shows a trend of the number of busy (used) threads over time. Four default threads are:</p> <p><code>\$default.be.mt\$</code>: The general thread pool controlled by the property <code>com.tibco.cep.runtime.scheduler.default.numThreads</code></p> <p><code>CacheCluster</code> handles the Agent and other Cache membership and other properties.</p> <p><code>CommonScheduledWorkManager</code>: Any Scheduler created with a single thread uses this default thread pool.</p> <p><code>CommonWorkManager</code>: Any WorkManager created with a single thread uses this default thread pool.</p>
Job Queue Best Performers	<p>The best performing TIBCO BusinessEvents-specific job queues. Job queue performance is based on how many jobs are pending in the queue. The fewer the number of pending jobs, the better the performance of the queue.</p> <p>(If there are only a few job queues, the best and worst performer charts are a mirror image of each other.)</p>
Job Queue Worst Performers	<p>The worst performing TIBCO BusinessEvents-specific job queues. Job queue performance is based on how many jobs are pending in the queue. The larger the number of pending jobs, the worse the performance of the queue.</p> <p>(If there are only a few job queues, the best and worst performer charts are a mirror image of each other.)</p>

Metric	Notes
Job Queue Usage	The count of active jobs in the TIBCO BusinessEvents-specific job queues.

Inference Agent Overview

You can monitor Inference agents using the TIBCO BusinessEvents Monitoring and Management (MM) component.



Inference Agent Reference

Inference agent reference is provided in the overview panel.

Inference Agent Metrics

Metric	Notes
Locks Held	The number of locks held by the agent.
RTC Statistics	<p>RTC performance in terms of latency and throughput:</p> <ul style="list-style-type: none"> • Latency shows the average time to complete an RTC. • Throughput shows the number of RTCs completed.
Worst Rule Performers	Rules sorted by average execution time. The longer the execution time, the worse the rule performance.

Metric	Notes
Best Rule Performers	Rules sorted by average execution time. The shorter the execution time, the better the rule performance.

Query Agent Overview

You can monitor Query agents using the TIBCO BusinessEvents Monitoring and Management (MM) component.



Query Agent Reference

Query agent reference is provided in the overview panel.

Query Agent Metrics

Metric	Notes
Entity Counts	<p>The number of entities in the query agent local cache:</p> <ul style="list-style-type: none"> Local Cache Entity Count: shows the number of entities in the query agent local cache. Incoming Entity Count: shows the number of entities arriving into the local cache.
Continuous Query Execution	<p>Shows metrics for the first ten continuous queries to be registered (only the first ten are shown, for performance reasons):</p> <ul style="list-style-type: none"> Pending: shows the number of cluster messages received by the query that are pending processing. Accumulated: shows the number of real-time cache changes that are pending while the query is still processing continuous query messages.

Metric	Notes
Snapshot Query Execution	<p>Shows metrics for the first ten snapshot queries to be registered (only the first ten are shown, for performance reasons):</p> <ul style="list-style-type: none"> • Pending: shows the number of cluster messages received by the query that are pending processing. • Accumulated: shows the number of real-time cache changes that are pending while the query is still processing continuous query messages.

Ontology (Cache Objects) Overview

The Cached Objects table displays data currently only when Coherence is the cache provider.

Ontology Overview										
Cached Objects										
Nam	Count	Gets	Puts	Get Time	Put Time	Hit Ratio	Max	Min	Expiry Delay	
be.gen.DataGene	1	0	3640	0	0	0	2147483647	1610612735	0	
be.gen.EventsAnc	0	0	0	0	0	0	2147483647	1610612735	0	
be.gen.EventsAnc	0	0	0	0	0	0	2147483647	1610612735	0	
StateTimeoutEver	0	0	0	0	0	0	2147483647	1610612735	0	
ObjectTableIDs	3554	0	3554	0	0	0	2147483647	1610612735	0	
be.monitored.clus	4	7956	6	0	0	100	2147483647	1610612735	0	
WorkList	0	0	0	0	0	0	2147483647	1610612735	0	
be.gen.ConceptM	1769	585	2354	0	0	100	2147483647	1610612735	0	
be.gen.EventsAnc	0	0	0	0	0	0	2147483647	1610612735	0	
be.gen.DataGene	0	0	0	0	0	0	2147483647	1610612735	0	
ClusterLocks	0	0	0	0	0	0	10000	7500	2000	
ObjectTableExtID	3554	2	3554	0	0	0	2147483647	1610612735	0	
be.gen.DataGene	0	0	0	0	0	0	2147483647	1610612735	0	
AgentTxn-1	4	0	3640	0	0	0	10000	7500	10000	
be.gen.DataGene	0	0	0	0	0	0	2147483647	1610612735	0	
be.gen.ConceptM	1769	0	1769	0	0	0	2147483647	1610612735	0	
be.gen.ConceptM	15	1768	15	0	0	100	2147483647	1610612735	0	
SequenceManage	0	0	0	0	0	0	2147483647	1610612735	0	
WorkManager	1	3	2	0	0	66.67	2147483647	1610612735	0	

The Ontology Overview panel has one large pane, the Cache Objects pane. It shows a list of all the objects currently in the cache.

Click the column headers to sort the display. You can use a multiple column sort. The primary sort column displays a number 1, and the arrow indicates the sort order. The secondary sort column displays a number 2 and so on.

Ontology Reference

The Ontology reference table shows attributes for each object.

Cache Objects Metrics

Metric	Notes
Name	The class name of the object.
Count	The number of instances of the object in the cache.
Gets	The number of read operations done on the object.
Puts	The number of write operations done on the object.
Get Time	The average time for a read operation.
Put Time	The average time for a write operation.
Hit Ratio	The ratio of "hits" versus "misses", where "hit" is defined as a 'read' for an object existing in the cache
Max	The maximum number of object instances that can be stored in the cache.
Min	The number of units to which the cache will shrink when it prunes. This is sometimes referred to as a "low water mark" of the cache.
Expiry Delay	The time-to-live in milliseconds for cached object instances.

MM Process Methods

Process methods apply at the process level, that is the processing unit in design-time terminology.

If you use the process methods at the cluster level, they affect all processing units in the cluster.

Process Methods

Property	Notes
Channels Group	
ReconnectChannels	Restarts all channels or a single channel.
resumeChannels	Resumes all channels or a single channel.
suspendChannels	Suspends all channels or a single channel.
ResumeDestinations	Resumes one or all of the destinations of one or all of the channels, depending on the arguments provided.

Property	Notes
SuspendDestinations	Suspends one or all of the destinations of one or all of the channels, depending on the arguments provided.
GetChannels	Retrieves channel information.
GetDestinations	Retrieves destination information.
GetSessionInputDestinations	Retrieves the destinations enabled for input in the specified agent.
Engine Group	
StopEngine	Shuts down the processing unit (engine).
GetHostInformation	Retrieves the value of the specified host information property, or of all properties if none specified.
GetNumberOfEvents	Retrieves the total number of events existing in the specified agent or in every agent.
GetNumberOfInstances	Retrieves the total number of instances existing in the specified agent or in every agent.
GetMemoryUsage	Retrieves the engine's memory usage information.
SetLogLevel	Sets the log level to one of the following: FATAL, ERROR, WARN, INFO, DEBUG, ALL, or OFF
Object Management Group	
GetEvent	Retrieves an event with the specified ID from the specified agent or from every agent.
GetInstance	Retrieves an instance with the specified ID from the specified agent or from every agent.
GetScorecards	<p>If URI is not provided: Returns a table with of all scorecards in the specified agent or in every agent.</p> <p>If URI is provided: retrieves scorecard information in the specified agent or in every agent.</p>
Profiler Group	

Property	Notes
StartFileBasedProfiler	Turns on the profiler and starts collecting data for the specified duration. When the time is complete or the Profiler is turned off, profiling data will be saved to a file in a comma separated format.
StopFileBasedProfiler	Turns off the profiler and stops collecting data.
Rule Service Provider Group	
ResumeRuleServiceProvider	Resumes the agents in the processing unit.
SuspendRuleServiceProvider	Suspends the agents in the processing unit.
Working Memory Group Applies only to inference agents. Other agent types are ignored.	
ActivateRule	Activates a rule in the specified inference agent or in every inference agent.
DeactivateRule	Deactivates a rule in the specified inference agent or in every inference agent.
ResetTotalNumberRulesFired	Resets the total number of rules fired to zero for the specified inference agent or for every inference agent.
GetWorkingMemoryDump	Gets the working memory dump for the specified inference agent or for every inference agent.
GetRule	Gets info about the rule with the specified URI for the specified inference agent or for every inference agent.
GetRules	Gets a table listing the rules deployed for the specified inference agent or for every inference agent.
GetTotalNumberRulesFired	Gets a table listing the total number of rules fired for the specified inference agent or for every inference agent.
GetRuleSession	Gets a table listing every active inference agent.

MM Inference Agent Methods

MM Inference Agent is group of methods you can use to manage a deployed cluster.

Inference Agent Methods

Property	Notes
Agent Group	
GetNumberOfEvents	Retrieves the total number of events currently in the agent's Rete network. Note that events with time to live zero (ttl=0) do not persist in the Rete network.
GetNumberOfInstances	Retrieves the total number of concept instances existing in this agent's Rete network.
Resume	Resumes the execution of this agent.
Suspend	Suspends the execution of this agent.
Channels Group	
GetSessionInputDestinations	Retrieves the destinations enabled for input in this agent.
Object Management Group	
GetEvent	Retrieves the event with the specified ID from this agent.
GetInstance	Retrieves the concept instance with the specified ID from this agent.
GetScorecards	If URI is not provided: Returns a table with of all of the scorecards in this agent. If URI is provided: Retrieves the agent's scorecard information.
Profiler Group	
	The performance profiler is primarily used towards the end of the development phase, to check for bottlenecks and refine the project design. For full details about using profiler as well as a detailed reference, see Performance Profiler in <i>TIBCO BusinessEvents Developer's Guide</i> .
StartFileBasedProfiler	Turns on the profiler and starts collecting data for the specified duration. When the time is complete or the profiler is turned off, profiling data is saved to a file in a comma-separated format.
StopFileBasedProfiler	Turns off the profiler and stops collecting data.

Property	Notes
Working Memory Group	
ActivateRule	Activates the specified rule in this agent.
DeactivateRule	Deactivates the specified rule in this agent.
ResetTotalNumberRulesFired	Resets to zero the total number of rules fired in this agent, resetting the statistics.
GetWorkingMemoryDump	Retrieves the working memory dump of this agent.
GetRule	Retrieves information about the rule with the specified URI in this agent.
GetRules	Retrieves a table listing the rules deployed in this agent.
GetTotalNumberRulesFired	Retrieves a table listing the total number of rules fired in this agent since startup or since statistics were last reset.
GetRuleSession	Retrieves a table with the name of this agent.

MM Query Agent Methods

MM Query Agent is group of methods you can use to manage a deployed cluster.

Query Agent Methods

Property	Notes
Agent Group	
Resume	Resumes the execution of this agent.
Suspend	Suspends the execution of this agent.
Channels Group	
GetSessionInputDestinations	Retrieves the destinations enabled for input in this agent.

Enterprise Archive (EAR) Files

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

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.

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**.



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. Instead, the Studio JVM is used to load all the Java classes and resources into memory until the build process is completed. You can choose to use the file-system based compiler to build EAR files by setting the appropriate options.

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. Instead the Studio JVM is used to load all Java classes and resources into memory until the build process is completed. However, you can use the file-system based compiler to build EAR files.

Procedure

1. In TIBCO BusinessEvents Studio, select the menu **Window > Preferences** to open the Preferences dialog.
2. Select **TIBCO BusinessEvents > Code Generation** on the left panel and then set the **Compilation Mode** to **File System**.

When using the Studio Tools utility to build an EAR file, set the option `-lc` to use the file-system based legacy compiler.



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. In BusinessEvents Studio Explorer, highlight the project name, then from the top menus select **Project > Build Enterprise Archive** .
If you see a message asking you to save all project resources, click **Yes**. (This message means an unsaved resource editor is open.)
4. At the Build Enterprise Archive dialog, complete values according to guidelines provided in [Enterprise Archive Reference](#).
5. Click **Apply** to save the configuration details.
To revert to the version already saved, click **Revert**.
6. 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	<p>Name of this EAR configuration. (Not the EAR filename.)</p> <p>Default value is the project name.</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-top: 10px;">  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. </div>
Author	<p>Person responsible for the EAR file.</p> <p>Default value is the currently logged-on user name.</p>
Description	Optional description.
Archive Version	Increments on each build of the EAR. You can also manually enter a version identifier.
Generate Debug Info	<p>Select this check box if you want to use the debugger.</p> <p>Default setting is checked.</p>
Include all service level global variables	Select to include service level global variables.
File Location	Browse to the directory in which you want to store the EAR file and enter an EAR filename.

