

TIBCO BusinessEvents™

Administration

*Software Release 4.0.1
November 2010*

The Power to Predict™

 **TIBCO®**
The Power of Now®

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 LICENSE.PDF) 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.

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.

TIB, TIBCO, TIBCO Software, TIBCO Adapter, Predictive Business, Information Bus, The Power of Now, The Power to Predict, TIBCO BusinessEvents, TIBCO ActiveSpaces, TIBCO ActiveMatrix BusinessWorks, TIBCO Rendezvous, TIBCO Enterprise Message Service, TIBCO PortalBuilder, TIBCO Administrator, TIBCO Runtime Agent, TIBCO General Interface, and TIBCO Hawk are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

EJB, Java EE, J2EE, JMS and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Excerpts from Oracle Coherence documentation are included with permission from Oracle and/or its affiliates. Copyright © 2000, 2006 Oracle and/or its affiliates. All rights reserved.

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.TXT 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 © 2004-2010 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Figures	xiii
Tables	xv
Preface	xvii
Changes from the Previous Release of this Guide	xviii
Related Documentation	xx
TIBCO BusinessEvents Documentation	xx
Accessing BusinessEvents Functions Reference Documentation	xxi
TIBCO BusinessEvents Event Stream Processing	xxi
TIBCO BusinessEvents Decision Manager	xxii
TIBCO BusinessEvents Data Modeling	xxii
TIBCO BusinessEvents Views	xxiii
Other TIBCO Product Documentation	xxiii
Typographical Conventions	xxiv
How to Contact TIBCO Support	xxvii
Chapter 1 Administration Overview	1
Introduction	2
Chapter 2 System-Level (TRA File) Configuration	5
Various System Level Configuration Tasks	6
Java and JVM Level Settings	6
For Cache Based Object Management on AIX	6
For TIBCO Enterprise Message Service and TIBCO Rendezvous Channels	6
JMX Properties for TIBCO BusinessEvents Monitoring and Management	7
Chapter 3 CDD Configuration Procedures	9
Overview of Cluster Deployment Configuration	10
Using Properties at Different Levels	10
Order of Precedence at Runtime	11
Task Summary	12
Cluster Tab	12
Collections Tab	12
Agent Classes Tab	13

Processing Units Tab	13
Entity Metadata Settings	13
Adding a Cluster Deployment Descriptor	14
Configuring an In Memory OM Cluster	15
Configuring a Cache OM Cluster — Cluster Tab	16
Configuring Well-Known Address Cluster Member Discovery	18
Configuring the Collections Tab (All OM Types)	21
Configuring Collections	21
Configuring Log Configurations	23
Configuring the Agent Classes Tab (All OM Types)	24
Configuring the Processing Units Tab (All OM Types)	27
Overriding Global Variables in the CDD File	29
Entity-Level Configuration for Cache and Backing Store	30
Metadata Properties for Entities (Events and Concepts)	30
Chapter 4 Cluster Deployment Descriptor Reference	35
Cluster Tab — General Settings	36
Cluster Tab — Cache OM — Configuration Setting	37
Cluster Tab — Cache OM — Backing Store Settings	38
Cluster Tab — Cache OM — Backing Store Properties	42
Cluster Tab — Cache OM — Domain Objects Settings	46
Default Level Settings	46
Entity Level Object Overrides and Related Settings	48
Cluster Tab — Cache OM — Coherence Properties	50
Collections Tab — Destinations Settings and Properties	56
Collections Tab — Log Configurations Overview	59
Collections Tab — Log Configurations Settings	61
Agents Tab Settings and Properties	64
Processing Units Tab — General Settings and Properties	70
Processing Units Tab — JMS Server Reconnection Properties	75
Chapter 5 Cluster Configuration for Berkeley DB OM	77
Configuring a Berkeley DB OM Cluster	78
Cluster Tab — Berkeley DB OM Settings and Properties	79
Additional Configuration Notes	83
Chapter 6 Basic MM Configuration	85
Overview	86

Before You Begin	86
MM Runtime Architecture	86
Summary of Configuration Tasks	87
Install and Configure Software for Remote Start and Deployment	88
SSH	88
TIBCO Hawk	89
PsTools	89
Install and Configure TIBCO Hawk for Machine Level Metrics	90
Configure JMX Properties in Monitored Engine TRA Files	92
Site Topology Overview	93
Editing the Site Topology File in a Text Editor or in the Graphical Editor	93
Summary of Site Topology Configuration	93
Project, Master, and Deployed Locations of CDD and EAR Files	95
Deployment-Specific Processing Units	95
Configure the Site Topology in TIBCO BusinessEvents Studio	97
Site Topology Reference	100
Site Settings	100
Cluster Settings	100
Deployment Unit Settings	102
Processing Unit Settings	103
Host Settings	104
Specify the Site Topology File and Basic MM Settings in MM.cdd	107
MM Agent Basic Configuration Reference	109
Configuring for WKA Cluster Discovery Configuration	115
Configure MM Console Properties	117
Chapter 7 MM Metrics and Features Configuration	119
Overview of Configuring MM Alerts, Thresholds, and Actions	120
Understanding and Configuring Alerts	121
Understanding Alerts	121
Configuring Alerts	121
Alert Configuration Reference	123
Specifying the Path to an Alert Metric Value (and a Reference Value)	125
Specifying the Alert Message	126
Pane Types Reference for Alert Configuration	128
Understanding Health Metric Rules	130
Cluster Member Paths	130
Two Types of Thresholds	131
Health Metric Rule Examples	132
Configuring Health Metric Rules	135

Health Metric Rule Configuration Reference	138
Configuring Actions	142
Action Configuration Reference	143
Chapter 8 Deploying and Managing Engines with MM	145
Starting MM Server and Logging On to the Console	146
Overriding Global Variables at Deploy Time	148
Deploying and Managing Engines in the MM UI	149
Manually Copy Custom Function and Third-Party Jars at Deploy-time	149
Deploying Cluster Engines	149
Starting, Stopping, Pausing, and Resuming Cluster Engines	149
Hot Deployment	150
Deploying or Starting Engines with the Command Line Utility	151
Configuring the MM Tools Utility TRA File	151
Deploying or Starting a Remote Engine	152
mm-tools Options Reference	152
Chapter 9 Monitoring and Managing a TIBCO BusinessEvents Cluster	155
Overview of Monitoring a TIBCO BusinessEvents Cluster	156
Reference to the Metrics	156
Health Indicators and Alerts	157
Cluster Explorer	158
Predefined and Unpredefined Members	159
Inactive Members	159
Working with Cluster Explorer	161
Generating Thread Analyzer Reports	161
Working with Panels and Panes	163
Working with MM Metric Panes	163
Cluster Overview	165
Cluster Overview Pane	165
System Alerts Pane	166
Machine Overview	167
Process Overview	168
Agent Overview	170
Inference Agent Overview	173
Query Agent Overview	174
Ontology (Cache Objects) Overview	175
MM Method Reference	177
Inference Agent Methods	177
Process Methods	179

Query Agent Methods	181
Chapter 10 Deployment Using TIBCO Administrator or at the Command Line	183
Deployment Overview	184
Manually Copy Custom Function and Third-Party Jars at Deploy-time	184
Starting a TIBCO BusinessEvents Engine at the Command Line	185
Supplementary Property Files	186
Deployment in a TIBCO Administrator Domain	188
TIBCO Administration Domains	188
Property Overrides and Precedence	188
Using AppManage for Scripted Deployment to a Domain	188
Overriding Global Variable Values in TIBCO Administrator	189
Before Deploying a Project in a TIBCO Administrator Domain	190
Update Engine TRA Files for TIBCO Hawk Information	190
Set Default or Specific CDD File and Processing Unit Names	190
Enable Hot Deployment, as needed	190
Enable Service-Settable Global Variable Overrides and Build the EAR	190
Deploying a Project in a TIBCO Administrator Domain	191
Chapter 11 Hot Deployment	195
Hot Deployment Overview	196
Modifications Allowed in Hot Deployment	197
Enabling Hot Deployment	199
Performing Hot Deployment in a TIBCO Administrator Domain	200
Performing Hot Deployment Outside a TIBCO Administrator Domain	202
Modify the Project as Needed and Build the EAR File	202
Replace the EAR File that was Used to Start the Engine	202
Chapter 12 Threading Models and Tuning	203
Threading Models Overview	204
Event Preprocessor and Rete Worker Thread Options	206
Shared Queue and Threads	206
Caller's Thread	207
Dedicated Workers	208
RTC Options — Single-Threaded or Concurrent	210
Concurrent RTC	210
Single-Threaded RTC	210
Post RTC Options — Cache-aside and Write-behind	212
Cache-aside	212
Write-behind	214

Chapter 13 Configuring User Authentication	217
User Authentication Overview	218
Pluggable JAAS Login Module	218
Authentication Options	218
Configuring User Authentication	219
Authentication Property Reference	221
Chapter 14 Configuring Access Control for a Project	223
Configuring Access Control—Overview	224
Establishing the User Roles	224
Guidelines for Configuring Access Control	224
Structure of the Access Control File	225
Working with Access Control Files	227
Creating and Modifying an Access Control File	227
Required Location of Access Control Files	227
Specifying and Grouping Decision Project Resources	227
Defining Permissions	228
Defining Permissions for Domain Models	229
Resource Types and Corresponding Action Types	231
Chapter 15 JDBC Backing Store Configuration	233
JDBC Backing Store Configuration Overview	234
Special Cases	235
Names that Exceed the DBMS Maximum Column Length	235
String Properties That Exceed the Maximum Database Column Length	236
Ontology Identifiers That Use Database Key Words	236
Resources Required for Setting Up the Database	237
DBMS Software and Installation Requirements	237
Provided Configuration Resources	238
JDBC Backing Store Database Configuration Tasks	241
Updating an Existing Backing Store Database Schema	248
What the Schema Update Utility Can and Can't Handle Automatically	248
The Procedure	250
Adding a JDBC Connection Resource to the Studio Project	252
Configuring Backing Store Properties in the CDD Editor	253
Enabling Backing Store and General Settings	253
Database Connection Settings	253
If a Custom Cache Configuration File is Used	254
Specifying Limited Cache Size	254
Backing Store Table Reference	255