Field	Description
Delete Temporary Files	<p>Before TIBCO BusinessEvents packages an EAR file, it generates the Java code in a temporary directory. After the files are packaged in the EAR file, then the temporary files and directory are deleted.</p> <p>You can keep the generated Java files, for example to troubleshoot some problem with an EAR file. To do so, clear the Delete Temporary Files check box, and specify where to store the Java files in the Compilation Directory field.</p> <p>Default setting is checked, meaning that temporary files are not saved.</p>
Compilation Directory	If you clear the Delete Temporary Files check box, specify the directory where you want to save the Java files generated during the process of building the EAR file.

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] [-x] [-lc] [-jc] [-o outputEarFile>] -p
studioProjectDir [-pl projectLibrariesFilePath] [-cp extendedClasspath]
```

For example:

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

See [Options for Building an EAR File](#)



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
<code>-core buildEar</code>	Within the <code>core</code> category of operations, specifies the <code>buildear</code> operation for building EAR files.
<code>-h</code>	Optional. Displays help.

Option	Description
-x	Optional. Overwrites the specified output file if it exists.
-lc	Optional. Specifies that the file-based legacy compiler must be used to build the EAR file. By default, the EAR files are built in memory.
-jc	Optional. Include 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 <i>projectDir</i> 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 path to be used, separated by a path separator.
-cp	Optional. Specifies the extended classpath to be used.

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	<p>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 <code>--propFile</code> parameter.</p> <p>For example, if you start the engine from a directory other than <code>BE_HOME/bin</code>, then you would generally use <code>--propFile</code> to specify <code>BE_HOME/bin/be-engine.tra</code>.</p>
--propVar	Used to provide a value for a specified variable. This value overrides any other design-time value. The format is <code>propVar-varName=value</code> . For example to specify the value of the <code>%jmx_port%</code> variable used in TRA files to configure a JMX connection, you might use this: <code>--propVar jmx_port=4567</code> .
-p	<p>Allows you to pass one or more supplementary property files to <code>be-engine</code>. Specify the path and filename.</p> <p>This is not needed very often. See Supplementary Property Files.</p>
-n	<p>Allows you to provide a name for the TIBCO BusinessEvents engine.</p> <p>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.</p>
-d	Starts the debugger service on the engine for remote debugging.
-c	<p>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.</p> <p>The default is <code>default.cdd</code>.</p>

Option	Description
<code>-u</code>	Specify the processing unit ID you want to use for this engine. This ID must exist in the CDD file you reference in the <code>-c</code> option. The default is <code>default</code> .
<i>EAR filename and path</i>	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 <code>tibco.repourl</code> as the EAR file path and name. To use this property, add it to the <code>be-engine.tra</code> 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, or Rule Management Server (RMS), or MM, or Views to start as a Windows NT service.



TIBCO Hawk and TIBCO Rendezvous 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/mm/bin/br-mm.tra` for MM

- `BE_HOME/views/bin/br-views.tra` for Views
2. Add the following environment paths under the common environment variables:


```
tibco.env.RV_HOME=<absolute path where TIBCO Rendezvous is installed>
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)
- `BE_HOME/mm/bin/MM.ear` for MM
- 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/5.1/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/mm/bin/be-mm.exe` for MM
- `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/5.1/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
- `BE_HOME/mm/bin/MM.cdd` for MM
- Absolute path of `ViewsProjectName.cdd` file for Views

For example, for TIBCO BusinessEvents engine for the FraudDetection project

```
tibco.clientVar.CDD=C\:/tibco/be/5.1/examples/standard/FraudDetection/FraudDetection/fd.cdd
```

7. Set the processing unit ID (PUID) for the engine to the PUID that is setup 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 setup 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.



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.



- 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 user name 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 Runtime](#) 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.



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 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` will start. 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 check box. See "Agent and Processing Unit Configuration" in *TIBCO BusinessEvents Developer's 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" check box 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.

Set stack size to 264K on HP-UX Itanium

The default stack size is not sufficient to create the Java Virtual Machine and start the engine on HP-UX Itanium. Edit the `be-engine.tra` file and set the stack size to 264K:

```
java.thread.stack.size=264K
```

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 check box is selected at design time in the Global Variable editor, you can override 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 check box and the Service Settable check box 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 check box and the Service Settable check box 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: 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` check box 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 located in the following folder

```
TRA_HOME/domain/domain_name/datafiles/application_name_root
```

Service and service instance level overrides

These are located in the TRA file generated by 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.



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.)



The only supported transportation option is **local**.

Deploying a Project EAR in a TIBCO Administrator Domain

Procedure

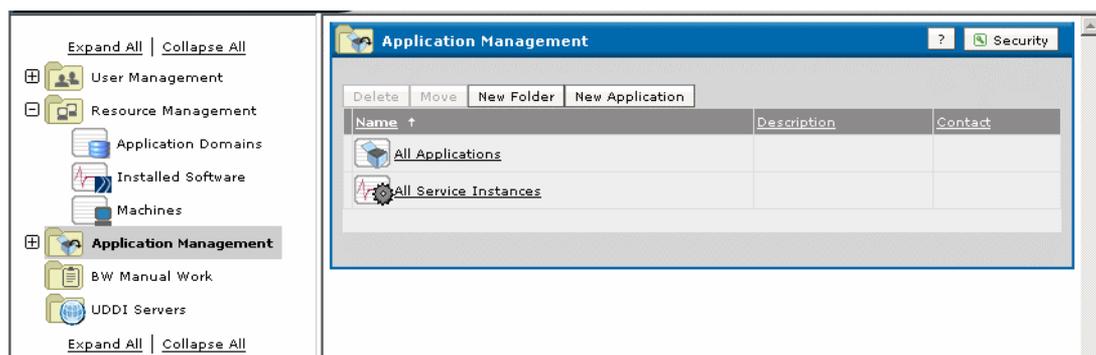
1. Ensure that the following are started on the machine whose engine properties you want to change:
 - TIBCO Administrator service for the administration domain.
 - TIBCO Hawk service for the administration domain.
2. Start the TIBCO Administrator GUI:
 - Windows: **Start > Programs > TIBCO > TIBCO Administrator 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 user name 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 user name 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.



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:

Variable	Default	Value
DepSetVar	thirteen	14
ServSetVar	13	14
MessageEncoding	ISO8859-1	ISO8859-1

TIBCO BusinessWorks and Adapters Deployment Repository Instance

Transport: local

Message Encoding: ISO8859-1

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.



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.

- 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.
 - Service-settable global variables are only available if the Include All Service Level Global Variables check box 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.

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

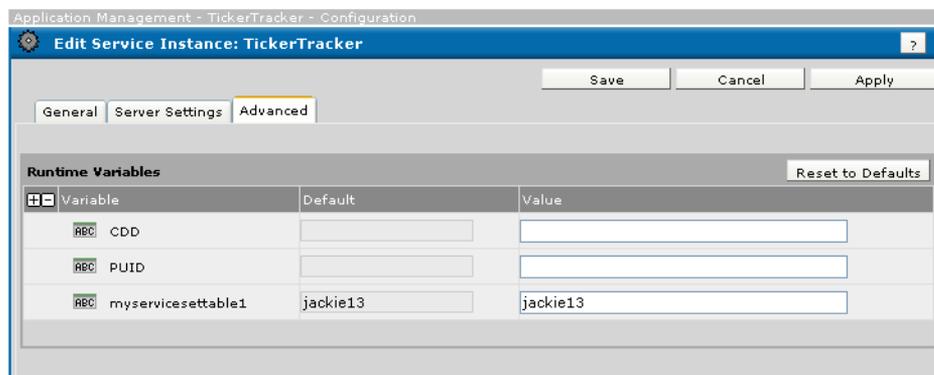
These steps will set project-related settings that apply to deployments on specific machines.

Prerequisites

Make sure you have finished all the steps as described in [Deploying on a Service Level](#).

Procedure

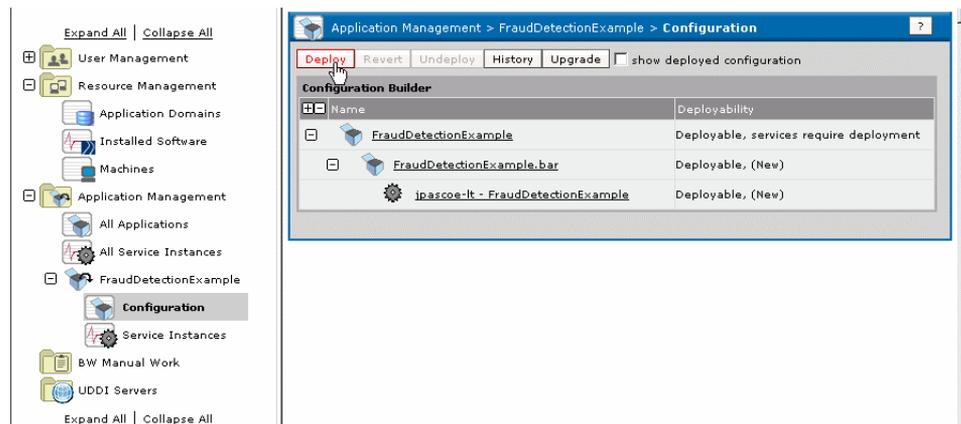
1. At the Configuration Builder panel, select a machine-level entry (`machineName - projectName`). These entries appear below the `application-name.bar` entry. Then select its **Advanced** tab. You see a dialog similar to the following:



Here you can also override any service-settable global variable values as desired. (See [Overriding of Global Variables in TIBCO Administrator](#) for important information.)

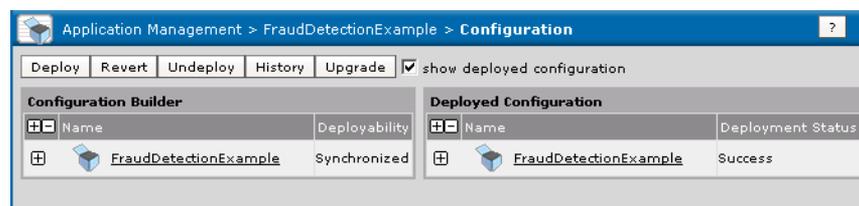
2. Click **Save** when you are done. The Configuration Builder panel appears again. The same project-related options are available here as at the .bar level, but here they apply only to an engine deployed to the selected machine. As desired, enter the name of the CDD file and processing unit (PUID) you 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** check box 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
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

Resource	New	Modify	Delete
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** check box 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** check box.
Repeat for all processing units you want to enable for hot deployment.
See Agent and Processing Unit Configuration in *TIBCO BusinessEvents Developer's 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 the following requirements are met:

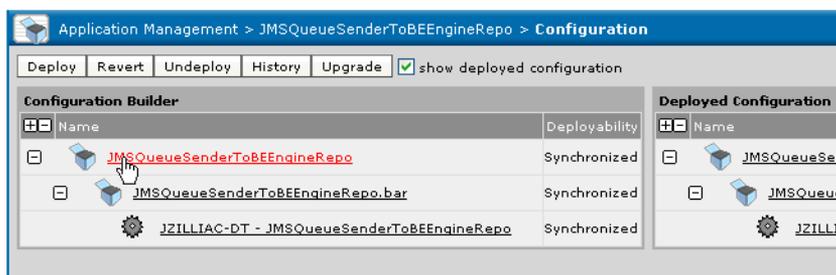
- The processing units that you want to hot deployed were already enabled for hot deployment before they were deployed (see [Enabling Hot Deployment](#)).
- Your project complies with the requirements for deploying to TIBCO Administrator.
- You have modified the TIBCO BusinessEvents Studio project and 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 user name 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 check boxes (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 check box is selected, you see an additional setting, Kill services that haven't stopped after (seconds). It is removed when you clear the check box.)

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 [Hot Deployment Supported Modifications](#) 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. To enable authentication for MM you must also set JMX properties in the `be-engine.tra` files.

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).



For the TIBCO BusinessEvents Monitoring and Management component, the provided JAAS login module is required.

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 `nsrole` 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.

Authentication In Various Components

Authentication is used in components of various TIBCO BusinessEvents products:

TIBCO BusinessEvents Monitoring and Management

JMX MBeans authentication is available but not enabled by default. You can enable it using a JMX property in the `be-engine.tra` file. See [Authentication Configuration](#) for instructions.

File based authentication is enabled by default. LDAP authentication is also supported. To configure the MM authentication mechanism, you set the `be.mm.auth.*` properties in the `MM.cdd` file. See [Authentication Property Reference](#).

This component also uses two predefined authorization roles. See [User Authorization for Administrator and User Roles](#).

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 [Configuring Access Control for a Project](#).