Primary Tables	255
Secondary Tables	256
Reverse Reference Tables	256
Class-to-Table Mapping	257
Appendix A Setting up an Oracle-Only Backing Store	259
Oracle-Only Backing Store Database Setup Overview	260
Oracle-Only Backing Store Requirements	260
Before You Begin Database Setup	260
Oracle-Only Backing Store Database Setup Tasks	260
Extra Procedure to Handle Long Database Identifier Names	261
After You Finish Database Setup	261
Maintaining a Backing Store — If Ontology Object Definitions Change	261
Resources Required for Setting Up the Oracle Database	262
Oracle-Only Backing Store Database Configuration Tasks	264
Updating an Existing Oracle Database Schema	268
Project Configuration for Oracle-Only Backing Store	270
Oracle-Only Backing Store Runtime Behavior	271
Handling Deleted Entities	272
Appendix B Determining the Engine Name	273
Appendix C Engine Startup and Shutdown Sequence	275
Startup Sequence	275
Shutdown Sequence	276
System Startup and Shutdown Sequence	276
Appendix D Advanced Caching Topics	277
Overriding and Extending the Operational Deployment Descriptor	278
Overriding Element Values in Engine Property Files	278
Defining Additional Elements and System Properties in Override Files	278
Specifying Operational Override File Locations	280
How the First Tier Override File Default Location is Specified	280
How to Specify a Different Location for the First Tier Override File	281
How a Second Tier Override File Default Location is Specified and Overridden	282
Understanding Entity Caches	283
Entity Cache Names Format	283
Caches for Ontology Objects	284
Caches for Internal Entities	284
Appendix E TIBCO Hawk Microagent Methods	287

TIBCO Hawk Methods Overview	289
Types of Methods	289
Enabling the TIBCO Hawk Microagent	289
For More Information	289
activateRuleSet()	290
activateTraceRole()	291
deactivateRuleSet()	292
deactivateTraceRole()	293
execute()	294
forceOMCheckpoint()	295
getChannels()	296
getDestinations()	297
getEvent()	298
GetExecInfo()	299
getHostInformation()	300
getInstance()	301
getMemoryUsage()	302
getNumberOfEvents()	303
getNumberOfInstances()	304
getOMInfo()	305
getRuleSet()	306
getRuleSets()	307
getScorecard()	308
getScorecards()	309
getSessionInputDestinations()	310
getSessions()	311
getStatus()	312
getTotalNumberRulesFired()	313
getTraceSinks()	314
reconnectChannels()	315
resetTotalNumberRulesFired()	316
resumeChannels()	317
resumeDestinations()	318
stopApplicationInstance()	319
suspendChannels()	320
suspendDestinations()	321

Index of Engine Properties323
Index325

Figures

Figure 1	MM Runtime Architecture.	87
Figure 2	Agent threading example — shared threads, concurrent RTC, cache-aside.	204

Tables

Table 1	General Typographical Conventions	xxiv
Table 2	Syntax Typographical Conventions	xxv
Table 3	Entity-level metadata properties for events and concepts	30
Table 4	Cluster Tab —General Settings	36
Table 5	Cluster Tab Reference — Cache OM — Configuration Setting	37
Table 6	Cluster Tab Reference — Cache OM — Backing Store Settings	38
Table 7	Cluster Tab — Cache OM — Backing Store Properties.	42
Table 8	Cluster Tab — Cache OM — Domain Object Default Settings	46
Table 9	Cluster Tab — Cache OM — Domain Object Override Settings	48
Table 10	Cluster Tab — Cache OM — Coherence Properties	50
Table 11	Destination Settings	56
Table 12	Log Configurations Settings	61
Table 13	Agents Tab Settings and Properties	64
Table 14	Processing Units Tab General Settings and Properties.	70
Table 15	Properties for Reconnecting to a JMS Server	75
Table 16	Cluster Tab —Berkeley DB OM Settings and Properties.	79
Table 17	Software Options for Deployment, Remote Start, and Remote Method Invocation	88
Table 18	Site Topology — Site Settings	100
Table 19	Site Topology — Cluster Settings	100
Table 20	Site Topology — Deployment Unit Settings	102
Table 21	Site Topology — Processing Unit Settings.	103
Table 22	Site Topology — Host Settings	104
Table 23	MM CDD mm-class Agent Class Properties	109
Table 24	MM Console Configuration Properties	117
Table 25	MM CDD mm-class Agent Class Alert Configuration	123
Table 26	Elements Used to Specify the Path to an Alert Metric Value	125
Table 27	Alert Message Parameters.	127
Table 28	Pane Type Details	128

Table 29	MM CDD mm-class Agent Class Health Metric Rule Configuration	138
Table 30	MM CDD mm-class Agent Class Action Configuration	143
Table 31	mm-tools Options	152
Table 32	Cluster-Level Metrics	165
Table 33	Machine-Level Metrics	167
Table 34	Process- Level Metrics	168
Table 35	Agent Overview (Common) Metrics	171
Table 36	Inference Agent Metrics	173
Table 37	Query Agent Metrics	174
Table 38	Cache Objects Metrics	175
Table 39	Inference Agent Methods	177
Table 40	Process Methods	179
Table 41	Query Agent Methods	181
Table 42	TIBCO BusinessEvents Command Line Engine Startup Options	185
Table 43	Hot Deployment Supported Modifications	197
Table 44	Authentication Configuration Properties	221
Table 45	Resource Types and Action Types	231
Table 46	Resources Required for JDBC Backing Store Implementation	238
Table 47	Secondary table structure	256
Table 48	Resources Required for Backing Store Implementation	262
Table 49	Internal Entity Caches	284

Preface

TIBCO BusinessEvents™ allows you to abstract and correlate meaningful business information from the events and data flowing through your information systems, and take appropriate actions using business rules. By detecting patterns within the real-time flow of events, BusinessEvents™ can help you to detect and understand unusual activities as well as recognize trends, problems, and opportunities. BusinessEvents publishes this business-critical information in real time to your critical enterprise systems or dashboards. With BusinessEvents you can predict the needs of your customers, make faster decisions, and take faster action.

BusinessEvents
The Power to Predict™

Topics

- [Changes from the Previous Release of this Guide, page xviii](#)
- [Related Documentation, page xx](#)
- [Typographical Conventions, page xxiv](#)
- [How to Contact TIBCO Support, page xxvii](#)

Changes from the Previous Release of this Guide

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

JMS Reconnection Properties

The table of reconnection properties has been moved. It was in Chapter 6, JMS Channels of *TIBCO BusinessEvents Developer's Guide*. Now the information is in the section [Processing Units Tab — JMS Server Reconnection Properties on page 75](#) of this guide.

Preloading Objects and Handles has Changed

See [Chapter 3, CDD Configuration Procedures, on page 9](#) and [Cluster Tab — Cache OM — Domain Objects Settings on page 46](#) for the new settings and related procedures.

No Preloading Settings in Entity Metadata

Preloading configuration is now done through the CDD file (see above). The [Entity-Level Configuration for Cache and Backing Store on page 30](#) section has been updated accordingly.

TIBCO BusinessEvents Monitoring and Management

Documentation is expanded in this release to explain the basic configuration in a step-wise manner, to provide more detail about deployment and managing engines, and to provide more detail about the monitoring features such as Thread Analyzer. See the following chapters:

- [Chapter 6, Basic MM Configuration, on page 85](#)
- [Chapter 8, Deploying and Managing Engines with MM, on page 145](#)
- [Chapter 9, Monitoring and Managing a TIBCO BusinessEvents Cluster, on page 155.](#)

Deployment Using TIBCO Administrator

You can now specify the CDD and processing unit to use for deployment. You can also override global variable values. The procedures have been updated accordingly. See [Chapter 10, Deployment Using TIBCO Administrator or at the Command Line, on page 183.](#)

Determining the Engine Name

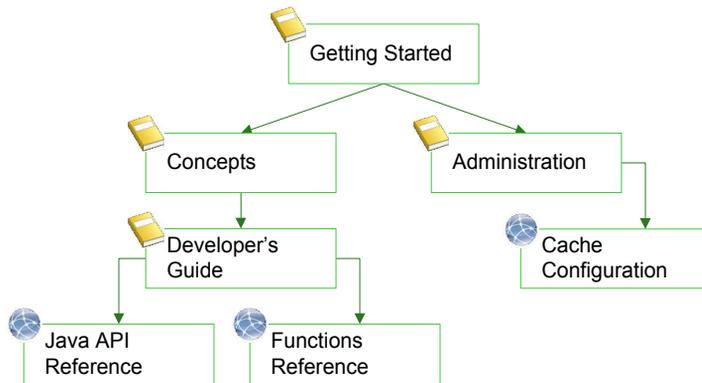
The short appendix [Appendix B, Determining the Engine Name](#), on page 273 has been updated to account for changes in how the engine name is determined in the 4.x product.

Related Documentation

This section lists documentation resources you may find useful.

TIBCO BusinessEvents Documentation

The following diagram shows the relationships between the main documents in the set.



Available in PDF and HTML formats



Available in HTML format only

In addition to the above, the set contains an installation guide, release notes, and a readme file. The complete set is described next.

- *TIBCO BusinessEvents Installation*: Read this manual for instructions on site preparation and installation.
- *TIBCO BusinessEvents Getting Started*: After the product is installed, use this manual to learn the basics of BusinessEvents. This guide provides step-by-step instructions to implement an example project and also explains the main ideas so you gain understanding as well as practical knowledge.
- *TIBCO BusinessEvents Architect's Guide*: If you are architecting an application using TIBCO BusinessEvents, read this guide for overview and detailed technical information to guide your work.
- *TIBCO BusinessEvents Developer's Guide*: After the architect has designed the system, use this guide to implement the design in TIBCO BusinessEvents Studio.

- *TIBCO BusinessEvents Administration*: This book explains how to configure, deploy, monitor, and manage a BusinessEvents application and the data it generates.
- Online References:
 - *TIBCO BusinessEvents Cache Configuration Guide*: This online reference is available from the HTML documentation interface. It provides configuration details for cache-based object management. Cache-based object management is explained in *TIBCO BusinessEvents Administration*.
 - *TIBCO BusinessEvents Java API Reference*: This online reference is available from the HTML documentation interface. It provides the Javadoc-based documentation for the BusinessEvents API.
 - *TIBCO BusinessEvents Functions Reference*: This online reference is available from the HTML documentation interface. It provides a listing of all functions provided with BusinessEvents, showing the same details as the tooltips available in the TIBCO BusinessEvents Studio rule editor interface.
- *TIBCO BusinessEvents Release Notes*: Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

Accessing BusinessEvents Functions Reference Documentation

All functions, including those used in add-ons, are documented in the HTML documentation interface for the BusinessEvents documentation set. The reference documentation is also available as tooltips in TIBCO BusinessEvents Studio.

To use the reference documentation for functions from the file system do the following:

1. Browse to `BE_HOME/doc/standard/html` and click **index.htm**. The HTML documentation interface appears.
2. In the left panel, browse to Online References and in the right panel choose TIBCO BusinessEvents Functions Reference. The reference opens in a new tab.
3. Click the navigation links to browse to the functions as desired.

TIBCO BusinessEvents Event Stream Processing

This BusinessEvents add-on is available separately, and includes the BusinessEvents Query Language features and the Pattern Matching Framework.

- *TIBCO BusinessEvents Event Stream Processing Installation*: Read this brief manual for installation instructions. A compatible version of TIBCO BusinessEvents must be installed first.

- *TIBCO BusinessEvents Query Developer's Guide*: This manual explains how to use the object query language to query various aspects of the running system.
- *TIBCO BusinessEvents Event Stream Processing Pattern Matcher Developer's Guide*: This manual explains how to use the pattern matcher language and engine to correlate event patterns in a running system.
- *TIBCO BusinessEvents Event Stream Processing Release Notes*: Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

TIBCO BusinessEvents Decision Manager



TIBCO BusinessEvents Decision Manager software does not on Linux or Solaris operating systems.

This BusinessEvents add-on is available separately. It incorporates a decision modeling business user interface, and associated runtime.

- *TIBCO BusinessEvents Decision Manager Installation*: Read this brief manual for installation instructions. A compatible version of TIBCO BusinessEvents must be installed first.
- *TIBCO BusinessEvents Decision Manager User's Guide*: This manual explains how business users can use decision tables and other decision artifacts to create business rules. It also covers configuration and administration of Rules Management Server, which is used for authentication, authorization, and approval processes.
- *TIBCO BusinessEvents Decision Manager Release Notes*: Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

TIBCO BusinessEvents Data Modeling

This BusinessEvents add-on is available separately. It contains state models and database concept features.

- *TIBCO BusinessEvents Data Modeling Installation*: Read this brief manual for installation instructions. A compatible version of TIBCO BusinessEvents must be installed first.
- *TIBCO BusinessEvents Data Modeling Developer's Guide*: This manual explains data modeling add-on features for BusinessEvents. The database concepts feature enables you to model BusinessEvents concepts on Database tables. The state modeler feature enables you to create state machines.

- *TIBCO BusinessEvents Data Modeling Release Notes*: Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

TIBCO BusinessEvents Views

This BusinessEvents add-on is available separately. It includes graphical dashboard components for run-time event monitoring.

- *TIBCO BusinessEvents Views Installation*: Read this manual for instructions on site preparation and installation.
- *TIBCO BusinessEvents Views Developer's Guide*: This guide explains how to use BusinessEvents TIBCO BusinessEvents Views to create meaningful metrics that are presented to business users in real-time for proactive decision making.
- *TIBCO BusinessEvents Views User's Guide*: This book explains how to monitor metrics in BusinessEvents TIBCO BusinessEvents Views and how to represent the business processes graphically.
- *TIBCO BusinessEvents Views Release Notes*: Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

Other TIBCO Product Documentation

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

- TIBCO ActiveSpaces®
- TIBCO Hawk®
- TIBCO Rendezvous®
- TIBCO Enterprise Message Service™
- TIBCO ActiveMatrix BusinessWorks™

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
<i>TIBCO_HOME</i> <i>ENV_HOME</i> <i>BE_HOME</i>	<p>Many TIBCO products must be installed within the same home directory. This directory is referenced in documentation as <i>TIBCO_HOME</i>. The value of <i>TIBCO_HOME</i> depends on the operating system. For example, on Windows systems, the default value is <code>C:\tibco</code>.</p> <p>Other TIBCO products are installed into an installation environment. Incompatible products and multiple instances of the same product are installed into different installation environments. The directory into which such products are installed is referenced in documentation as <i>ENV_HOME</i>. The value of <i>ENV_HOME</i> depends on the operating system. For example, on Windows systems the default value is <code>C:\tibco</code>.</p> <p>TIBCO BusinessEvents installs into a directory within <i>ENV_HOME</i>. This directory is referenced in documentation as <i>BE_HOME</i>. The value of <i>BE_HOME</i> depends on the operating system. For example on Windows systems, the default value is <code>C:\tibco\be\4.0</code>.</p>
code font	<p>Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:</p> <p>Use <code>MyCommand</code> to start the foo process.</p>
bold code font	<p>Bold code font is used in the following ways:</p> <ul style="list-style-type: none"> • In procedures, to indicate what a user types. For example: Type admin. • In large code samples, to indicate the parts of the sample that are of particular interest. • In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, <code>MyCommand</code> is enabled: <code>MyCommand [enable disable]</code>

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
<i>italic font</i>	<p>Italic font is used in the following ways:</p> <ul style="list-style-type: none"> • To indicate a document title. For example: See <i>TIBCO BusinessWorks Concepts</i>. • To introduce new terms. For example: A portal page may contain several <i>portlets</i>. Portlets are mini-applications that run in a portal. • To indicate a variable in a command or code syntax that you must replace. For example: <code>MyCommand <i>pathname</i></code>
Key combinations	<p>Key name separated by a plus sign indicate keys pressed simultaneously. For example: <code>Ctrl+C</code>.</p> <p>Key names separated by a comma and space indicate keys pressed one after the other. For example: <code>Esc, Ctrl+Q</code>.</p>
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2 Syntax Typographical Conventions

Convention	Use
[]	<p>An optional item in a command or code syntax.</p> <p>For example:</p> <pre>MyCommand [optional_parameter] required_parameter</pre>
	<p>A logical 'OR' that separates multiple items of which only one may be chosen.</p> <p>For example, you can select only one of the following parameters:</p> <pre>MyCommand param1 param2 param3</pre>

Table 2 *Syntax Typographical Conventions*

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

How to Contact TIBCO Support

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

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

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

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

<https://support.tibco.com>

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

Chapter 1 **Administration Overview**

This brief chapter outlines the administration tasks.

Topics

- [Introduction, page 2](#)

Introduction

This guide documents what you must do after a TIBCO BusinessEvents Studio project is ready to be prepared for deployment and use.

The main administration tasks and the chapters in which they are documented are as follows:

- System level configuration as needed. Edit the engine TRA file for settings are read before the engine starts.
 - [Chapter 2, System-Level \(TRA File\) Configuration, page 5](#)
- Configuring the Cluster Deployment Descriptor (CDD) for cluster settings, object management settings, and deploytime configuration of processing units and agents. In addition some entity metadata settings set various defaults that interact with the CDD settings.
 - [Chapter 3, CDD Configuration Procedures, page 9](#)
 - [Chapter 4, Cluster Deployment Descriptor Reference, page 35](#)
 - [Chapter 5, Cluster Configuration for Berkeley DB OM, page 77](#)
 - [Threading Models and Tuning on page 203](#)
- Configuring the TIBCO BusinessEvents Monitoring and Management (MM) component. Two kinds of configuration are documented: basic configuration is about connecting with the cluster to be monitored, 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. The second type of configuration is to set up the MM settings such as health level metric thresholds, alerts, and actions.
 - [Chapter 6, Basic MM Configuration, page 85](#)
 - [Chapter 7, MM Metrics and Features Configuration, page 119](#)
- Starting MM, deploying monitoring, and managing a Cache based cluster.
 - [Chapter 8, Deploying and Managing Engines with MM, page 145](#)
 - [Chapter 9, Monitoring and Managing a TIBCO BusinessEvents Cluster, page 155](#)
- Using TIBCO Administrator to deploy and manage a Cache based cluster.
 - [Chapter 10, Deployment Using TIBCO Administrator or at the Command Line, page 183](#)

- You may also wish to use the hot-deployment feature.
 - [Chapter 11, Hot Deployment, page 195](#)
- Configuring authentication and authorization. Certain components use authentication (BEMM, TIBCO BusinessEvents Views, TIBCO BusinessEvents Decision Manager) and certain components use authorization (TIBCO BusinessEvents Decision Manager).
 - [Chapter 13, Configuring User Authentication, page 217](#)
 - [Chapter 14, Configuring Access Control for a Project, page 223](#)
- Most production systems use Cache OM with a backing store. The JDBC backing store implementation is the preferred implementation. The legacy Oracle-only backing store implementation is deprecated.
 - [Chapter 15, JDBC Backing Store Configuration, page 233](#)
 - [Appendix A, Setting up an Oracle-Only Backing Store, page 259](#)
- Appendixes provide further information that may be useful
 - [Appendix B, Determining the Engine Name, page 273](#)
 - [Appendix C, Engine Startup and Shutdown Sequence, page 275](#)
 - [Appendix D, Advanced Caching Topics, page 277](#)
 - [Appendix E, TIBCO Hawk Microagent Methods, page 287](#)

Chapter 2 **System-Level (TRA File) Configuration**

The engine executable files each have an associated configuration file with the extension `.tra`. These files are updated only for system-level property settings. All other runtime project configuration is entered in the CDD file.

Topics

- [Various System Level Configuration Tasks, page 6](#)

Various System Level Configuration Tasks

As needed, configure the TRA file for system-level settings that must be set before the TIBCO BusinessEvents engine starts. Other settings go in the CDD file.

You should also be aware of the `BE_HOME/studio/eclipse/studio.tra`. Also in the same location, `config.ini` file has Eclipse related settings.

Java and JVM Level Settings

TRA file settings at this level are not documented here. See Java documentation as needed. The TRA file also contains some helpful comments for such properties.

For Cache Based Object Management on AIX

When TIBCO BusinessEvents installed on AIX uses cache-based object management, you must add this property to all TRA files, and set the value to true.

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

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

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



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. See [Chapter 6, Basic MM Configuration, on page 85](#) for details.