TIBCO BusinessEvents Views

Authentication is available but not enabled by default. The following TIBCO BusinessEvents Views example project is configured for file-based authentication: *BE_HOME/examples/views/TickerTracker*.

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.



For MM authentication, you must also configure JMX properties. See [JMX Properties and To-Be-Monitored Engine TRA Files](#).

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 Monitoring and Management:

- *BE_HOME/mm/project/emonitor* > MM.cdd > mm-class agent class > properties > mm > auth property group
- *ToBeMonitored_Project* > *project.cdd* > Cluster properties > auth property group

For TIBCO BusinessEvents Decision Manager:

- *BE_HOME/rms/project/BRMS* > RMS.cdd > Cluster properties > RMS property group

For TIBCO BusinessEvents Views:

For TIBCO BusinessEvents Views:

- *Your_Project* > *project.cdd* > *dashboard-class agent* properties.

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]`

For MM authentication use the property:

`be.mm.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](#). Familiarity with LDAP is required. Details are not provided in this guide.
- To configure file-based authentication, see [Configuring File-Based Authentication](#).



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.
For MM authentication use the property: `be.mm.auth.file.location`.
2. Locate and open the password file. Its location is specified in the CDD file.
See [step 1](#) 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;
```

For MM authorization roles see [User Authorization for Administrator and User Roles](#)



- 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 Monitoring and Management and TIBCO BusinessEvents Decision Manager. See [Configuring Access Control for a Project](#).

Authentication Property Reference for the TRA File

To avoid conflict with properties used by other components that use authentication, a parallel set of properties (for the TRA and CDD file) is used for MM configuration. These are authentication properties used in the TRA file.

Property	Notes
<code>java.property.be.engine.jmx.connector.port</code>	<p>Specify this property in each relevant engine TRA files to open the JMX connector port for monitoring and management.</p> <p>It is also used for hot deployment of decision tables in TIBCO BusinessEvents Decision Manager (see TIBCO BusinessEvents Decision Manager User's Guide for details).</p> <p>The JMX port is specified as <code>%jmx_port%</code>, when MM is used. See Authentication Configuration for more details.</p> <p>For other components that use this property, you must also specify the value using the <code>%jmx_port%</code> variable, if you also use MM.</p> <p> You can also set the JMX connector port for deployment with TIBCO Administrator using this CDD property: <code>be.engine.jmx.connector.port</code></p>
<code>#java.property.be.engine.jmx.connector.authenticate</code>	<p>Set to <code>true</code> to enable authentication.</p> <p>Set to <code>false</code> (or leave commented) to disable authentication.</p> <p>The default is <code>false</code>.</p>

Common Authentication Properties for the CDD File

To avoid conflict with properties used by other components that use authentication, a parallel set of properties (for the TRA and CDD file) is used for MM configuration. These are common authentication properties used in the CDD file.

Property	Notes
<code>java.security.auth.login.config</code>	<p>Provides the absolute location for the login module configuration used by JAAS. Only advanced users should change this value (additional configuration is also needed).</p> <p>The locations of the provided files are as follows:</p> <p>TIBCO BusinessEvents Decision Manager: <code>BE_HOME/rms/config/security/jaas-config.config</code></p> <p>TIBCO BusinessEvents Views: <code>BE_HOME/views/config/jaas-config.config</code></p> <p>TIBCO BusinessEvents Monitoring and Management: <code>BE_HOME/mm/config/jaas-config.config</code></p>
<code>be.auth.type</code> <code>be.mm.auth.type</code>	<p>Specifies the authentication mechanism. Allowable values are as follows:</p> <p><code>file</code>: File-based authentication. Uses a password file.</p> <p><code>ldap</code>: LDAP-based authentication Uses a pre-existing LDAP setup in use in your environment. Add and configure the properties shown in .</p> <p>Default is <code>file</code></p>
<code>be.auth.file.location</code> <code>be.mm.auth.file.location</code>	<p>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:</p> <p>TIBCO BusinessEvents Decision Manager: <code>BE_HOME/rms/config/security/users.pwd</code>.</p> <p>TIBCO BusinessEvents Views: Create a file for each project. A sample file is provided in <code>BE_HOME/examples/views/TickerTracker/config/tickertrackerusers.pwd</code>.</p> <p>TIBCO BusinessEvents Monitoring and Management: <code>BE_HOME/mm/config/users.pwd</code>.</p>

LDAP Authentication Properties for the CDD File

To avoid conflict with properties used by other components that use authentication, a parallel set of properties (for the TRA and CDD file) is used for MM configuration. These are LDAP authentication properties used in the CDD file.

Property	Notes
<code>be.auth.ldap.type</code>	

Property	Notes
	<p>Use this property to only if you want to use OpenLDAP for LDAP authentication. The property is not required for Oracle directory server or Windows Active Directory server.</p> <p>Set his property to openldap to use the RMS server with OpenLDAP</p>
be.auth.ldap.port be.mm.auth.ldap.port	
	Specifies the port for LDAP authentication.
be.auth.ldap.adminDN be.mm.auth.ldap.adminDN	
	<p>Specifies the base distinguished name (DN) for admin login.</p> <p>For example: cn=Directory Administrators, dc=na, dc=tibco, dc=com.</p>
be.auth.ldap.adminPassword be.mm.auth.ldap.adminPassword	
	Specifies the password for the LDAP administrator DN.
be.auth.ldap.baseDN be.mm.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 be.mm.auth.ldap.roleAttr	
	<p>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</p> <p>Default value is nsroleDN (for Oracle Directory Server).</p>
be.auth.ldap.uidattr be.mm.auth.ldap.uidattr	
	<p>Specifies the name of the attribute used by the LDAP server for user name information. Allowable values are as follows:</p> <p>uid for Oracle Directory Server</p> <p>cn for ActiveDirectory.</p> <p>Default value is uid.</p>
be.auth.ldap.useRoleDN be.mm.auth.ldap.useRoleDN	

Property	Notes
	<p>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.</p> <p>Set this property to false to use only the name of the attribute, which is shown in the notes for the <code>be.auth.ldap.uidAttr</code> property.</p> <p>Default value is <code>true</code>.</p>
<code>be.auth.ldap.objectClass</code> <code>be.mm.auth.ldap.objectClass</code>	
	<p>Specifies the <code>objectClass</code> attribute value for DS.</p> <p>Many object classes can exist, for example, <code>inetOrgPerson</code> on Oracle Directory Server, and <code>user</code> on Active Directory.</p> <p>If search should span all object classes, keep this value empty or specify an asterisk ("<code>*</code>").</p>
<code>be.auth.ldap.dnAttr</code> <code>be.mm.auth.ldap.dnAttr</code>	
	<p>Specifies the name of the attribute that contains the fully qualified name.</p> <p>Default value is <code>distinguishedName</code>.</p>
<code>be.auth.ldap.ssl</code> <code>be.mm.auth.ldap.ssl</code>	
	<p>Specifies a secure connection to the LDAP host is to be established.</p> <p>Default value is <code>false</code>. Set the property to <code>true</code> to enable a secure connection.</p>

Access Control Configuration

Access control is a core product feature used by RMS projects and available in the TIBCO BusinessEvents Decision Manager add-on and TIBCO BusinessEvents WebStudio.

TIBCO BusinessEvents Monitoring and Management also uses two roles (see [User Authorization for Administrator and User Roles](#)).

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.



- 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 user name should not be same. Each user name 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](#) on page 3.



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>
```

```

    </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.



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 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.

Permissions—ALLOW and DENY

The value of the `action` element is one of the key words `ALLOW` or `DENY`.

The value of the key word 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.

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, ways you can add or edit access control files, and where to place the files so they can be used by the RMS, Decision Manager components, and TIBCO BusinessEvents WebStudio.

Examples shipped with the product contain access control files 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.



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.  Users can check out only those resources they are allowed to read.
	update	Update TIBCO BusinessEvents project resources that were checked out earlier. 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).
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.)

Resource Type	Allowable Action Types	(If action is ALLOW) Enables Users to . . .
	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.
RULETEMPLATEINST ANCE	read	View business rules.
	add_inst	Create business rule for the rule template.
	del_inst	Delete business rule.
RULETEMPLATEVIEW	read	Checkout rule template views.

Dockerize TIBCO BusinessEvents

Using the tools and scripts provided in the TIBCO BusinessEvents software, you can run a TIBCO BusinessEvents application inside a Docker container.

Docker provides a way to run applications securely isolated in a container, packaged with all its dependencies and libraries. Your application can run in any environment as all the dependencies are already present in the image of the application. Testing and deployment of this image is simpler as your image is fully portable. Using Docker, you can run many applications that all rely on different libraries and environments on a single kernel as the containers are lightweight and run without the extra load of a hypervisor. Thus, you can get more out of your hardware by shifting the "unit of scale" for your application from a virtual or physical machine, to a container instance.

A BusinessEvents application comprises a common BusinessEvents runtime and project (application) specific BusinessEvents code running inside the BusinessEvents runtime. Thus to dockerize a BusinessEvents application you have to build a BusinessEvents base Docker image as well as the BusinessEvents application Docker image. BusinessEvents base image cannot run standalone as it does not contain application code. You need to run BusinessEvents application specific image which is built on the product base image by adding project specific artifacts to the image.

For complete information on Docker, refer to the Docker documentation at <https://docs.docker.com>.

Key Docker Concepts

Dockerfile

Docker can build images automatically by reading the instructions from a Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.

Docker Image

An image is a filesystem and parameters to use at run time. It does not have a state and it never changes.

Container

A container is a running instance of an image.

Data Volumes

A data volume is a specially-designated directory within one or more containers that bypasses the Union File System.

Data volumes are designed to persist data, independent of the container's life cycle. Docker therefore never automatically deletes volumes when you remove a container, nor will it "garbage collect" volumes that are no longer referenced by a container.

Running TIBCO BusinessEvents in Docker

You can deploy and run a BusinessEvents application in a Docker using the Docker image of the BusinessEvents application.

Procedure

1. Verify that the Dockerfile for TIBCO BusinessEvents is present at `BE_HOME/docker/bin` folder. See [Dockerfile for TIBCO BusinessEvents](#) on page 184.
2. Build the BusinessEvents Docker image using the script file provided by BusinessEvents. See [Building a Docker Image for TIBCO BusinessEvents®](#) on page 186.
3. Generate the BusinessEvents application Dockerfile using the utility provided by BusinessEvents. See [Generating BusinessEvents Application Dockerfile](#) on page 188.

4. Build the BusinessEvents application Docker image using the utility provided by BusinessEvents. See [Building BusinessEvents Application Docker Image](#) on page 190.
5. Run the BusinessEvents application image in Docker. See [Running TIBCO BusinessEvents® Application in Docker](#) on page 191.

Dockerfile for TIBCO BusinessEvents

TIBCO BusinessEvents provides the Dockerfiles for installation of TIBCO BusinessEvents software inside the Docker container.

Dockerfile is a script, which consists of various commands to automatically perform actions on a base image to create a new one. Dockerfiles simplify the process of deployment. The Dockerfiles for various Linux platforms are located at the `BE_HOME/docker/bin` folder. The following Dockerfiles are provided with the BusinessEvents installation:

- `Dockerfile (Ubuntu)`
- `Dockerfile.centos`
- `Dockerfile.debian`
- `Dockerfile.fedora`
- `Dockerfile.opensuse`
- `Dockerfile.rhel`

The Dockerfiles begin with `FROM` command which specifies the image that starts the build process. For example, `FROM centos:latest`.

After the specifying the base image define various other methods, commands and arguments (or conditions), in return, provide a new image which is to be used for creating docker containers. The Dockerfile is then supplied to the Docker daemon to build an image.

Syntax of the Dockerfile commands is:

```
Command argument argument ...
```

For example,

```
# Print "Hello docker!"
RUN echo "Hello docker!"
```

The following sections identify the key commands that setup key configurations for the BusinessEvents Docker image. For more details on each command of Dockerfile, refer to the Docker documentation at <https://docs.docker.com/engine/reference/builder/>.

Environment Variables (ENV Command)

The ENV command is used to set the environment variables (one or more). These variables consist of "key = value" pairs which can be accessed within the container by scripts and applications alike. The usage of the ENV command is:

```
ENV key value
```

The default BusinessEvents Dockerfile setup the following environment variables using the ENV command:

- `CDD_FILE`: Path of the BusinessEvents application CDD file. The application Dockerfile provides this value.
- `EAR_FILE`: Path of the BusinessEvents application enterprise archive. The application Dockerfile supplies this value.
- `PU`: The name of the processing unit to run. The value is provided at the runtime by the user. The default value is `default`.

- `AS_DISCOVER_URL`: Discovery URL of the ActiveSpace. The value is provided at the runtime by the user.
- `ENGINE_NAME`: TIBCO BusinessEvents engine name. The default value is `be-engine`.
- `LOG_LEVEL`: Logging level for BusinessEvents. The value is overridden at run time by the user. The default value is `na`.