TIBCO BusinessEvents Views and TIBCO BusinessEvents Decision Manager also have engine TRA files you must update.

For TIBCO Enterprise Message Service and TIBCO Rendezvous Channels

For JMS Channels To use a TIBCO Enterprise Message Service server as a JMS destination, you must do one of the following:

- If the software is installed locally, set the `EMS_HOME` variable in the `BE_HOME/bin/be-engine.tra` files.

- If the software is not installed locally you must copy the `jms.jar` and `tibjms.jar` files to `BE_HOME/lib/ext` and also add them to the classpath in the `be-engine.tra` files.

For Rendezvous Channels To use TIBCO Rendezvous destination, you must set the `RV_HOME` variable in the `BE_HOME/bin/be-engine.tra` files.

JMX Properties for TIBCO BusinessEvents Monitoring and Management

For the MM component to monitor the cluster, various JMX settings are required. Settings are provided in the TRA file and must be uncommented.

For details see [Configure JMX Properties in Monitored Engine TRA Files on page 92](#).

Chapter 3 CDD Configuration Procedures

This chapter explains how to configure a cluster for deployment, using the Cluster Deployment Descriptor (CDD) editor.

Reference tables for procedures in this manual are provided in [Chapter 4, Cluster Deployment Descriptor Reference](#), on page 35.

Configuration for Berkeley DB OM (deprecated feature) is provided in [Chapter 5, Cluster Configuration for Berkeley DB OM](#), on page 77.

Topics

- [Overview of Cluster Deployment Configuration](#), page 10
- [Task Summary](#), page 12
- [Adding a Cluster Deployment Descriptor](#), page 14
- [Configuring an In Memory OM Cluster](#), page 15
- [Configuring a Cache OM Cluster — Cluster Tab](#), page 16
- [Configuring the Collections Tab \(All OM Types\)](#), page 21
- [Configuring the Agent Classes Tab \(All OM Types\)](#), page 24
- [Configuring the Processing Units Tab \(All OM Types\)](#), page 27
- [Entity-Level Configuration for Cache and Backing Store](#), page 30

Overview of Cluster Deployment Configuration

This chapter explains how to configure deployment settings for a TIBCO BusinessEvents cluster. Configuration is done using the Cluster Deployment Descriptor (CDD) editor. Settings are saved in XML, to the CDD file.

This chapter assumes you have read and understood the object management and other deploy-time configuration topics in *TIBCO BusinessEvents Architect's Guide*, especially Chapter 6, Object Management Options and Chapter 7, Distributed Cache OM.

Basic settings for all object management (OM) types relate to the following:

- Distribution of project resources among the agent classes.
- Defining which agent classes are deployed in each processing unit.
- Which OM type to use.
 - In Memory OM
 - Berkeley DB OM (deprecated)
 - Cache OM

Processing units configured in the CDD editor are used in the Site Topology editor, for deployment using the TIBCO BusinessEvents Monitoring and Management component. You map these logical units to the physical resources in the deployment environment (also known as the deployment topology). See [Site Topology Overview on page 93](#) and sections following.

Additional settings are used if certain features or add-ons are used only, for example HTTP channel properties and database concepts properties.

Using Properties at Different Levels

The scope of a property depends on the property sheet you add it to. Not all properties are valid at all levels. Use your judgment.

For example, properties that include the agent class name, such as `Agent.AgentClassName.checkDuplicates`, can be used this way. Here is the scope of each level for the `AgentClassName` properties:

Cluster level Applies to all `AgentClassName` agents in the cluster.

Processing unit level Applies any `AgentClassName` agent deployed in the specified processing unit.

Agent class level applies to any `AgentClassName` agent, used in any processing unit

Order of Precedence at Runtime

Note that the TRA files should be used only for system-level settings that must be read before the JVM starts. All other properties are set in the CDD.

- Properties in the deployed TRA file override properties and settings in the CDD file.
- CDD override properties override CDD settings (in predefined fields).
- CDD properties set at a lower level override CDD properties set at a higher level. Cluster is the highest level, then processing unit, then agent.
- CDD properties override EAR file properties.



Global variable overrides in the CDD file are ignored if you deploy using TIBCO BusinessEvents Monitoring and Management or TIBCO Administrator. Instead, define global variable overrides in those components. They add the overrides to the end of the CDD file.

Task Summary

Tasks are divided between the bottom tabs as explained in this section. The Source tab enables you to view the XML file generated by the editor. It is strongly recommended that you do not manually edit this file. In addition entity metadata settings play a role and are documented in this chapter.

Cluster Tab

At the Cluster tab, you select an object management (OM) type and configure as needed:

- For In Memory OM see [Configuring an In Memory OM Cluster, page 15](#).
- For Berkeley DB OM see See [Chapter 5, Cluster Configuration for Berkeley DB OM, on page 77](#).



Do not use Berkeley DB OM for new projects: it is a deprecated feature.

- For Cache OM, see [Configuring a Cache OM Cluster — Cluster Tab, page 16](#). This OM is generally used for production deployments. A backing store is also used in most cases.

Collections Tab

At the Collections tab, you can, as desired, group rules, rule functions, and destinations into collections so that they can be easily assigned to agent classes (and processing units in the case of log configurations).



Remember to put rule functions for use at start up into different groups from those used at shut down so you can select them appropriately at the agent classes tab.

Destinations require additional configuration, which can be done in this tab. (Destinations that are added to agent classes individually can be configured at the Agent Classes tab.)

You also set up log configurations as needed at this tab.

See [Configuring the Collections Tab \(All OM Types\), page 21](#).

Agent Classes Tab

At the Agent Classes tab, you define properties for the type of agent or agents you plan to deploy. Berkeley DB OM and In Memory OM use only inference agent classes. For cache OM, you configure agent classes for different types of agents: cache agents, inference agents, query agents (if TIBCO BusinessEvents Event Stream Processing add-on is used), and dashboard agents (if TIBCO BusinessEvents Views is used). See [Configuring the Agent Classes Tab \(All OM Types\)](#), page 24.

The "Monitoring & Management" agent type is used only for the MM server.

Processing Units Tab

In the Processing Units tab, you define which agents to include in the processing unit, and which logging configuration to use. Depending on the OM, you also configure some additional settings. See [Configuring the Processing Units Tab \(All OM Types\)](#), page 27.

Entity Metadata Settings

Certain entity metadata settings set backing store and other runtime behavior. For this reason they are included in this chapter. See [Metadata Properties for Entities \(Events and Concepts\)](#) on page 30.

Adding a Cluster Deployment Descriptor

The first step in configuring the CDD is to provide a name, and choose an object management type. The type you choose determines many of the configuration choices you make in the CDD editor.



Names in the CDD must conform to the NCName datatype. See the following page for more details:

<http://www.w3.org/TR/REC-xml-names/#NT-NCName>

Add the CDD File and Select an Object Management Type

1. In Studio Explorer, right click the folder where you want to store the CDD and select **New > Cluster Deployment Descriptor**. You see the New Cluster Configuration Wizard.
2. In the File name field, type a name for the CDD and click **Next**. (You can change the name in the editor as desired).

For deployment, TIBCO Administrator by default looks for a processing unit called default and a CDD file called default.

3. At the Object Management Selection page, select an object management type for the deployment, then click **Finish**.



When you choose an object management type in the wizard, defaults and common properties for that OM type are added in the CDD. If you later switch to a different object management type (at the Cluster tab), those defaults and properties remain. You must manually reconfigure the CDD for the object management type you change to — or, if you are just beginning, simply start over, and select the correct object management type in the wizard.

Configuration instructions for each object management type are in the following sections:

- [Configuring an In Memory OM Cluster on page 15](#)
- [Configuring a Cache OM Cluster — Cluster Tab on page 16](#)
- [Configuring a Berkeley DB OM Cluster on page 78](#)

Configuring an In Memory OM Cluster

Use this procedure if you picked In Memory OM in the second page of the New Cluster Configuration wizard.

For the wizard procedure, see [Adding a Cluster Deployment Descriptor on page 14](#).

1. In the Cluster tab click **General**. On the right, specify the following:
 - The cluster name and message encoding.
 - As desired, an Author name, any comment you wish to record. (Version and date are not editable.)See [Cluster Tab — General Settings on page 36](#) for details.
2. **Object Management** should be set to In Memory. If not you can right-click and choose **Change to In Memory**.
See the note in the section [Adding a Cluster Deployment Descriptor on page 14](#) for implications of switching here.
3. Save. No other cluster-level configuration is required for In Memory clusters.
4. Go to [Configuring the Collections Tab \(All OM Types\) on page 21](#) to continue configuration.

Configuring a Cache OM Cluster — Cluster Tab

Use this procedure if you picked Cache OM in the second page of the New Cluster Configuration wizard.

For the wizard procedure, see [Adding a Cluster Deployment Descriptor on page 14](#).

To Configure the Cluster Tab for Cache OM

See [Chapter 4, Cluster Deployment Descriptor Reference, on page 35](#) for reference tables mentioned in this section.

1. In the Cluster tab click **General**. On the right, specify the following:
 - The cluster name and message encoding.
 - As desired, an Author name, any comment you wish to record. (Version and date are not editable.)

See [Cluster Tab — General Settings on page 36](#) for details.

2. If Cache OM is not already selected, right-click the Object Management node and choose **Change to Cache**.

A tree of option groups appears on the left. Click an option to display an appropriate configuration panel on the right.

3. In the panel on the left, click **Properties**. Expand to see the provided coherence group of properties. Add properties as needed and provide values as explained next.

Backup Count

4. In the following property, define the number of members of the distributed cache service (that is, cache agents) that hold the backup data for each unit of storage in the cache.

```
tangosol.coherence.distributed.backupcount
```

See [Backup Count on page 50](#) for more details.

- Cluster Discovery 5. Configure properties for cluster discovery. To see the property sheet, click **Properties** in the list on the left.



You can use either multicast or well-known-address (WKA) discovery, as appropriate. See Cache Cluster Discovery in *TIBCO BusinessEvents Architect's Guide* for guidelines about which option is right for your use. Multicast is more commonly used and that is what's explained in this procedure. To see instructions for well-known address configuration see [Configuring Well-Known Address Cluster Member Discovery on page 18](#). Then continue at [step 7](#) after you have finished the WKA configuration.

To Use Multicast Cluster Member Discovery Multicast is the default discovery method, and defaults for multicast discovery are appropriate for many environments. Specify values if defaults for the following properties are not appropriate in your environment:

```
tangosol.coherence.clusteraddress
tangosol.coherence.clusterport
tangosol.coherence.ttl
```



Specifying one or more well-known addresses disables all multicast communication. Remove any well-known address properties, if any.

See [Multicast Discovery Properties on page 51](#) for details

6. For multicast discovery you may also need to set these properties as explained in [Localhost and Localport Properties on page 51](#):

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

- Other Properties 7. In some circumstances you may need to add more cluster-related properties such as the following:

```
tangosol.coherence.override
tangosol.coherence.cacheconfig
```

See [Other Coherence Properties on page 55](#) for details on the above.

- Backing Store 8. If you are using a backing store do the following:
- Click **Backing Store** in the tree on the left and complete the fields on the right.
 - Click **Backing Store > Connection** and complete the fields on the right to reference the JDBC connection for the backing store, and the pool settings.
 - Add additional properties to the Cluster properties sheet as needed (if you are using Oracle Strategy).

See [Cluster Tab — Cache OM — Backing Store Settings on page 38](#) for details.

- Domain Objects
9. Click **Domain Objects > Default** and specify the default object management settings for domain objects (entities). See [Table 8, Cluster Tab — Cache OM — Domain Object Default Settings, on page 46](#) for settings.
 - In the Mode field select Cache Only, Cache+Memory, or Memory Only.
 - Select whether to preload entities, preload handles, or both at startup.
 - Also specify the preload fetch size, to preload a subset of the objects into the cache or object table. The setting applies to all preloaded objects and object handles.
 10. As desired, add domain object override entries. Any entity types not overridden use the Default level settings. See [Table 9, Cluster Tab — Cache OM — Domain Object Override Settings, on page 48](#) for override settings.
 - a. In the tree on the left, click **Overrides** and click **Add**.
 - b. Select the `/uri` entry that appears on the left, and configure the settings on the right. You can use the defaults set in [step 9](#), or override them

You can also edit existing override entries, and remove entries not needed (by clicking Remove).
 11. If the object uses Cache+Memory, also set the Pre-processor setting and the Subscribe Cluster setting as needed.
- Database Concepts
12. If you use database concepts, available in the TIBCO BusinessEvents Data Modeling add-on product, select Database Concepts on the left and complete the fields in the configuration panel on the right. See *TIBCO BusinessEvents Data Modeling Developer's Guide* for details.

Go to [Configuring the Collections Tab \(All OM Types\) on page 21](#) to continue configuration.

Configuring Well-Known Address Cluster Member Discovery

If multicast cluster member discovery will not be used in your environment, configure well-known address discovery instead. At least one machine configured as a WKA machine must be running in order for other machines to join the cluster.

To understand how well-known address discovery works see Cluster Member Discovery Using Well-Known-Addresses in *TIBCO BusinessEvents Architect's Guide*.

To Configure Well-Known Address Cluster Member Discovery

This is a sub-procedure of [To Configure the Cluster Tab for Cache OM on page 16](#).

To configure well-known address discovery, you must configure machine-specific settings at the cluster level, and at the processing unit level. Machines where WKA PUs are to be deployed are called WKA machines in this procedure.

One processing unit deployed to a WKA machine must have the additional WKA configuration. Additional processing units can be deployed to a WKA machine, configured in the usual way, and they will discover and join the cluster in the usual way at runtime.

For details about the properties, see [Well-Known Address Properties on page 54](#).

1. Follow the procedure [To Configure the Cluster Tab for Cache OM on page 16](#) through [step 4](#).
2. Open the cluster tab Properties sheet and add a pair of WKA properties for each machine you want to configure as a well-known address machine:

```
tangosol.coherence.wka1 HostIP
tangosol.coherence.wka1.port Hostport
```

For example at the cluster level you might have these two WKA machines:

```
Property: tangosol.coherence.wka1 Value: 10.97.118.151
```

```
Property: tangosol.coherence.wka1.port Value: 8098
```

```
Property: tangosol.coherence.wka2 Value: 10.97.118.152
```

```
Property: tangosol.coherence.wka2.port Value: 8098
```

3. Open the Processing Units tab properties sheet for a processing unit (PU). Configure one set of WKA properties to match a cluster level set of WKA properties:

```
tangosol.coherence.localhost HostIP
tangosol.coherence.localport Hostport
```



If you will deploy using BEMM, also set the following property:

```
be.engine.hostaddress HostIP
```

and set same the value as in `tangosol.coherence.localhost`. See [Configuring for WKA Cluster Discovery Configuration on page 115](#) for more steps you must take.

(For better organization, you can put these properties into a property group, named as desired.)

Make sure the PU level property values match corresponding values set at the cluster level. That is, set the `localhost` property to the IP of the host where

you will deploy the PU, and set the `localport` property to the port defined in the cluster properties `localport` property.

Repeat this step until you have configured one PU with matching PU-level properties for each cluster-level set of WKA properties.

Each of these PUs must be deployed to the machine with the specified IP.

For example, if you configured the two well-known addresses shown in [step 2](#), then at the processing unit level you would configure a processing unit to be deployed on each of those machines as follows:

Property: `tangosol.coherence.localhost`. Value: `10.97.118.151`

Property: `tangosol.coherence.localport`. Value: `8098`

Property: `tangosol.coherence.localhost`. Value: `10.97.118.152`

Property: `tangosol.coherence.localport`. Value: `8098`

At deploy time you must select those processing units to deploy on the appropriate (matching) WKA machine. It can be helpful if the name of the PU contains the machine name or other reminder that this PU must be deployed to a specific machine.

See [Localhost and Localport Properties on page 51](#)

4. Continue at [step 7](#) after you have finished the WKA configuration to complete cluster-level configuration.

Configuring the Collections Tab (All OM Types)

The purpose of collections is to provide collections of rules, rule functions, and destinations. These collections are a convenience feature used when configuring agent classes.

Two collections are predefined: an all-rules collection and an all-functions collection.

Log Configurations

Also defined at this tab are log configurations. Because they require more explanation, details are provided in [Collections Tab — Log Configurations Overview on page 59](#), and [Collections Tab — Log Configurations Settings on page 61](#)

Configuring Collections

The various kinds of collections are configured in the same way, except that destinations and log configurations have additional configuration properties.

Before Defining Collections

First define the rules, functions, and destinations (project resources) that you will use to configure the collections. (You can add also resources during CDD configuration and return to the CDD editor to continue.) Adding project resources is documented in *TIBCO BusinessEvents Developer's Guide*.

Using References

A collection can have references to items (rules, rule functions, or destinations), and also references to other collections of the same type. References are identified in the groups tree by a reference symbol (↗). This mechanism enables you to reuse collections for more efficient configuration.

To Configure Collections

The procedure is in general the same for rules, destinations, and functions, so in these instructions, the word *item* is used to refer to the specific rule or rule function or destination.

For the log configurations procedure, see [To Add a Log Configuration on page 23](#)

1. In the Collections tab do any of the following:
 - To add a new collection, select the parent for the collection type, Rules, Destinations, Functions, or Log Configurations as needed, and click **Add**.
 - In the *Item* Collection field that appears on the right, enter a name for the group and click **Add** again.
 - To add *items* and *item* group references to a collection select the item collection and then click **Add**.

You see the Select *Items* dialog.

2. In the Select *Items* dialog do any of the following:
 - To add *items*, in the **Items** tree click the checkboxes of *items* you want to add to the group you are defining.
 - To add collection references, in the **Collection References** tree click the checkboxes of collections you want to add (by reference) to the collection you are defining.

When you select a collection on the left, you see details on the right: For example, the path to item you selected, and the names of collections you selected.

3. For function collections only, reorder the functions as needed, so that they execute in the correct order at runtime (that is, at startup or shutdown). Highlight a rule function in the tree on the left, and then click Move Up or Move Down as needed.
4. For destination collections only, configure each destination in turn. Select the destination on the left and complete the settings on the right to define characteristics such as the threading model to use, and the event preprocessor. See [Collections Tab — Destinations Settings and Properties on page 56](#) for information about each setting.
5. Save.

To Update Collections

- To remove an item in a collection or the collection itself, select the item or the group on the left and click **Remove**.
- To reorder rule functions in a function collection, select a rule function in the tree on the left, then click Move Up or Move down. This is important for startup and shutdown rule functions. Ensure that you put startup and shutdown rule functions into appropriate separate collections.

- You can change the URI (project path) of project resources to match their actual locations. To change the URI of an item, select the item on the left and edit the URI on the right.

Configuring Log Configurations

To Add a Log Configuration

1. In the Collections tab select Log Configurations and click **Add**.
2. In the Configuration panel, give the log configuration a name.
3. Add the log levels you want to enable in this configuration. See [Collections Tab — Log Configurations Overview on page 59](#) for an explanation of the logging levels, modules, and syntax details and see [Collections Tab — Log Configurations Settings on page 61](#) for details on the fields.
4. If you want to send the log output to files, in the Files area click **Enable** and configure the fields.
5. If you want to redirect the STDERR and STDOUT streams to files, in the Send to Terminal area click **Enable** and specify the file names and locations.
6. If you want to use a custom line layout, click **Enable** and configure the fields.
7. Save.

Configuring the Agent Classes Tab (All OM Types)

Various agent types are available depending on the object management (OM) type and on the add-on products used:

- **Inference Agent:** Used with all OM types.
- **Cache Agent:** Used with Cache OM only.
- **Query Agent:** Used with Cache OM only and available only if TIBCO BusinessEvents Event Stream Processing software is used.
- **Dashboard:** Used with Cache OM only and available only if TIBCO BusinessEvents Views software is used. Also requires JDBC backing store.
- **Monitoring and Management (shown as Monitoring & Management):** Used internally by the Monitoring and Management component. Do not add any agents of this class. See [Chapter 6, Basic MM Configuration, page 85](#) for more details.

When you choose an object management type in the New Cluster Configuration wizard, one agent class of each type available for that OM type is provided ready for you to configure. The exception is the monitoring and management agent type, which is used internally.