```
# BusinessEvents Environment Variables
ENV CDD_FILE no-default
ENV PU default
ENV EAR_FILE no-default
ENV ENGINE_NAME be-engine
ENV LOG_LEVEL na
ENV AS_DISCOVER_URL self
```

Data Volumes (VOLUME Command)

The `VOLUME` command is used to enable access from your container to a directory on the host machine (i.e. mounting it). The usage of the `VOLUME` command is:

```
VOLUME /dir1, /dir2 ...
```

Using data volumes you can persist the data across docker runs. For example, in the default Dockerfile ActiveSpaces Shared Nothing file stores, log file locations, and Rule Management Server directories are configured. The Docker volumes for them are created and all internal file paths are rooted to the specified directories. The following values are overridden in the CDD which is built into the application image

- `/mnt/tibco/be/logs` - Directory where log files are stored.
- `/mnt/tibco/be/data-store` - Directory where shared nothing data is stored.
- `/opt/tibco/be/BE_SHORT_VERSION/rms/config/security` - Directory which holds the rule management server application's ACL (permission configuration) and `user.pwd` files.
- `/opt/tibco/be/BE_SHORT_VERSION/examples/standard/WebStudio` - The repository directory for BusinessEvents Webstudio which where all projects are stored.
- `/opt/tibco/be/BE_SHORT_VERSION/rms/config/notify` - Directory where email notification configuration files are stored.
- `/opt/tibco/be/BE_SHORT_VERSION/rms/lib/locales` - Directory where the user locale configuration is stored.
- `/opt/tibco/be/BE_SHORT_VERSION/rms/bin/logs` - Directory where rule management server and Webstudio logs are stored.

`BE_SHORT_VERSION`: TIBCO BusinessEvents software version in the short form. For example, for TIBCO BusinessEvents version 5.5.0 the `BE_SHORT_VERSION` is `5.5`.

```
#BusinessEvents Volumes
VOLUME /mnt/tibco/be/logs
VOLUME /mnt/tibco/be/data-store

#RMS Volumes
VOLUME /opt/tibco/be/${BE_SHORT_VERSION}/rms/config/security
VOLUME /opt/tibco/be/${BE_SHORT_VERSION}/examples/standard/WebStudio
VOLUME /opt/tibco/be/${BE_SHORT_VERSION}/rms/config/notify
VOLUME /opt/tibco/be/${BE_SHORT_VERSION}/rms/lib/locales
VOLUME /opt/tibco/be/${BE_SHORT_VERSION}/rms/shared
```

Ports (EXPOSE Command)

The `EXPOSE` command is used to associate a specified port to enable networking between the running process inside the container and the outside world (that is, the host). The usage of the `EXPOSE` command is:

```
EXPOSE port1 port2 ...
```

By default the following ports are exposed by the base BusinessEvents image:

- 50000: This is the ActiveSpaces listen port exposed by the base image.
- 5555: This is the JMX port exposed by the base image.
- 8090 and 5000: These are the rule management server port exposed by the base image.

```
# This will always be the listen port for AS
EXPOSE 50000

# JMX Port
EXPOSE 5555

#RMS PORTs
EXPOSE 8090 5000
```

These ports can be mapped during Docker run.

Building a Docker Image for TIBCO BusinessEvents®

TIBCO BusinessEvents provides a script file to build Docker image for BusinessEvents. This batch file (for Windows and Linux) is located at the `BE_HOME/docker/bin` folder.

In Linux platform, you can also generate the Docker image for BusinessEvents of reduced size using an alternate script. For details, see [\(Linux Only\) Building a Lightweight Docker Image for TIBCO BusinessEvents®](#) on page 187.

Prerequisites

Before building the Docker image for BusinessEvents, ensure that Dockerfile of BusinessEvents is created. See [Dockerfile for TIBCO BusinessEvents](#) for more details.

Procedure

- Go to the `BE_HOME/docker/bin` folder and run the Docker image building script `build_be_image.sh` (in Linux) or `build_be_image.bat` (in Windows).

```
./build_be_image.sh -l <INSTALLERS_LOCATION> -v <BE_VERSION> -e <BE_EDITION> -i <IMAGE_VERSION> -a <BE_ADDONS> --hf=<BE_HOTFIX> --as-hf=<AS_HOTFIX> -d <DOCKERFILE>
```

Docker Image Building Script Arguments

Argument	Required/Optional	Description
<code>-l/--installers-location</code>	Required	Location where TIBCO BusinessEvents and TIBCO Activespaces installers are located.
<code>-e/--edition</code>	Required	TIBCO BusinessEvents software edition. The values are: <ul style="list-style-type: none"> • standard • enterprise
<code>-v/--version</code>	Required	TIBCO BusinessEvents software version (3 part number).
<code>-i/--image-version</code>	Required	Version number that you want to assign to the image (for example, v01)

Argument	Required/Optional	Description
-a/--addons	Optional	(Optional) Comma-separated values for required addons. The values are: <ul style="list-style-type: none"> • process • views • datamodeling • decisionmanager • eventstreamprocessing
--hf	Optional	Additional TIBCO BusinessEvents hotfix version (for example, 1)
--as-hf	Optional	Additional TIBCO ActiveSpaces hotfix version (for example, 1)
-d/--dockerfile	Optional	Dockerfile to be used for generating image. You can use your own Dockerfile or you can edit and use the Dockerfiles provided with the BusinessEvents installation. The default value is Dockerfile.

For example:

```
./build_product_image.sh -l /home/pkgs/5.5 -e standard -v 5.5.0 -i v01 -a process --hf=1
```



- Run the Docker image building script from *BE_HOME/docker/bin* folder as the folder contains other files that are required for building the docker image.
- Encapsulate all the arguments between double quotes (").

(Linux Only) Building a Lightweight Docker Image for TIBCO BusinessEvents®

TIBCO BusinessEvents also provides a script file to build a lightweight Docker image for TIBCO BusinessEvents. This script file is available only for Linux platform at the *BE_HOME/docker/frominstall* folder.

Prerequisites

Before building the Docker image for TIBCO BusinessEvents, ensure that Dockerfile of TIBCO BusinessEvents is created. See [Dockerfile for TIBCO BusinessEvents](#) for more details.

Procedure

- Navigate to the *BE_HOME/docker/frominstall* folder and run the Docker image building script *build_be_image_frominstallation.sh*.

```
./build_be_image_frominstallation.sh [-h <BE_HOME_LOCATION>] -v <BE_VERSION> -i <IMAGE_VERSION> -d <DOCKERFILE> [-o <true/false>]
```

Docker Image Building Script Arguments

Argument	Required/Optional	Description
<code>-v/--version</code>	Required	TIBCO BusinessEvents software version (3 part number).
<code>-i/--image-version</code>	Required	Version number that you want to assign to the image (for example, v01)
<code>-h/--BE_HOME</code>	Optional	Specify <i>BE_HOME</i> location. This is optional if the script is run from its default location.
<code>-o/--overwrite</code>	Optional	Specifies whether to overwrite <code>be.tar</code> file (if already present) at <i>BE_HOME/docker/bin</i> . The values are: <ul style="list-style-type: none"> <code>true</code> - overwrite the <code>be.tar</code> and continue <code>false</code> - abort the script and exit. The default value is <code>false</code> .
<code>-d/--dockerfile</code>	Optional	Dockerfile to be used for generating image. You can use your own Dockerfile or you can edit and use the Dockerfiles provided with the TIBCO BusinessEvents installation. The default value is <code>Dockerfile.fromtar</code> .

For example:

```
./build_product_image.sh -v 5.5.0 -i v01 -o true
```



Run the Docker image building script from *BE_HOME/docker* folder as the folder contains other files too, which are required for building the Docker image.

Generating BusinessEvents Application Dockerfile

For building the BusinessEvents application Docker image, you require the BusinessEvents application Dockerfile. TIBCO BusinessEvents provide an utility to generate the application Dockerfile automatically based on the supplied arguments.

If you want to include external files, external JAR file, global variables, backing store shared resource, or expose more ports for the application, see [BusinessEvents Docker Utility Features](#) on page 189.

Prerequisites

Before generating the BusinessEvents application Dockerfile, ensure that BusinessEvents Docker image is created. See [Building the TIBCO BusinessEvents Docker Image](#) for more details.

Procedure

- Go to the *BE_HOME/docker/bin* folder and run the application Dockerfile generation executable `be-docker-gen`.

```
./be-docker-gen -t <TARGET_DIRECTORY> -i <BE_BASE_DOCKER_IMAGE> -m <MAINTAINER> -e <EMAIL> -l <LABEL> -o <OVERWRITE_DOCKERFILE> -h <HELP>
```

Docker Image Building Script Arguments

Argument	Required/Optional	Description
-t/-target-dir	Required	Path of the directory where the application CDD file, enterprise archive (EAR) file, and external jars are stored.
-i/-image	Required	Name of the base BusinessEvents Docker image. The naming convention of the Docker image consists of BusinessEvents version and image version number: <i>BusinessEventsVersion-ImageVersion</i> . For example, 5.5.0-v01.
-m/-maintainer	Required	Name of the author of the Docker image.
-e/-email	Required	Email of the author of the Docker image.
-l/-label	Optional	Label that you want attach to this application Dockerfile. You can specify multiple labels.
-o	Optional	Specifies whether to overwrite the existing Dockerfile (if already present) at the target directory. The values for the -o parameter are: <ul style="list-style-type: none"> • true • false The default value of the -o parameter is false.
-h/-help	Optional	Displays usage of the be-docker-gen command.

For example:

```
./be-docker-gen -t /home/app -i 5.5.0-v01 -m FraudDetection -e fdcache@abc.com -l ApplicationName=FraudDetection -o true
```

BusinessEvents Docker Utility Features

The BusinessEvents Docker utilities provide various features that you can use to successfully run BusinessEvents application in Docker.

Custom Docker Files

To include additional environment variables in the generated Dockerfile, place the custom file with the extension `.custom` in the target *directory*. The Dockerfile generation utility merge these custom files with the generated Dockerfile.

External JAR Files

If an application is dependent on some external/third party JAR files, copy the required JAR files to the *target directory*. The BusinessEvents application Docker image generation utility copies these JAR files to the BusinessEvents Docker image and add them to the engine classpath.

Global Variables

The Dockerfile generation utility queries the enterprise archive (EAR) and CDD file, and exposes the global variables as environment variables in the Dockerfile.

If a global variable is overridden in the CDD file, the CDD global variable takes precedence and the Dockerfile is updated with the CDD value.

Docker Ports

If your application requires to expose its network ports, use BusinessEvents global variable for such ports and ensure that name of the global variable has a suffix `_PORT` in it. The Dockerfile generation utility exposes all such global variables using Docker `EXPOSE` command in the Dockerfile.

Docker Volumes

If an application needs to access an external file system or needs to persist file system data across Docker runs, then expose the required file system paths as Volumes in the Dockerfile. In those file system paths, the name of the global variable should contain a suffix `_PATH`. All those global variables with suffix `_PATH` are mapped to Docker Volume.

Backing Store Shared Resources

If an application has a backing store with the Shared All mode then the associated shared resource properties are exposed in the Dockerfile as environment variables. These variables are then updated in the shared resource of the backing store as global variables.

The following properties are exposed as global variables:

- `BACKINGSTORE_JDBC_DRIVER`
- `BACKINGSTORE_JDBC_URL`
- `BACKINGSTORE_JDBC_USERNAME`
- `BACKINGSTORE_JDBC_PASSWORD`
- `BACKINGSTORE_JDBC_POOL_SIZE`
- `BACKINGSTORE_JDBC_LOGIN_TIMEOUT`
- `BACKINGSTORE_JDBC_USE_SSL`



Before generating the dockerfile for your application, ensure that the required driver libraries are present in the target directory.

External Files

If your application requires an external file for the Docker, run those files should be included in the Docker image. To include these files, declare a global variable in the application project with the suffix `_FILE` and the path of the file as its value. Also, place those files in the target directory under the folder `files`. The utility creates directories for the file path and add the files to that location. The file location is also added as a Volume in the Dockerfile, so that you can map that location and update files at run time.

Building BusinessEvents Application Docker Image

TIBCO BusinessEvents provides a script file to build Docker image for BusinessEvents application using the Dockerfile generated for the application. This batch file (for Windows and Linux) is located at the `BE_HOME/docker/bin` folder.

To include additional BusinessEvents engine properties in the application Docker image, add a properties file with all the additional properties in it and extension `.props` in the *target directory*.

Prerequisites

Before building the BusinessEvents application Docker image, ensure that Dockerfile of BusinessEvents application is created. See [Generating BusinessEvents Application Dockerfile](#) for more details.

Procedure

- Go to the `BE_HOME/docker/bin` folder and run the application Docker image building script `build_app_image.sh` (in Linux) or `build_app_image.bat` (in Windows).