Using Collections

In the Agent Classes section (on the left) you see categories of collections. They contain any collections you configured at the Collections tab (see [Configuring the Collections Tab \(All OM Types\) on page 21](#)). These can help you quickly configure an agent.

Rule Collections It can be convenient to organize rules into collections for use in different agents. Select rule collections and individual rules as needed to define what rules will execute on an agent at runtime. (Query agents don't have rule collections.)

Destination Collections Different agents may use different sets of destinations (as configured for channels in the project). Select destination collections and individual destinations as needed to define an agent's listeners.

Startup and Shutdown Functions Collections Select function collections and individual functions as needed, to define which functions execute at engine startup and shutdown respectively. The order of the functions (including the order of functions in collections) is the order in which they execute at runtime.

To Add an Agent Class

You can begin by configuring classes provided by the wizard. You can rename the classes as desired. Then add more classes as needed.

1. In the Agent Classes tab, click **Add Agent**.
2. In the New Agent Class dialog enter an Agent Class Name.
3. Select an Agent Class Type from the list. Valid types for your cluster depend on object management type, and whether you use any TIBCO BusinessEvents add-on products.
 - Inference Agent: Used by all OM types
 - Cache Agent: Used with Cache OM only.
 - Query Agent: Used with Cache OM only and available only if TIBCO BusinessEvents Event Stream Processing is used.
 - Dashboard: Used with Cache OM only and available only if TIBCO BusinessEvents Views is used.
4. Click OK to return to the main page. The new agent name appears on the left.

To Configure an Inference, Cache, Dashboard, or Query Agent Class

1. As needed complete steps in [To Add an Agent Class on page 24](#). Select the Agent Class type, Inference, Cache, Dashboard, or Query
2. Select the agent name on the left. Appropriate settings for that agent type appear in the Configuration panel. Complete the settings and add any additional properties, as explained in [Agents Tab Settings and Properties on page 64](#).
3. For inference, dashboard, and query agent types, configure the collections you want to use. In the agent tree on the left, click each type of collection in turn and configure as explained next. (In the instructions below, the word *item* stands in for destination, function, and rule.)
 - a. Highlight a category of collections (for example Destination Collections).
 - b. Click **Add**. You see the Select *items* dialog.
 - c. In the upper section of the dialog, select individual project *item* resources, as desired.
 - d. In the lower section of the dialog (the Reference Groups section), select *item* collections you defined earlier, as desired.
 - e. Click **OK**. A list of *item* IDs appears in the box on the right.
4. If you added any individual destinations to the Destination Groups category, highlight their name on the left and configure their settings on the right. See [Collections Tab — Destinations Settings and Properties on page 56](#) for details.

(Destinations within destination collections are configured at the Collections tab.)

5. You can do any of the following:
 - Click a collection category on the left to see a list of collections and *items* you selected from that category on the right.
 - Expand a category on the left and click a collection reference within it. You see a list of its item IDs and paths, and any collection references within that collection, on the right.
 - Edit the project paths for individual items you add here. You would do this if the project location of that item changed.
6. Save.

Configuring the Processing Units Tab (All OM Types)

To configure a processing unit, you add the items you configured earlier, and any additional properties required.

If you don't find a configuration item you require, click the appropriate tab and add it, then return to Processing Units tab and continue configuration.

Processing units are used in deploytime configuration, in the topology file. One processing unit named default is provided out of the box. You can change this name. It has no significance.

To Add a Processing Unit

See [Processing Units Tab — General Settings and Properties on page 70](#) for guidelines on the settings and properties.

1. At the Processing Units tab do the following:
 - Select the default processing unit and configure it. You can rename it as needed.
 - Click **Add** to add more processing units as needed.
2. In the Name field, enter the name for the processing unit as needed.

For deployment, TIBCO Administrator by default looks for a processing unit called default and a CDD file called default.
3. In the Log Configuration field, browse to and select one log configuration.
4. Check the Hot Deploy checkbox if you want to enable hot deployment. See [Chapter 11, Hot Deployment, on page 195](#) for details.
5. If you use TIBCO BusinessEvents Data Modeling, check the Enable DB Concepts checkbox to enable database concept functionality on this processing unit.
6. In the Agents section, click **Add** and select an agent.

7. If needed, assign to each agent a key and a priority.



Agent Instance Properties and Deployment The agent key identifies an instance of this agent class uniquely at runtime. It is used to retrieve scorecards from the backing store. Scorecards are local to an agent instance. The agent priority also differentiates between deployed instances of an agent class for failover and fallback purposes if you use fault tolerance.

Ensure that you deploy a processing unit containing such agent-specific values one time only. Create as many processing units as you need to provide appropriate values in deployed instances of the agent class. See [Deployment-Specific Processing Units on page 95](#) for more details.

8. In the Properties section, add any additional configuration properties as required. For example see [Localhost and Localport Properties on page 51](#) for one use case.



The Http Properties for a processing unit are used to configure the internal HTTP server, used with the HTTP Channel. See HTTP Channel Configuration Properties in *TIBCO BusinessEvents Developer's Guide*.

Overriding Global Variables in the CDD File

Global variables are added in the TIBCO BusinessEvents Studio Global Variables editor. See *Working with Global Variables in TIBCO BusinessEvents Developer's Guide* for details.

If you plan to start TIBCO BusinessEvents engines at the command line, you can override the global variable values in the CDD file as explained here.



Global variable overrides set here are ignored by TIBCO Administrator and TIBCO BusinessEvents Monitoring and Management (BEMM) If you will deploy using TIBCO Administrator or BEMM, add global variable overrides using the features provided in TIBCO Administrator or BEMM (whichever you will use), not in the CDD.

To Override Global Variables in the CDD

Add properties using this format:

```
tibco.clientVar.GVName
```

The value of *GVName* must exactly match the value set in the TIBCO BusinessEvents Studio Global Variables editor.

Provide the override value as the property value.

Add such properties at the appropriate level in the CDD, depending on the desired scope of the override: cluster, processing unit, or agent class.

Entity-Level Configuration for Cache and Backing Store

Using their entity resource metadata properties, you can configure individual concepts and events, both at the type and at the individual property level. Such configuration is not generally required. However, it is available for advanced object management tuning or for special situations.

See *Managing Storage and Retrieval of Entity Objects in TIBCO BusinessEvents Architect's Guide* for an overview that is relevant to these settings.

Metadata Properties for Entities (Events and Concepts)

Metadata properties in an entity's Metadata section are used to fine-tune backing store behavior. Additional meta properties that may appear are used with TIBCO BusinessEvents add-ons. Only backing-store metadata properties are documented in this section. In most cases metadata property configuration is not required.

Metadata properties exist both at the entity level, and at the property level.

Table 3 Entity-level metadata properties for events and concepts

Field	Global Var?	Description
Backing Store Properties		
Note These properties are ignored if the backing store feature has not been configured for the application or if it is configured but disabled in the CDD editor. See Cluster Tab — Cache OM — Backing Store Settings on page 38 .		
hasBackingStore		Set this field to false to specify entities that are excluded from the backing store. Possible values are true and false. Default is true.
Type Name		For the Oracle-only (legacy) backing store, you can enter a custom type name, instead of the Oracle type name generated by the backing store scripts. This is useful for giving short or meaningful names to types.

Table 3 Entity-level metadata properties for events and concepts

Field	Global Var?	Description
Table Name		<p>You can enter a custom table name, instead of the name generated by the backing store scripts. This is useful for giving short or meaningful names to types. This setting is used for both the JDBC backing store and the Oracle-only (legacy) backing store. See Set Metadata Properties for Long Identifiers, as Desired on page 242 for more details.</p>
Cache-Related Behavior		
These properties are used only if the application uses a backing store.		
Evict From Cache on Update		<p>Used only if both of the following are the case:</p> <ul style="list-style-type: none"> • Cache-aside strategy for database updates is used • The property <code>Agent.AgentClassName.cacheTxn.updateCache</code> is set to false. <p>If set to true: When a rule action changes the value of any of this entity's properties, then the entity instance is evicted from the cache (updates are saved in the backing store)</p> <p>Use as needed to improve performance and cache memory management. For example, if an entity is not accessed frequently, it may save memory in the cache if the entity is evicted from cache after it is updated.</p> <p>Possible values are true and false.</p> <p>Default is true.</p>

Table 3 Entity-level metadata properties for events and concepts

Field	Global Var?	Description
Constant		<p>This field applies to entities that use the cache-only mode or the cache+memory mode.</p> <p>The processing unit has a special local cache used only for entities marked as Constant. Entities placed in this cache are only removed when they are explicitly deleted. If the processing unit finds an entity in the constant cache, it will use it without checking in the cluster.</p> <p>If set to true The entity is marked "Constant", and uses the constant cache.</p> <p>If set to false (default value) The entity does not use the constant cache.</p>
Check for Version		<p>This field applies to concepts that use cache-only mode or cache+memory mode.</p> <p>The processing unit uses a local cache, of limited size, to improve access time to the concepts stored in the cluster cache. When a processing unit finds a concept instance in this local cache, the Check for Version setting determines whether the processing unit will use the instance directly, or instead check in the cluster cache for more recent version.</p> <p>If set to true (default value) The processing unit will check in the cluster cache for a more recent version. If a more recent version exists, it will be used, and will replace the one found in the local cache</p> <p>If set to false The processing unit will use the instance found locally.</p>

Table 3 Entity-level metadata properties for events and concepts

Field	Global Var?	Description
Is Cache Limited		<p>If a limited cache is used at the global level, you can set this property to false so that this entities instances are all stored in the cache.</p> <p>If a limited cache is not used at the global level, you can set this property to true so that this entity's instances are stored in the cache only to a certain limit, and additional objects are stored in the backing store.</p> <p>There is no entity-level setting for the size of a limited cache. The size is set using this cluster-wide property:</p> <pre>be.engine.limited.cache.back.size.limit</pre> <p>This property is set in the CDD file. See Cluster Tab — Cache OM — Backing Store Settings on page 38 and Cluster Tab — Cache OM — Backing Store Properties on page 42 for more details.</p> <p>Possible values are true and false.</p> <p>Default is true.</p>

Cluster Deployment Descriptor Reference

This chapter provides reference tables for the CDD file settings and related properties.

For configuration procedures, see [Chapter 3, CDD Configuration Procedures](#), on page 9.

Topics

- [Cluster Tab — General Settings](#), page 36
- [Cluster Tab — Cache OM — Configuration Setting](#), page 37
- [Cluster Tab — Cache OM — Backing Store Settings](#), page 38
- [Cluster Tab — Cache OM — Backing Store Properties](#), page 42
- [Cluster Tab — Cache OM — Domain Objects Settings](#), page 46
- [Cluster Tab — Cache OM — Coherence Properties](#), page 50
- [Collections Tab — Destinations Settings and Properties](#), page 56
- [Collections Tab — Log Configurations Overview](#), page 59
- [Collections Tab — Log Configurations Settings](#), page 61
- [Agents Tab Settings and Properties](#), page 64
- [Processing Units Tab — General Settings and Properties](#), page 70
- [Processing Units Tab — JMS Server Reconnection Properties](#), page 75

Cluster Tab — General Settings

Table 4 Cluster Tab —General Settings

Property	Notes
Cluster Name	<p>Specifies the name of the cache cluster.</p> <p>Required.</p> <p>Note Do not use the name <code>\$cluster</code>. It is a reserved name.</p> <p>Defaults to the CDD name.</p>
Message Encoding	<p>The encoding used in Rendezvous messages exchanged between TIBCO applications.</p>
Author	<p>The name of the author of this CDD, as desired.</p> <p>Defaults to currently logged-on user name.</p>
Comment	<p>Any comments as desired. Comments persist across versions.</p>
Version	<p>View-only field to record the version of the CDD, for information only. You could, for example, check whether deployed CDDs are all using the same version.</p>
Date	<p>View-only field to record creation time of this version.</p>

Cluster Tab — Cache OM — Configuration Setting

For General settings see [Cluster Tab — General Settings on page 36](#). For cache OM properties see [Cluster Tab — Cache OM — Coherence Properties on page 50](#).

Table 5 Cluster Tab Reference — Cache OM — Configuration Setting

Property	Notes
Cache Agent Quorum	<p>Specifies a minimum number of storage-enabled nodes (that is, a quorum) that must be active in the cluster when the system starts up before the following occur:</p> <ul style="list-style-type: none"> • Data is preloaded from the backing store, if a backing store is configured and preloading is configured. (See Cluster Tab — Cache OM — Backing Store Settings on page 38 and Cluster Tab — Cache OM — Domain Objects Settings on page 46.) • The other agents in the cluster become fully active. <p>The property does not affect the running of the deployed application after startup (though a message is written to the log file if the number of cache servers running falls below the number specified in this property).</p> <p>As a guideline, set to the number of cache servers configured.</p> <p>Default is 1.</p>

Cluster Tab — Cache OM — Backing Store Settings

Unless otherwise noted, these settings are used both for the legacy (Oracle Types or Oracle-only) backing store and for the JDBC backing store. See [JDBC Backing Store Configuration, page 233](#) and [Appendix A, Setting up an Oracle-Only Backing Store, on page 259](#) for setup details.

Table 6 Cluster Tab Reference — Cache OM — Backing Store Settings (Sheet 1 of 4)

Property	Notes
Enabled	<p>If checked, the backing store configured for this cluster is enabled.</p> <p>A limited cache is used by default when the backing store is enabled. To override this default at the cluster level, add the following property and set its value to false: <code>be.engine.cluster.isCacheLimited</code>. You can also override the global setting at the entity level. See Entity-Level Configuration for Cache and Backing Store on page 30.</p> <p>To set the size of the limited cache, add and configure the property: <code>be.engine.limited.cache.back.size.limit</code></p> <p>Note: Individual entities can be set to not use the backing store. See Metadata Properties for Entities (Events and Concepts) on page 30.</p> <p>If not checked, either the cluster does not have a backing store or the backing store is temporarily disabled.</p> <p>See Cluster Tab — Cache OM — Backing Store Properties, page 42 for more on these properties.</p> <p>Default is unchecked.</p>

Cache Loader Class

Name of the backing store class. Select from the options:

`com.tibco.be.jdbcstore.BECoherenceJdbcStore` — for JDBC backing store.

`com.tibco.be.oracle.BECoherenceOracleStore` — for legacy Oracle backing store.

When JDBC backing store is used, the cache-aside database write method is used by default. See notes for Cache Aside below.

Table 6 Cluster Tab Reference — Cache OM — Backing Store Settings (Sheet 2 of 4)

Property	Notes
Type	<p>Used for JDBC backing store only. Select which of the supported DBMS products to use: <code>oracle</code> or <code>sqlserver</code>.</p> <p>Default is <code>oracle</code>.</p>
Strategy	<p>Used for JDBC backing store only.</p> <p>If you use Oracle Database, you have the option of using either the TIBCO BusinessEvents internal pooling implementation, or Oracle Database's implementation. Possible values are as follows:</p> <p>jdbc Use the internal pooling mechanism.</p> <p>oracle Use Oracle's pooling mechanism (see the class <code>OracleConnectionCacheManager</code> in the package <code>oracle.jdbc.pool</code>). When set to <code>oracle</code> then the TIBCO BusinessEvents pooling property values are used to set their corresponding to Oracle Database properties.</p> <p>Default is <code>oracle</code>.</p>
Cache Aside	<p>Used only if the Backing Store Enabled checkbox is checked. Ignored otherwise (internally set to false).</p> <p>When JDBC backing store is used, cache-aside is required and is used automatically, and this property is ignored.</p> <p>If the legacy Oracle-only backing store is used, you can choose between these two options:</p> <ul style="list-style-type: none"> • Write-behind Writes data to the cache and then to the backing store. One write-behind thread is used for each entity type. See also <code>tangosol.coherence.distributed.threads</code> which is set in the Processing Units tab (see Processing Units Tab — General Settings and Properties on page 70.) • Cache-aside Writes data to the cache and at the same time to the backing store. User controls are available for the threading and queue size. See Post RTC Options — Cache-aside and Write-behind on page 212.

Table 6 Cluster Tab Reference — Cache OM — Backing Store Settings (Sheet 3 of 4)

Property	Notes
Enforce Pools	<p>Check this property if you want to enforce connection pool properties.</p> <p>Default is unchecked.</p>
Backing Store > Connection settings	
<p>Try running with default pool values and monitor the behavior. Using more connections improves runtime performance and can also speed up recovery in the event of a failure.</p> <p>Pool settings are used only if Enforce Pools is checked.</p>	
URI	<p>Specifies the project path, that is, the path from the project root to the JDBC Connection resource, to define the connection to the backing store. For example:</p> <pre data-bbox="297 730 915 756">/SharedResources/JDBC Connection.sharedjdbc</pre>
Min Size	<p>Used by JDBC backing store only if Cluster tab > Backing Store > Strategy is set to <code>oracle</code>. Also used by the Oracle-only legacy backing store.</p> <p>Oracle Database Strategy If the Cluster tab > Backing Store > Strategy field is set to <code>oracle</code>, then Oracle Database strategy settings are used, and this property corresponds to the <code>OracleConnectionCacheManager</code> class property <code>MinLimit</code>.</p> <p>Default is 10.</p>
Max Size	<p>Maximum number of JDBC connections in the JDBC connection pool used for the backing store. Connections do not exceed the maximum.</p> <p>The value of this property overrides the value of the Max Connections setting in the JDBC Connection resource.</p> <p>Oracle Database Strategy If the Cluster tab > Backing Store > Strategy field is set to <code>oracle</code>, then Oracle Database strategy settings are used, and this property corresponds to the <code>OracleConnectionCacheManager</code> class property <code>MaxLimit</code>.</p> <p>Default is 10.</p>

Table 6 Cluster Tab Reference — Cache OM — Backing Store Settings (Sheet 4 of 4)

Property	Notes
Initial Size	<p>Specifies the initial size of the JDBC connection pool used for the backing store, when it is created on startup. For example:</p> <pre data-bbox="342 373 906 407">be.backingstore.dburi.pool.initial.0 10</pre> <p>Oracle Database Strategy If the Cluster tab > Backing Store > Strategy field is set to <code>oracle</code>, then Oracle Database strategy settings are used, and this property corresponds to the <code>OracleConnectionCacheManager</code> class property <code>InitialLimit</code>.</p> <p>Default is 10.</p>

Cluster Tab — Cache OM — Backing Store Properties

Add properties as needed depending on backing store configuration.

Table 7 Cluster Tab — Cache OM — Backing Store Properties

Property	Notes
Database Connection Properties	
	Used only if Enforce Pools (see Cluster Tab — Cache OM — Backing Store Settings on page 38) is checked.
<code>be.backingstore.dburi.pool.waitTimeout.0</code>	<p>Used only if the Strategy setting (see Cluster Tab — Cache OM — Backing Store Settings on page 38) is set to <code>oracle</code>.</p> <p>Oracle Database Strategy If the Cluster tab > Backing Store > Strategy field is set to <code>oracle</code>, then Oracle Database strategy settings are used, and this property corresponds to the <code>OracleConnectionCacheManager</code> class property <code>WaitTimeout</code>.</p> <p>Specifies behavior when a connection is requested and there are already Cluster tab > Connection > Max Size connections active. If the <code>be.backingstore.dburi.pool.waitTimeout.0</code> value is greater than zero (0), each connection request waits for up to the specified number of seconds. If no connection is returned to the pool before the timeout elapses, a <code>No Database Connection available</code> exception is thrown.</p> <p>The <code>waitTimeout</code> and <code>inactivityTimeout</code> properties specify wait periods to minimize the creation and destruction of connections (an expensive operation).</p> <p>Default is 1 second.</p>
<code>be.backingstore.dburi.pool.inactivityTimeout.0</code>	<p>Oracle Database Strategy If the Cluster tab > Backing Store > Strategy field is set to <code>oracle</code>, then Oracle Database strategy settings are used, and this property corresponds to the <code>OracleConnectionCacheManager</code> class property <code>InactivityTimeout</code>.</p> <p>Specifies the number of seconds an unused connection remains available (so that other threads can use it). After this period, the connection is closed and removed from the pool.</p> <p>Default value is 900 seconds.</p>