```
./build_app_image.sh <APP_IMAGE_NAME>:<APP_IMAGE_VERSION> <TARGET_DIRECTORY>
```

Where:

- `APP_IMAGE_NAME` - Name of the application image.
- `APP_IMAGE_VERSION` - Version of the application image.
- `TARGET_DIRECTORY` - The directory where the generated application Dockerfile, application CDD file, and application enterprise archive (EAR file) are present.

For example:

```
./build_app_image fdcache:v01 /home/temp
```

Running TIBCO BusinessEvents® Application in Docker

By using the BusinessEvents application Docker image, you can run the BusinessEvents application in Docker.

Prerequisites

- Install Docker on the machine and perform the initial setup based on your operating system. Refer to the *Docker documentation* at <https://docs.docker.com> for complete details on Docker installation.
- Ensure that a network bridge exists for internal communication between Docker images. You can use the `network create` command of Docker to create the network bridge:


```
docker network create <BRIDGE_NAME>
```
- Build the BusinessEvents application Docker image, see [Building BusinessEvents Application Docker Image](#).

Procedure

- Execute the run command and provide value for the following options to run the BusinessEvents application.

```
docker run --net=<BRIDGE_NETWORK> --name=<CONTAINER_NAME> -e PU=<PU_NAME> -e <ADDITIONAL_ENV_VARS> <APPLICATION_IMAGE_NAME>
```

Where:

- `--net=<BRIDGE_NETWORK>` - Specify the name of the network bridge that you have created. This connects the container to the specified network.
- `--name=<CONTAINER_NAME>` - Specify a name that you want to assign to the container.
- `<APPLICATION_IMAGE_NAME>` - Specify the name of the BusinessEvents application Docker image.
- `-e <PU_NAME> <ADDITIONAL_ENV_VARS>` - Use the `-e` option to set environment variables, as required, with syntax `VAR=Value`. You can use the following environmental variables that at the run time.
 - **AS_DISCOVER_URL**: Specify the discover URL, which enables members to discover each other in the datagrid. For example:

```
docker run --net=simple-bridge --name=inference -e AS_DISCOVER_URL=tcp://cache:50000 -e PU=default -p 8109:8109 fdcache:v01
```

Here the docker name of the cache server "cache" is used for the `AS_DISCOVER_URL` of the inference agent. Since all agents running on the same Docker host can resolve Docker names

to their IP addresses on the network, thus you can create clusters across instances on the same Docker.

- **COMPONENT:** This environment variable is required for rule management server. Specify this environment variable value as `rms` to run the rule management server. See [Running BusinessEvents Rule Management Server \(RMS\) in Docker](#) on page 193 for more details.
- **PU:** Specify the processing unit that need to be started. For example, running the application with "cache" as processing unit:

```
docker run --net=be_network --name=cache -e PU=cache fdcache:v01
```

- **LOG_LEVEL:** Specify the override value for the predefined log level. You can specify the comma separated values for the log patterns required. If the `LOG_LEVEL` environment variable is not specified, the `log-config` of the CDD file is used. The pattern configurations are same as the the `log-config` of the CDD file. For example:

```
docker run --net=simple-bridge --name=cache -e PU=cache -e LOG_LEVEL=:debug fdcache:v01
```

- **DOCKER_HOST:** Specify the host, where the `docker run` command is executed. This environment variable is required for remote JMX connections to the running container. For example:

```
docker run --net=be_network --name=sample -p 5555:5555 -e PU=default -e DOCKER_HOST=10.97.123.56 sample:v01
```



The default JMX port for engines running inside docker is 5555. This port should be mapped with a local port, defined in the base Dockerfile, at run time using `-p` option.

- **AS_PROXY_NODE:** Specifies whether the container run as a proxy node. Set the value to `true`, to start the node in proxy mode. For example:

```
docker run ... -e AS_PROXY_NODE=true ...
```

The port 50001 is the default ActiveSpaces remote listen port which can be specified while connecting to the proxy node. For example:

```
docker run ... -e AS_DISCOVER_URL=tcp://<container_name>:50001?remote=true ...
```

- **TRA properties:** You can specify any of the BusinessEvents engine and JVM properties as an environment variable.

To use the property, append `tra.` at the beginning of the property name. For example, to use `java.extended.properties`, provide `tra.java.extended.properties` and its value as environment variable. The value of the environment variable `tra.java.extended.properties` overwrites the value of the `java.extended.properties` property in the `be-engine.tra` file.

You can also specify few JVM properties, such as, `-Xms`, `-Xmx`, and `-Xss` as environment variable individually. These individual JVM properties, when specified as environment variable, take precedence over the JVM properties defined in the `tra.java.extended.properties` environment variable. Other JVM properties, such as, garbage collection properties still have to be defined under the `tra.java.extended.properties` environment variable. The following table lists the environment variables that you can use for these JVM property options.

Environment Variables for JVM Properties

Task	JVM Property Option	Environment Variable
Set initial Java heap size	<code>-Xms</code>	<code>tra.java.heap.size.initial</code>

Task	JVM Property Option	Environment Variable
Set maximum Java heap size	-Xmx	tra.java.heap.size.max
Set Java thread stack size	-Xss	tra.java.stack.size

For example:

```
docker run -e "tra.java.heap.size.initial=1024m" -e
"tra.java.heap.size.max=1024m" -e "tra.java.stack.size=2m" -
e="tra.java.extended.properties=-server -Xms512m -Xmx512m -javaagent:
%BE_HOME%/lib/cep-base.jar -XX:MaxMetaspaceSize=256m -XX:+UseParNewGC -XX:
+UseConcMarkSweepGC" com.tibco.be.fd:v016
```

In the previous example, `tra.java.heap.size.initial=1024m` and `tra.java.heap.size.max=1024m` takes precedence over the `-Xms512m` and `-Xmx512m` options of `tra.java.extended.properties`. Thus, the initial Java heap size and maximum Java heap size is set to 1024M instead of 512M. Also, the `tra.java.stack.size=2m` environment variable sets the `-Xss` option of `java.extended.properties` property in the `be-engine.tra` file to 2M.

- **Global Variable:** You can specify a global variable as an environment variable to override its value. Provide the global variable name and its value as an environment variable. For example, to specify value for the global variable `HOSTNAME` as `localhost`, run the following command:

```
docker run ... -e "HOSTNAME=localhost" ...
```

Running BusinessEvents Rule Management Server (RMS) in Docker

Rule management server is an integral part of BusinessEvents for using WebStudio and Decision Manager.

You can configure the rule management server volumes in the BusinessEvent base Docker image. See [Dockerfile for TIBCO BusinessEvents](#) on page 184 for more details on the rule management server volumes.

Procedure

1. Generate the TIBCO BusinessEvents base Docker image.
See [Building a Docker Image for TIBCO BusinessEvents®](#) on page 186 for more details.
2. Generate the rule management server Dockerfile.
The process is same as you generate the Dockerfile for any BusinessEvents application with the only addition that you need to put `RMS.ear` and `RMS.cdd` in the target directory. See [Generating BusinessEvents Application Dockerfile](#) on page 188 for more details.
3. Build the rule management server Docker image.
The process is same as you build the Docker image for any BusinessEvents application. See [Building BusinessEvents Application Docker Image](#) on page 190 for more details.
4. Start the rule management server container using the following command:

```
docker run --net=<BRIDGE_NETWORK> --name=<RMS_CONTAINER_NAME> -e COMPONENT="rms"
-e PU=default <RMS_IMAGE_NAME>
```

where:

- `--net=<BRIDGE_NETWORK>` - Specify the name of the network bridge that you have created. This connects the container to the specified network.

- `--name=<RMS_CONTAINER_NAME>` - Specify a name that you want to assign to the rule management server container.
- `-e PU=<PU_NAME>` - Specify the processing unit that you want to start with the command.
- `-e COMPONENT="rms"` - Set the environment variable `COMPONENT` value as "rms" for the rule management server.
- `<RMS_IMAGE_NAME>` - Specify the name of the BusinessEvents rule management server Docker image.

Setting up BusinessEvents Multihost Clustering on Amazon EC2 Instances Using Docker

You can setup BusinessEvents multihost clustering on Amazon Elastic Compute Cloud (Amazon EC2) instances using Docker and Weave Net.

Prerequisites

- An Amazon Web Services (AWS) account. Refer to the Amazon EC2 documentation at <https://aws.amazon.com/documentation/ec2/> to learn Amazon EC2 concepts and how to use the Amazon EC2 console.
- TIBCO BusinessEvents application image. See [Dockerize TIBCO BusinessEvents](#) on page 183 for more details on running TIBCO BusinessEvents on Docker.
- (Optional) Docker Hub registry account or any other Docker registry account. Refer to the <https://docs.docker.com/> to learn more about Docker.
- Weave Net for multi host docker networking. Refer to the Weave Net documentation at <https://www.weave.works/docs/net/latest/features/> to learn on how to use Weave Net and how to integrate with Docker.
- Amazon Elastic File System configuration (EFS) for shared nothing persistence. Refer to the Amazon EFS documentation at <https://aws.amazon.com/documentation/efs/> to learn about Amazon EFS concepts and configurations.
- Relational Database Service configuration (RDS) for shared all persistence. Refer to the Amazon RDS documentation at <https://aws.amazon.com/documentation/rds/> to learn about Amazon RDS concepts and configurations.

Setting up Standalone Amazon EC2 Instances

For BusinessEvents multihost clustering, you must create Amazon Elastic Cloud Compute (Amazon EC2) instances and configure Docker and Weave Net on each of them. This setup is common for shared all and shared nothing persistence options.

Procedure

1. Login to Amazon EC2 console with your credentials.
Refer to Amazon EC2 documentation at <https://aws.amazon.com/documentation/ec2/> for more details on setting Amazon EC2 account.
2. In the Amazon EC2 console, create a new security group with the following inbound rules.

Inbound Rules

Rule No.	Type	Protocol	Port	Source
1	SSH	TCP	22	Anywhere
2	Custom TCP Rule	TCP	6783	Anywhere
3	Custom UDP Rule	UDP	6783	Anywhere
4	Custom TCP Rule	TCP	<HTTP Port as per BusinessEvents project>	Anywhere



Port TCP/UDP 6783 is required for weave networking. You can configure source according to your requirement.

Refer to Amazon EC2 documentation at <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html> for details on how to create security group.

- On the Amazon EC2 console, create two or more Standalone Amazon EC2 instances of type Ubuntu or CentOS or as per your requirement. Specify the configuration parameters according to your requirement in the wizard.
 - Select the default Virtual Private Cloud (VPC) for testing purpose or you can use an customized one.
 - Select the security group created earlier in [Step 2](#).
 - Generate a new key pair (.pem) per instance and save it.

Refer to the Amazon EC2 documentation at <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/LaunchingAndUsingInstances.html> for more details on launching an instance.

- In the Amazon EC2 console, on **Review Instance Launch** page, check the details of your instance, and after the verification click **Launch**.
- Ensure that all instances are in the "running" state and status check are marked with no error.
- Note down the public and private IP address/DNS of all instances, which can be later used for connection.
- Change the permission of PEM key.


```
> chmod 400 mykey.pem
```
- Securely login to Amazon EC2 instances using an SSH client.


```
> ssh -i /path/to/mykey.pem ec2-user@<public IP address of EC2 instance or public DNS>
```



User name could be ec2-user or ubuntu as per the Amazon EC2 instance type.

- Install Docker on all Amazon EC2 instances.

Refer to the installation instructions mentioned in the Docker documentation at <https://docs.docker.com/engine/installation/>.

- Install Weave Net all EC2 instances.

```
> sudo curl -L git.io/weave -o /usr/local/bin/weave
> sudo chmod a+x /usr/local/bin/weave
```

Refer to the installation instructions in the Weave Net documentation at <https://www.weave.works/docs/net/latest/installing-weave/>.

- Start weave on each instance, and provide it other peers private IP addresses.

On Instance 1,

```
> weave launch
```

On Instance 2,

```
> weave launch <HostName/Private IP address of Instance 1>
```

- Run the following command and check status of the peers connection.

```
> weave status
```

If the connection is successful, the status displays the number of established connections. For example,

```
Peers: 2 (with 2 established connections)
```

Configuring Amazon RDS for Shared All Persistence



In this approach BusinessEvents application image is built locally and the docker registry is used to push or pull images. You can also build images directly on Amazon EC2 instances. If required, you can also configure separate VPC and security group.

Prerequisites

Check Amazon Relational Database Service (RDS) prerequisites at http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_SettingUp.html.

Procedure

- Create a Amazon RDS of type Oracle.
Refer to the Amazon RDS documentation at http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_GettingStarted.CreatingConnecting.Oracle.html for more details on how to do it.
- Use default VPC, same used for Amazon EC2 instances. Also, in the same security group add one more inbound rule for database port.

Inbound Rules

Type	Protocol	Port	Source
Custom TCP Rule	TCP	1501	Anywhere

Or if required, you can create a separate security group for the database instance

- After the database instance is running and is in "available" state, you can establish a connection to it using any SQL client.
- Create a BusinessEvents specific user and run all BusinessEvents specific scripts that are required.
- After the database setup is ready, use the same database setup in the JDBC shared resource. You can use the database instance endpoint as **Database URL** in the JDBC shared resource. Use the **Test Connection** feature to check if the connection is successful.
- Create BusinessEvents application docker image locally on any machine and push it to docker registry.
See [Running TIBCO BusinessEvents in Docker](#) on page 183 for more details on how to do it.
- Pull this BusinessEvents application docker image on all Amazon EC2 instances.
After the BusinessEvents application image is available on all EC2 instances, you can run BusinessEvents application containers.
- Set the Weave environment on all Amazon EC2 instances for running BusinessEvents application containers.

```
> eval $(weave env)
```

9. Start containers on all Amazon EC2 instances.

For example,

```
//Start cache 1 on instance 1
docker run -d --name=cache1SA -e PU=cache <username>/fdstore_sharedall:GA
//Start cache 2 on instance 2
docker run -d --name=cache2SA -e PU=cache -e AS_DISCOVER_URL=tcp://cache1SA:
50000 <username>/fdstore_sharedall:GA
//Start inference on instance 2
docker run -d --name=InfSA -p 8209:8209 -e PU=default -e AS_DISCOVER_URL=tcp://
cache1SA:50000 <username>/fdstore_sharedall:GA
```

Ensure that all BusinessEvents application containers are connected to each other and inference is processing events at port 8209.

10. For sending events using `readme.html` of the example application, replace `localhost` with the public IP address of instance where the inference container is running.
As long as the RDS database instance is in running state, data is persisted.
11. To check the data recovery, stop all Amazon EC2 instances and start them again.
12. Restart all stopped containers and check that the data is recovered in cache containers.

Configuring Amazon EFS for Shared Nothing Persistence

Procedure

1. Create a Amazon Elastic File System (EFS) with the same Virtual Private Cloud (VPC) and security group as of the Amazon EC2 instances.
Refer to the Amazon EFS documentation at <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEFS.html> for detailed steps on how to create an Amazon EFS file system.
2. Note down DNS name which is required while mounting EFS on Amazon EC2 instances.
3. Open an SSH client and connect to your Amazon EC2 instance.
4. Install the NFS client on all Amazon EC2 instances.
On an Amazon Linux, Red Hat Enterprise Linux, or SuSE Linux instance, run the following command:

```
> sudo yum install -y nfs-utils
```

On an Ubuntu instance, run the following command:

```
> sudo apt-get install nfs-common
```
5. Create a new directory on all Amazon EC2 instances, such as "efs".

```
> sudo mkdir efs
```
6. Mount your file system by using the EFS DNS name.

```
> sudo mount -t nfs4 -o
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2 fs-
cb8b5e62.efs.us-west-2.amazonaws.com:/ efs
```

If the connection was not successful, refer to Amazon EFS troubleshooting documentation at <http://docs.aws.amazon.com/efs/latest/ug/troubleshooting.html>.
7. Run the following command to see the mount:

```
> df -T
```
8. Update BusinessEvents application CDD with shared nothing datastore path as `/mnt/tibco/be/data-store`, which is declared as `VOLUME` in BusinessEvents base dockerfile.
9. Create BusinessEvents application docker image locally on any machine and push it to docker registry.
See [Running TIBCO BusinessEvents in Docker](#) on page 183 for more details on how to do it.

10. Pull this BusinessEvents application docker image on all Amazon EC2 instances.
Once the BusinessEvents application image is available on all EC2 instances, you can run BusinessEvents application containers.
11. Set the Weave environment on all Amazon EC2 instances for running BusinessEvents application containers.

```
> eval $(weave env)
```
12. Start containers on all Amazon EC2 instances.
For example,

```
//Start cache 1 on instance 1
docker run -d --name=cache1SN -v /home/ubuntu/efs:/mnt/tibco/be/data-store -e
PU=cache <username>/fdstore_sharednothing:GA
//Start cache 2 on instance 2
docker run -d --name=cache2SN -v /home/ubuntu/efs:/mnt/tibco/be/data-store -e
PU=cache -e AS_DISCOVER_URL=tcp://cache1SN:50000 <username>/
fdstore_sharednothing:GA
//Start inference on instance 2
docker run -d --name=InfSN -v /home/ubuntu/efs:/mnt/tibco/be/data-store -p
8209:8209 -e PU=default -e AS_DISCOVER_URL=tcp://cache1SN:50000 <username>/
fdstore_sharednothing:GA
```

Ensure that all BusinessEvents application containers are connected to each other and inference is processing events at port 8209.
13. For sending events using `readme.html` of the example application, replace `localhost` with the public IP address of instance where the inference container is running.
As long as EFS is in running state, data is persisted.
14. To check the data recovery, stop all Amazon EC2 instances and start them again. Mount the EFS target again as mentioned in [Step 6](#).
15. Restart all stopped containers and check that the data is recovered in cache containers.

Running TIBCO BusinessEvents® on AWS Based Kubernetes Cluster

Kubernetes is an open-source platform designed to automate deploying, scaling, and operating application containers. Kubernetes can run application containers on clusters of physical or virtual machines.

For more information about Kubernetes, see the Kubernetes documentation (<https://kubernetes.io/docs/home/>).

In BusinessEvents, to form a cluster there are discovery nodes which define or start a cluster and other cache nodes and inference nodes (non-discovery nodes), which connect to one or more discovery nodes and become a member of the cluster. In Kubernetes, each BusinessEvents node runs as a Kubernetes *pod*. Pods can communicate with each other using their IP addresses. However, due to the dynamic nature of the IP addresses, non-discovery nodes cannot always connect to the discovery nodes. Thus, to resolve this, discovery nodes are modeled as Kubernetes *services*. The service is reachable by its name using the Kubernetes DNS. The non-discovery nodes use indirection using the Kubernetes service to connect to the discovery nodes.

Prerequisites

Download and install the following CLIs on your system:

CLI	Download and Installation Instruction Link
kops	https://github.com/kubernetes/kops/blob/master/docs/aws.md
kubect1	https://kubernetes.io/docs/tasks/tools/install-kubect1/

CLI	Download and Installation Instruction Link
aws	https://aws.amazon.com/cli/

Procedure

1. Set up a Kubernetes cluster on Amazon Web Services (AWS). For more information, see [Setting up a Kubernetes Cluster on AWS](#) on page 199.
2. Create Docker image of TIBCO BusinessEvents application. For more information, see [Building BusinessEvents Application Docker Image](#) on page 190.
3. Go to the EC2 Container Services dashboard and create a repository with the same name as the Docker image of TIBCO BusinessEvents application. Upload the BusinessEvents application image to the repository and for help you might use the **View Push Commands** button.



AWS Repository name must be same as the Docker image of TIBCO BusinessEvents application.

For more information on how to create a repository in Amazon AWS, refer to <https://docs.aws.amazon.com/AmazonECR/latest/userguide/repository-create.html>.

4. Based on your application architecture you deploy BusinessEvents cluster on Kubernetes:
 - For deployment of BusinessEvents Cluster for No Backing Store cluster, see [Deploying TIBCO BusinessEvents® Cluster for No Backing Store on AWS](#) on page 200
 - For deployment of BusinessEvents Cluster for Shared Nothing storage, see [Deploying TIBCO BusinessEvents® Cluster for Shared Nothing Storage on AWS](#) on page 201.
 - For deployment of BusinessEvents Cluster for Shared All storage, see [Deploying BusinessEvents Cluster for Shared All Storage on AWS](#) on page 204.

Setting up a Kubernetes Cluster on AWS

Set up a Kubernetes cluster with AWS for running TIBCO BusinessEvents® application.

Procedure

Creating Cluster

1. Create an S3 storage to store the cluster configuration and state. You can use either AWS CLI or AWS console to create the storage.

The sample AWS CLI command for creating S3 storage is:

```
aws s3 mb s3://be-bucket
```

For more information on Amazon Simple Storage Service (Amazon S3), see the Amazon S3 Documentation at <https://aws.amazon.com/documentation/s3/>.

2. Create the Kubernetes cluster on AWS using the following command:

```
kops create cluster --zones us-west-2a --master-zones us-west-2a --master-size t2.large --node-size t2.large --name becluster.k8s.local --state s3://<s3-bucket-name> --yes
```

Where,

- `s3-bucket-name` is the name of the S3 storage created earlier.
- `becluster.k8s.local` is the name of the cluster being created. Use `k8s.local` prefix to identify a gossip-based Kubernetes cluster and you can skip the DNS configuration.

For more information on the `kops create cluster` command either use the `help` parameter or refer to the `kops` tool documentation at <https://github.com/kubernetes/kops/tree/master/docs>.

Validating Cluster

3. Validate your cluster using the `validate` command.

```
kops validate cluster
```

Node and master must be in ready state. The `kops` utility stores the connection information at `~/.kops/config`, and `kubectl` uses the connection information to connect to the cluster.

Deleting the Cluster

4. If needed, you can delete the cluster using the following command:

```
kops delete cluster becluster.k8s.local --state=s3://<s3-bucket-name> --yes
```

Deploying TIBCO BusinessEvents® Cluster for No Backing Store on AWS

BusinessEvents cluster can be deployed on AWS using the configuration files (YAML format), which contain the configuration details for deployment including environment variables.

Prerequisites

For deploying BusinessEvents cluster for No Backing Store on AWS, you must first set up Kubernetes cluster on AWS and then upload your Docker image on AWS. For more information, see [Running TIBCO BusinessEvents® on AWS Based Kubernetes Cluster](#) on page 198.

Procedure

1. Create Kubernetes resources, required for deploying BusinessEvents cluster, using the YAML files. These resources include deployment and services for the cluster. Thus, to deploy a BusinessEvents cluster, create:

- a discovery node (pod) to start the cluster
- a service to connect to discovery node
- a cache agent node which connects to the discovery node service
- an inference agent node which connects to the discovery node service
- a service to connect to the inference agent.

You can find the following sample YAML files at `BE_HOME\cloud\kubernetes\`.

Sample Kubernetes Resource YAML Files for No Backing Store

File Name	Resource	Resource Type	Description
<code>bediscoverynode.yaml</code>	Discovery node	Deployment	Set up the container (containers) with the docker image (image) of the application. Provide a label (labels) to the deployment which the discovery node service can use as selector. Specify only one replica (replicas) of the discovery node.
<code>bediscovery-service.yaml</code>	Discovery node service	Service (Internal)	Set up the service to connect to the discovery node. Specify the label of the discovery node as the value of selector. Other nodes in the cluster use this service to connect to the discovery node. Specify the protocol and port to connect to this service.

File Name	Resource	Resource Type	Description
becacheagent.yaml	Cache agent node	Deployment	Set up the container (containers) with the docker image (image) of the application. Specify replicas value and start as many cache agent as specified in the value. Connect to the discovery node service using the discovery protocol and port specified in the discovery node service.
beinferenceagent.yaml	Inference agent node	Deployment	Set up the container (containers) with the docker image (image) of the application. Provide a label (labels) to the deployment which the inference agent service can use as selector. Specify at least one replica (replicas) of the inference agent node. Connect to the discovery node service using the discovery protocol and port specified in the discovery node service.
befdservice.yaml	Inference agent service	Service (LoadBalancer/External)	Set up the service to externally connect to the inference agent. Setup the label of the inference agent as the value of the selector variable for connection. Specify the protocol and port to connect to this service externally.

2. Run the create command of kubectl utility by using the YAML files to deploy the BusinessEvents cluster.

For example, deploy the cluster by using the following sample files:

```
kubectl create -f bediscovery-service.yaml
kubectl create -f bediscoverynode.yaml
kubectl create -f becacheagent.yaml
kubectl create -f beinferenceagent.yaml
kubectl create -f befdservice.yaml
```

You can also get the external IP to the external service of the cluster by using the `get services` command. You can then use that IP to connect to the cluster.

```
kubectl get services befdservice
```

You can check the logs of individual BusinessEvents container pods using the following command:

```
kubectl logs <pod>
```

Deploying TIBCO BusinessEvents® Cluster for Shared Nothing Storage on AWS

By using the Kubernetes elements such as the StatefulSets object and dynamic volume provisioning features, you can create TIBCO ActiveSpaces and Shared Nothing deployments.

StatefulSets gives deterministic names to the pods. Along with dynamic volume provisioning, StatefulSets also give deterministic names to PersistentVolumeClaims (PVC). This ensures that when a particular member of a StatefulSet goes down and comes up again, it attaches itself to the same PVC. For more information about the Kubernetes concepts of StatefulSets, dynamic volume provisioning, and PersistentVolumeClaims, refer to the Kubernetes documentation at <https://kubernetes.io/docs/concepts/>.

Prerequisites

Ensure that CDD of your application is configured to use Shared Nothing. For deploying BusinessEvents cluster for Shared Nothing storage on AWS, you must first set up Kubernetes cluster on AWS and then upload your docker image to an AWS docker registry. For more information, see [Running TIBCO BusinessEvents® on AWS Based Kubernetes Cluster](#) on page 198.

Procedure

1. In AWS, create an EFS file system.

For more information on the steps to create an EFS file system, see Amazon EFS documentation at <https://docs.aws.amazon.com/efs/latest/ug/gs-step-two-create-efs-resources.html>. Specify the Kubernetes cluster Virtual Private Cloud (VPC) and **Security Group** while creating a mount target for the file system. On the File Systems page, verify that the mount target shows the **Life Cycle State** as Available. Under **File system access**, you see the file system **DNS name**. Make a note of this DNS name.

After successful creation of the EFS file system, note its **File System ID**, which can be used for creating EFS provisioner.

2. Create the EFS provisioner and other associated resources. Specify all the connection setup values for the EFS file system in a `manifest.yaml` file and run the `kubectl` command to create the EFS provisioner.
 - a) Download the sample `manifest.yaml` file from <https://raw.githubusercontent.com/kubernetes-incubator/external-storage/master/aws/efs/deploy/manifest.yaml> and edit it according to your setup.
 - b) In the `configmap` section specify **File System ID** of the newly created EFS as the value of the `file.system.id: variable` and **Availability Zone** of the newly created EFS as the value of the `aws.region: variables`.
 - c) In the `Deployment` section, specify DNS name of the newly created EFS as the value of the `server: variable`.
 - d) Run the `kubectl` command to apply the settings in `manifest.yaml`.


```
kubectl apply -f manifest.yaml
```
 - e) Ensure that the EFS provisioner pod is in the running state using the `kubectl` command.


```
kubectl get pods
```
3. Now, create Kubernetes resources, required for deploying TIBCO BusinessEvents cluster for the shared nothing storage, using the YAML files.

These resources include StatefulSets and services for the cluster. Thus, to deploy a TIBCO BusinessEvents cluster, create:

- A service to connect to discovery node
- A cache agent node which connects to the discovery node service
- An inference agent node that connects to the discovery node service
- An external service to connect to the inference agent

You can find sample YAML files for shared nothing at `BE_HOME\cloud\kubernetes\sn`. The following are the sample files that are provided (configured for FraudDetectionStore example):

Sample Kubernetes Resource YAML Files for Shared Nothing Storage

File Name	Resource	Resource Type	Description
bediscovery-service-sn.yaml	Discovery node service	Service (Internal)	Setup the service to connect to the discovery node. Setup the label of the cache agent as the value of the <code>selector</code> variable for connection. Other nodes in the cluster use this service to connect to the discovery node. Specify the <code>protocol</code> and <code>port</code> to connect to this service.
becacheagent-sn.yaml	Cache agent node	StatefulSet	Setup the container (<code>containers</code>) with the docker image (<code>image</code>) of the application. Specify <code>replicas</code> value and start as many cache agent as specified in the value. Connect to the discovery node service using the discovery protocol and port specified in the discovery node service.
beinferenceagent-sn.yaml	Inference agent node	StatefulSet	Setup the container (<code>containers</code>) with the docker image (<code>image</code>) of the application. Provide a label (<code>labels</code>) to the StatefulSet which the inference agent service can use as <code>selector</code> . Specify at least one replica (<code>replicas</code>) of the inference agent node. Connect to the discovery node service using the discovery protocol and port specified in the discovery node service.
befdservice-sn.yaml	Inference agent service	Service (LoadBalancer/External)	Setup the service to externally connect to the inference agent. Setup the label of the inference agent as the value of the <code>selector</code> variable for connection. Specify the <code>protocol</code> and <code>port</code> to connect to this service externally.

4. Run the `create` command of `kubectl` utility using the YAML files to deploy the BusinessEvents cluster.

For example, deploy the cluster by using the following sample files:

```
kubectl create -f bediscovery-service-sn.yaml
```

```
kubectl create -f becacheagent-sn.yaml
```

```
kubectl create -f beinferenceagent-sn.yaml
```

```
kubectl create -f befdservice-sn.yaml
```

You can also get the external IP to the external service of the cluster by using the `get services` command. You can then use that IP to connect to the cluster.

```
kubectl get services befdservice
```

You can check the logs of individual BusinessEvents container pods using the following command:

```
kubectl logs <pod>
```

Deploying BusinessEvents Cluster for Shared All Storage on AWS

BusinessEvents cluster can be deployed for shared all storage on AWS based Kubernetes cluster using the configuration files (YAML format).

Prerequisites

Ensure that your application connection properties for database use global variables. For deploying BusinessEvents cluster for Shared All storage on AWS, you must first set up Kubernetes cluster on AWS and then upload your Docker image to an AWS Docker registry. For more information, see [Running TIBCO BusinessEvents® on AWS Based Kubernetes Cluster](#) on page 198.

Procedure

1. Create an Amazon RDS based instance and configure it to connect to a BusinessEvents supported database (Oracle, MySQL, DB2, and so on).
For configuration details, refer to the Amazon RDS documentation at <https://aws.amazon.com/documentation/rds/>.
2. Create Kubernetes resources, required for deploying BusinessEvents cluster, using the YAML files. These resources include deployment, services, and ConfigMap for the cluster. Thus, to deploy a BusinessEvents cluster, create:

- A ConfigMap to specify environment variables for database connection
- A discovery node (pod) to start the cluster
- A service to connect to discovery node
- A cache agent node that connects to the discovery node service
- An inference agent node that connects to the discovery node service
- A service to connect to the inference agent

You can find the sample YAML files for shared all storage at `BE_HOME\cloud\kubernetes\sa\`. For details, see [Sample Kubernetes Resource Files for Shared All Storage](#) on page 205.

3. Run the `create` command of `kubectl` utility using the YAML files to deploy the BusinessEvents cluster.

For example, deploy the cluster using the sample files:

```
kubectl create -f db-configmap.yaml
kubectl create -f bediscovery.yaml
kubectl create -f bediscovery-service-sn.yaml
kubectl create -f becache.yaml
kubectl create -f beinference.yaml
kubectl create -f befdservice.yaml
```

You can also get the external IP to the external service of the cluster using the `get services` command. You can then use that IP to connect to the cluster.

```
kubectl get services befdservice
```

You can check the logs of individual BusinessEvents container pods using the following command:

```
kubectl logs <pod>
```

Sample Kubernetes Resource Files for Shared All Storage

TIBCO BusinessEvents® provides sample YAML files at `BE_HOME\cloud\kubernetes\sa` to create Kubernetes resources for deploying TIBCO BusinessEvents cluster for the shared all storage.

Sample Kubernetes Resource YAML Files for Shared All Storage

File Name	Resource	Resource Type	Description
<code>db-configmap.yaml</code>	ConfigMap	ConfigMap	Setup the environment variables for the database connection. These environment variables are used by deployment instances (<code>bediscovery.yaml</code> , <code>becache.yaml</code> , and <code>beinference.yaml</code>) for connecting to the database.
<code>bediscovery.yaml</code>	Discovery node	Deployment	Setup the container (containers) with the docker image (image) of the application. Provide a label (labels) to the deployment which the discovery node service can use as selector. Specify only one replica (replicas) of the discovery node. Provide database connection values for the global variables, that are used in the application, using the ConfigMap environment variables.
<code>bediscovery-service.yaml</code>	Discovery node service	Service (Internal)	Setup the service to connect to the discovery node. Specify the label of the discovery node as the value of selector. Other nodes in the cluster use this service to connect to the discovery node. Specify the protocol and port to connect to this service.
<code>becache.yaml</code>	Cache agent node	Deployment	Setup the container (containers) with the docker image (image) of the application. Specify replicas value and start as many cache agent as specified in the value. Connect to the discovery node service using the discovery protocol and port specified in the discovery node service. Provide database connection values for the global variables, that are used in the application, using the ConfigMap environment variables.
<code>beinference.yaml</code>	Inference agent node	Deployment	Setup the container (containers) with the docker image (image) of the application. Provide a label (labels) to the deployment which the inference agent service can use as selector. Specify at least one replica (replicas) of the inference agent node. Connect to the discovery node service using the discovery protocol and port specified in the discovery node service. Provide database connection values for the global variables, that are used in the application, using the ConfigMap environment variables.

File Name	Resource	Resource Type	Description
befdservice.yaml	Inference agent service	Service (LoadBalancer/External)	Setup the service to externally connect to the inference agent. Setup the label of the inference agent as the value of the <code>selector</code> variable for connection. Specify the <code>protocol</code> and <code>port</code> to connect to this service externally.

Running RMS Applications in Kubernetes

To use TIBCO BusinessEvents® WebStudio in Kubernetes cluster, you must setup TIBCO BusinessEvents and Rule Management Server (RMS) in AWS based Kubernetes.

Prerequisites

Download and install the following CLIs on your system:

CLI	Download and Installation Instruction Link
kops	https://github.com/kubernetes/kops/blob/master/docs/aws.md
kubect1	https://kubernetes.io/docs/tasks/tools/install-kubect1/
aws	https://aws.amazon.com/cli/

Procedure

1. Set up a Kubernetes cluster on Amazon Web Services (AWS). For more information, see [Setting up a Kubernetes Cluster on AWS](#) on page 199.
2. In AWS, create an EFS file system.

For more information on the steps to create an EFS file system, see Amazon EFS documentation at <https://docs.aws.amazon.com/efs/latest/ug/gs-step-two-create-efs-resources.html>. Specify the Kubernetes cluster Virtual Private Cloud (VPC) and **Security Group** while creating a mount target for the file system. On the File Systems page, verify that the mount target shows the **Life Cycle State** as **Available**. Under **File system access**, you see the file system **DNS name**. Make a note of this DNS name.

After successful creation of the EFS file system, note its **File System ID**, which can be used for creating EFS provisioner.
3. Create the EFS provisioner and other associated resources. Specify all the connection setup values for the EFS file system in a `manifest.yaml` file and run the `kubect1` command to create the EFS provisioner.
 - a) Download the sample `manifest.yaml` file from <https://raw.githubusercontent.com/kubernetes-incubator/external-storage/master/aws/efs/deploy/manifest.yaml> and edit it according to your setup.
 - b) In the `configmap` section specify **File System ID** of the newly created EFS as the value of the `file.system.id: variable` and **Availability Zone** of the newly created EFS as the value of the `aws.region: variables`.
 - c) In the `Deployment` section, specify DNS name of the newly created EFS as the value of the `server: variable`.
 - d) Run the `kubect1` command to apply the settings in `manifest.yaml`.


```
kubect1 apply -f manifest.yaml
```

e) Ensure that the EFS provisioner pod is in the running state using the `kubectl` command.

```
kubectl get pods
```

4. As RMS is running in a Docker container, separate external storage needs to be setup for required files and artifacts. For this, create EFS based `PersistentVolumeClaim` (PVC) using the configuration files (YAML format). The following sample YAML files for creating PVCs are located at `BE_HOME\cloud\kubernetes\rms`:

Sample Kubernetes YAML Files for Creating PVCs

File Name	PVC Name	Description
<code>efs-project-pvc.yaml</code>	<code>efs-webstudio</code>	Setup the PVC for storing BusinessEvents project files.
<code>efs-ac-pvc.yaml</code>	<code>efs-security</code>	Setup the PVC for storing BusinessEvents project ACL files, such as, <code>CreditCardApplication.ac</code> .
<code>efs-shared-pvc.yaml</code>	<code>efs-shared</code>	Setup the PVC for storing RMS artifacts after hot deployment, such as, rule template instances.
<code>efs-notify-pvc.yaml</code>	<code>efs-notify</code>	Setup the PVC for storing Email notification files, such as, <code>message.stg</code> .

5. Run the `create` command of `kubectl` utility using the YAML files to create PVCs on the EFS file system.

For example, create PVCs using the sample files:

```
kubectl create -f efs-project-pvc.yaml
```

```
kubectl create -f efs-ac-pvc.yaml
```

```
kubectl create -f efs-shared-pvc.yaml
```

```
kubectl create -f efs-notify-pvc.yaml
```

6. Mount the EFS file system into the Kubernetes EC2 instance nodes.
For more information on how to mount EFS file system on EC2 instance, refer AWS Documentation at <https://docs.aws.amazon.com/efs/latest/ug/mounting-fs.html>.
After successful mounting, PVCs on EFS are available for uploading files.
7. Transfer files from your system to the respective PVCs.

- RMS project files (for example, project files at `BE_HOME\examples\standard\WebStudio\`) to the project storage PVC (`efs-webstudio`)
- RMS security files and project access control (`.ac`) files at `BE_HOME\rms\config\security` to the security storage PVC (`efs-security`).
- RMS notification related files (for example, `message.stg` at `BE_HOME\rms\config\notify`) to the notify storage PVC (`efs-notify`).

For information on how to transfer files from your system to EC2 instance, refer to the *Amazon AWS Documentation* at <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html>.

8. Define an internal JMX Kubernetes service using the supplied sample YAML configuration file `bejmx-service.yaml`.

RMS server use this service to connect to the JMX port of a BusinessEvents pod.

```
kind: Service
apiVersion: v1
metadata:
```

```

name: bejmx-service
spec:
  selector:
    dep_name: beinference-label
  ports:
    - protocol: TCP
      port: 5555
      targetPort: 5555

```

- Build the RMS Docker image for deploying to Kubernetes.

For every project add JMX configuration in `RMS.cdd`, for example:

```

<property name="ProjectName.ws.applicableEnvironments" type="string"
value="QA,PROD"/>
<property name="ProjectName.QA.ws.jmx.hotDeploy.enable" type="boolean"
value="true"/>
<property name="ProjectName.QA.ws.jmx.host" type="string" value="bejmx-
service.default.svc.cluster.local"/>
<property name="ProjectName.QA.ws.jmx.port" type="integer" value="5555"/>
<property name="ProjectName.QA.ws.jmx.user" type="string" value=""/>
<property name="ProjectName.QA.ws.jmx.password" type="string" value=""/>
<property name="ProjectName.QA.ws.jmx.clusterName"
value="CreditCardApplication"/>
<property name="ProjectName.QA.ws.jmx.agentName" value="inference-class"/>

```

For every project configure `jmx.host` as the fully qualified name (FQN) of the local JMX Kubernetes service defined earlier, for example, `bejmx-service.default.svc.cluster.local`. Also, for every project configure `jmx.port` as the JMX port number defined in the JMX Kubernetes service, for example, 5555.

- Create RMS Docker image. For more information, see [Running BusinessEvents Rule Management Server \(RMS\) in Docker](#) on page 193.
- Go to the EC2 Container Services dashboard and create a repository with the same name as the RMS Docker image. Upload the RMS image to the repository. You may use the **View Push Commands** button to view how to do that.



AWS Repository name should be same as the RMS Docker image.

For more information on how to create a repository in Amazon AWS, refer to <https://docs.aws.amazon.com/AmazonECR/latest/userguide/repository-create.html>.

- Create Kubernetes resources, required for deploying BusinessEvents cluster, using the YAML files. These resources include deployment and services for the cluster. Thus, to deploy a BusinessEvents cluster, create:

- a discovery node (pod) to start the cluster.
- a service to connect to discovery node.
- a cache agent node that connects to the discovery node service.
- an inference agent node that connects to the discovery node service.
- a service to connect to the inference agent.
- an RMS node containing RMS docker image.
- an external service to connect to the RMS node.

You can find sample YAML files at `BE_HOME\cloud\kubernetes\rms`. For details, see [Sample Kubernetes Resource Files for RMS](#) on page 209.

- Run the `create` command of `kubectl` utility using the YAML files to deploy the BusinessEvents and RMS cluster.

For example, deploy the cluster using the sample files:

```

kubectl create -f bejmx-service.yaml
kubectl create -f bediscovery-service.yaml

```

```
kubectl create -f bediscovery.yaml
kubectl create -f becache.yaml
kubectl create -f beinference.yaml
kubectl create -f beinference-service.yaml
kubectl create -f berms.yaml
kubectl create -f berms-service.yaml
```

You can get the external IP to the external RMS service, that is, TIBCO BusinessEvents WebStudio URL, using the `get services` command. You can then use that URL to connect to TIBCO BusinessEvents WebStudio from your browser.

```
kubectl get services berms-service -o wide
```

You can check the logs of individual TIBCO BusinessEvents container pods using the following command:

```
kubectl logs <pod>
```

Sample Kubernetes Resource Files for RMS

TIBCO BusinessEvents® provides sample YAML files at `BE_HOME\cloud\kubernetes\rms` to create Kubernetes resources for deploying RMS.

Sample Kubernetes Resource YAML Files for RMS

File Name	Resource	Resource Type	Description
<code>bediscovery.yaml</code>	Discovery node	Deployment	Setup the container (containers) with the docker image (image) of the application. Provide a label (labels) to the deployment which the discovery node service can use as selector. Specify only one replica (replicas) of the discovery node. Provide JMS Kubernetes service name created earlier (<code>bejmx-service.default.svc.cluster.local</code>) as the value of <code>DOCKER_HOST</code> . Specify the volume mounts (<code>volumeMounts:</code>) to use the shared PVCs created earlier.
<code>bediscovery-service.yaml</code>	Discovery node service	Service (Internal)	Setup the service to connect to the discovery node. Specify the label of the discovery node as the value of selector. Other nodes in the cluster use this service to connect to the discovery node. Specify the protocol and port to connect to this service.

File Name	Resource	Resource Type	Description
becache.yaml	Cache agent node	Deployment	Setup the container (containers) with the docker image (image) of the application. Specify replicas value and start as many cache agent as specified in the value. Connect to the discovery node service using the discovery protocol and port specified in the discovery node service. Provide JMS Kubernetes service name created earlier (bejmx-service.default.svc.cluster.local) as the value of DOCKER_HOST. Specify the volume mounts (volumeMounts:) to use the shared PVCs created earlier.
beinference.yaml	Inference agent node	Deployment	Setup the container (containers) with the docker image (image) of the application. Provide a label (labels) to the deployment which the inference agent service can use as selector. Specify at least one replica (replicas) of the inference agent node. Connect to the discovery node service using the discovery protocol and port specified in the discovery node service. Provide JMS Kubernetes service name created earlier (bejmx-service.default.svc.cluster.local) as the value of DOCKER_HOST. Specify the volume mounts (volumeMounts:) to use the shared PVCs created earlier.
beinference-service.yaml	Inference agent service	Service (LoadBalancer/External)	Setup the service to externally connect to the inference agent. Setup the label of the inference agent as the value of the selector variable for connection. Specify the protocol and port to connect to this service externally.
berms.yaml	Discovery node	Deployment	Setup the container (containers) with the RMS docker image (image). Provide a label (labels) to the deployment which the RMS node service can use as selector. Specify the volume mounts (volumeMounts:) to use the shared PVCs created earlier.
berms-service.yaml	Discovery node service	Service (LoadBalancer/External)	Setup the service to externally connect to the RMS node. Specify the label of the RMS node as the value of selector. Specify the protocol and port to connect to this service.

TIBCO Hawk Microagent Methods

TIBCO BusinessEvents Monitoring and Management component is the preferred way to deploy, monitor, and manage TIBCO BusinessEvents applications.

You can also use TIBCO Administrator for deployment and for many monitoring and management functions. To augment the monitoring and management functions in 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 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\(\)](#)
- [getEvent\(\)](#)
- [GetExecInfo\(\)](#)
- [getHostInformation\(\)](#)
- [getInstance\(\)](#)
- [getJoinTable](#)
- [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

Before using Hawk methods, enable the TIBCO Hawk microagent in the TIBCO BusinessEvents engine property file `BE_HOME/bin/be-engine.tra`.

Procedure

1. Open the `be-engine.tra` file for editing, add the following property and set it to true to enable Hawk microagent:
`Hawk.Enabled = true`
2. Set the `tibco.env.HAWK_HOME` property value to the installation location of TIBCO Hawk.
3. If you are using non-default transport parameters for TIBCO Hawk, add the following properties also for setting Hawk service, network, and daemon parameters:

```
Hawk.Service=9999
Hawk.Network=
Hawk.Daemon=
```



To enable TIBCO Hawk Console, set the following property to true in the `TIBCO_Admin_HOME/domain/domain_name/bin/tibcoadmin/domain_name.tra` file:

```
hawk.console.enabled=true
```

activateRule()

Activate 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 (optional).
URI	URI of the RuleSet.
Activated	Set if the RuleSet is activated.

deactivateRule()

Deactivate a RuleSet in the Session

Type

ACTION

Parameters

Name	Description
Session	Name of the Session
URI	URI of the RuleSet

Returns

Type	Description
Session	Name of the Session.
URI	URI of the RuleSet.
Deactivated	Is the RuleSet deactivated?

execute()

Runs a special command.

Type

ACTION_INFO

Parameters

Name	Description
Command	The special command to execute
Parameters	Parameters (optional)

Returns

Type	Description
Line	Line Number.
Name	Name.
Value	Value.

getChannels()

Retrieves Channel Info.

Type

INFO

Parameters

Name	Description
URI	URI of the Channel (optional)

Returns

Type	Description
Line	Line Number
URI	URI of the Channel.
State	Current state of the Channel

getCacheRecoveryInfo()

Gets the Cache recovery information.

Timeout (milliseconds): 10000

Type

Open, Synchronous, IMPACT_INFO

Arguments

Name	Description
Session	Name of the Session

Returns

Name	Description
Return	None

Elements

Name	Description
Line	Line number
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 Info.

Type

INFO

Parameters

Name	Description
Channel URI	URI of the Channel (optional).
Destination Name	Name of the Destination (optional).

Returns

Type	Description
Line	Line Number.
Channel URI	URI of the Channel.
Destination URI	URI of the Destination.

Type	Description
Nb in	Number of Events in.
Rate in	Rate of Events in.
Nb out	Number of Events out.
Rate out	Rate of Events out.

getEvent()

Retrieves an Event from a Session.

Type

INFO

Parameters

Name	Description
Session	Name of the Session.
Id	ID of the Event.
External	True if using the event's external ID, false if using the internal ID.

Returns

Type	Description
Line	Line 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()

Gets 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()

Gets host information properties.

Type

INFO

Parameters

Name	Description
Name	Name of host information property to get (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
Id	ID of the Instance.
External	True if using the instance's external ID, false if using the internal ID.

Returns

Type	Description
Line	Line number.
Session	Name of the Session.
Type	Attribute or Property.
Name	Name of the Attribute or Property.
Value	Value of the Attribute or Property.

getJoinTable

Retrieves a join table from the Session(s).

GetLoggerNamesWithLevels()

Gets the list of registered loggers with their current log level.

Type

INFO

Parameters

No parameters

Returns

A MAP of the registered logger names with their current log level.

getMemoryUsage()

Gets engine memory usage information.

Type

INFO

Parameters

No parameters.

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.

Type	Description
PercentUsed	Estimate of the percentage of max memory used.

getNumberOfEvents()

Gets the total number of events existing in a Session.

Type

INFO

Parameters

Name	Description
Session	Name of the Session

Returns

Type	Description
Line	Line number.
Session	Name of the Session.
Number	Total Number of Events

getNumberOfInstances()

Gets the total number of instances existing in a Session.

Type

INFO

Parameters

Name	Description
Session	Name of the Session

Returns

Type	Description
Line	Line number.
Session	Name of the Session.
Number	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	Line 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	Line Number.
Session	Name of the Session.
URI	URI of the RuleSet

Type	Description
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	Line 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	Line 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	Line 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 (optional).

Returns

Type	Description
Line	Line number.
Destination	Destination URI.
Preprocessor	Destination preprocessor URI.

getSessions()

Retrieves session names.

Type

INFO

Parameters

No parameters.

Returns

Type	Description
Line	Line 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.

Type	Description
Application Name	Name of the application.
Uptime	Time elapsed since startup.
Process ID	Process ID of the application.
Host	Name of 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	Line Number.
Session	Name of the Session.
Number of Rules Fired	Total number of rules fired since the last reset.

getTraceSinks()

Gets 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	Line 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 this is empty).

Returns

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 channels.

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 logger to a specific log level. 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, DEBUT, ALL, or OFF

Returns

Returns nothing.

SetLogLevel(Stringnameorpattern String Level)

This API can be used to set a specific logger to a specific level, such as `SetLogLevel("as.kit", "debug")` will set "as.kit" to debug where as `SetLogLevel("as*", "debug")` will set 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 then `SetLogLevel("as.kit", "info")`, then "as.kit" will be INFO.

Similarly, if you decide to invoke `SetLogLevel("as.kit", "debug")` and then `SetLogLevel("as*", "info")`, then "as.kit" which 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.

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 writes the profile data into a file specified when the Profiler was turned on.

suspendChannels()

Suspends channels.

Type

ACTION

Parameters

Name	Description
URI	URI of the Channel to suspend (optional).

Returns

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

Returns nothing.

suspendRuleServiceProvider ()

Suspends the RleServiceProvider.

Type

ACTION

Parameters

Has no parameters

Returns

nothing