Table 7 Cluster Tab — Cache OM — Backing Store Properties

Property	Notes
<code>be.backingstore.readtimeout</code>	<p>Oracle Database Strategy If the Cluster tab > Backing Store > Strategy field is set to <code>oracle</code>, then Oracle Database strategy settings are used, and this property corresponds to the <code>OracleConnectionCacheManager</code> class property <code>ReadTimeout</code>.</p> <p>Use this property to handle situations where engines running inference agents hang when the JDBC connection to the backing store is slow or intermittent. If no response is received from the database within the specified period, a call is aborted.</p> <p>Time unit is milliseconds.</p> <p>A value of 0 (zero) means that no timeout is set.</p> <p>Default value is 0</p>
<code>be.backingstore.commitSize</code>	<p>Specifies the number of transactions committed at one time. Set this property to the desired number of transactions to suit your needs.</p> <p>Default is 10.</p>
<code>be.engine.cluster.cleanup</code>	<p>Used by Oracle-Only backing store only.</p> <p>By default, deleted entities are removed from the Oracle-only backing store automatically at system startup. Set this property to <code>false</code> to disable that behavior.</p> <p>Default is <code>true</code>.</p>

Table 7 Cluster Tab — Cache OM — Backing Store Properties

Property	Notes
<code>be.backingstore.dburi.schema.0</code>	<p>Specifies the user whose schema is used for the backing store. This property enables you use to a different schema than the schema of the user specified in the JDBC Connection resource. For example:</p> <pre>be.backingstore.dburi.0 some_user</pre> <p>The JDBC connection resource credentials are always used to connect to the backing store.</p> <p>The user specified in the JDBC connection resource must have CRUD privileges to the schema of the user specified in the <code>be.backingstore.dburi.schema.0</code> property.</p> <p>Default value: If the <code>be.backingstore.dburi.schema.0</code> property is not specified, the schema of the user specified in the JDBC connection resource is used.</p>

Cache Size Properties

`be.engine.cluster.isCacheLimited`

If set to true the cache size is limited. Limited caches can be used only when a backing store is used to store entries in excess of the limit.

This property value is internally set to true if the Cluster tab > Object Management > Backing Store checkbox is checked. You can override the setting using this property, if you want to use an unlimited cache size with a backing store.

You can also override the global setting at the entity level. See [Entity-Level Configuration for Cache and Backing Store on page 30](#).

Ignored otherwise (internally set to false).

Possible values: true or false.

Default is true if backing store is enabled; false if not

Also see notes for `be.engine.limited.cache.back.size.limit`.

Table 7 Cluster Tab — Cache OM — Backing Store Properties

Property	Notes
<code>be.engine.limited.cache.back.size.limit</code>	<p>Specifies the size of the limited cache, in number of cache entries for each object type in each agent where local storage is enabled (cache agents, and other agents where the local storage property is explicitly enabled).</p> <p>Default is 10000 (entries per object type)</p>
<code>be.engine.limited.objectTable.back.size.limit</code>	<p>Specifies the maximum size of the object table cache, in number of entries.</p> <p>Used with limited cache only.</p> <p>See The Role of the Object Table in <i>TIBCO BusinessEvents Architect's Guide</i> for more details about the object table.</p> <p>Default is 100000 entries</p>

Cluster Tab — Cache OM — Domain Objects Settings

Configure object management options at the default (that is global) level, and as needed add entries to configure overrides at the object level.

Two tables are provided, one for the settings at the default level, and one for the similar settings that can be set for individual objects.

All objects use the default level settings unless you add an entry for them at the override level.

Default Level Settings

Table 8 Cluster Tab — Cache OM — Domain Object Default Settings

Property	Notes
Mode	<p>With Cache OM, you can keep memory objects in the cache or Rete network using the following cache modes. The setting you choose here applies to all objects except those for which you explicitly override this value.</p> <p>Memory Only: Objects are not persisted in the cache. They are kept in the Rete network (working memory) only.</p> <p>Cache Only: Objects are persisted in the cache. They must be loaded into working memory as needed. This is the most common choice for a cache cluster.</p> <p>Cache+Memory (also written as Cache Plus Memory): Objects are persisted in the cache. They are also kept current in the Rete network.</p> <p>Note: Due to issues with concurrency, use Cache Plus Memory only for constants and objects that change infrequently.</p> <p>Note If you set the mode to Memory Only, the rest of the properties in this section are not relevant and are ignored.</p> <p>See Chapter 8, Cache Modes and Project Design in <i>TIBCO BusinessEvents Architect's Guide</i> to understand the effect of this setting.</p>

Table 8 Cluster Tab — Cache OM — Domain Object Default Settings

Property	Notes
Preload Entities	<p>Specifies whether objects are loaded into the cache from the backing store on system startup (both normal startup and recovery).</p> <p>Checked: All objects are preloaded into the cache from the backing store. Lower level settings can override this setting by excluding specified objects.</p> <p>Unchecked: No objects are preloaded. into the cache from the backing store. Lower level settings can override this setting by including specified objects.</p> <p>Default is unchecked.</p>
Preload Handles	<p>Specifies whether object handles are loaded into the <code>ObjectTable</code> cache. The <code>ObjectTable</code> cache holds references (handles) to the objects themselves.</p> <p>Handles are used in the object table. See <i>The Role of the Object Table in TIBCO BusinessEvents Architect's Guide</i> for more details.</p> <p>Checked: All object handles are preloaded. Lower level settings can override this setting by excluding handles for specified objects.</p> <p>Unchecked: No object handles are preloaded into the cache from the backing store. Lower level settings can override this setting by including handles for specified objects.</p> <p>Default is unchecked.</p>
Preload Fetch Size	<p>If Preload Entities or Preload Handles or both are checked, this setting specifies the number of entity objects or handles (or both) to preload for each entity type whose objects or handles (or both) are configured to be preloaded.</p> <p>This setting applies to both objects and handles and can't be set differently for each.</p> <p>Objects and handles are fetched in a non-deterministic manner.</p> <p>This setting can be overridden at the entity level.</p> <p>Set to 0 to preload all. Set to a number to load that number of objects or handles (or both).</p> <p>Default is 0. Ignored unless Preload Entities or Preload Handles or both are checked.</p>

Entity Level Object Overrides and Related Settings

Table 9 Cluster Tab — Cache OM — Domain Object Override Settings

Property	Notes
Entity URI	<p>Specifies the project path to the entity for which overrides are being set. Enter manually. For example: /Concepts/MyConcept.</p>
Mode	<p>Overrides the Default level setting. See Mode in Table 8, Cluster Tab — Cache OM — Domain Object Default Settings, on page 46 for details.</p> <p>Note If you set the mode to Memory Only, the rest of the properties in this section are not relevant and are ignored.</p>
Preload Entities	<p>Specifies whether objects of the specified type are loaded into the cache from the backing store on system startup (both normal startup and recovery).</p> <p>Overrides the Preload Entities setting at the Default level. See Table 8, Cluster Tab — Cache OM — Domain Object Default Settings, on page 46.</p> <p>Default: Use the Preload Entities setting specified at the default level.</p> <p>True: Objects of the specified type are preloaded into the cache from the backing store. If the default level setting is not to preload entities, you can use this override to preload selected entities.</p> <p>False: No objects of the specified type are preloaded into the cache from the backing store. If the default level setting is to preload entities, you can use this override to not preload selected entities.</p> <p>Default is "default"</p>

Table 9 Cluster Tab — Cache OM — Domain Object Override Settings

Property	Notes
Preload Handles	<p>Specifies whether object handles for the specified type are loaded into the cache from the backing store on system startup (both normal startup and recovery).</p> <p>Overrides the Preload Handles setting at the Default level. See Table 8, Cluster Tab — Cache OM — Domain Object Default Settings, on page 46.</p> <p>Default: Use the Preload Handles setting specified at the default level.</p> <p>True: Handles for the specified type are preloaded into the cache from the backing store. If the default level setting is not to preload handles, you can use this override to preload selected entities' handles.</p> <p>False: No handles for the specified type are preloaded into the cache from the backing store. If the default level setting is to preload handles, you can use this override to preload selected entities' handles.</p> <p>Default is "default"</p>
Preload Fetch Size	<p>Overrides the Preload Fetch Size setting at the Default level. See Table 8, Cluster Tab — Cache OM — Domain Object Default Settings, on page 46 for details on the general usage of this setting.</p>
Pre-processor	<p>If this object uses Cache+Memory mode, and Subscribe Cluster is checked, specify a subscription preprocessor. This preprocessor is generally used to provide locking to ensure data consistency. See <i>Using Locks to Ensure Data Integrity Within and Across Agents</i> in <i>TIBCO BusinessEvents Architect's Guide</i> for details.</p>
Subscribe Cluster	<p>If this object uses Cache+Memory mode, check this checkbox to subscribe to subscription RTCs, so that changes to this object in one Rete Network are also changed in the Rete networks across all inference agents. See <i>Using Locks to Ensure Data Integrity Within and Across Agents</i> in <i>TIBCO BusinessEvents Architect's Guide</i> for details.</p>

Cluster Tab — Cache OM — Coherence Properties

Add properties as needed to configure multicast cluster member discovery, or well-known address cluster member discovery. Also define the backup count for cluster objects.



If you used the Wizard to set the object management type, a property group called coherence contains all the multicast discovery properties and the backup count property. You can add and delete properties as required for your discovery configuration.

Groups are for convenience only. You can group properties differently or not at all, as you find convenient.

If you are using well-known addresses for cluster member discovery, delete the multicast properties and instead add the well-known address properties.

Table 10 Cluster Tab — Cache OM — Coherence Properties (Sheet 1 of 4)

Property	Notes
Backup Count	
<code>tangosol.coherence.distributed.backupcount</code>	<p>The backup count specifies the number of members of the distributed cache service that hold the backup data for each unit of storage in the cache. Recommended values are 0, 1, or 2.</p> <p>Value of 0 means that in the case of abnormal termination, some portion of the data in the cache will be lost. Value of N means that if up to N cluster nodes terminate at once, the cache data will be preserved.</p> <p>A backup count of 1 means one server plus one backup, that is, two cache servers (or storage enabled nodes if cache servers are not used).</p> <p>To maintain the partitioned cache of size M, the total memory usage in the cluster does not depend on the number of cluster nodes and will be in the order of $M*(N+1)$.</p> <p>Default is 1.</p>

Table 10 Cluster Tab — Cache OM — Coherence Properties (Sheet 2 of 4)

Property	Notes
Multicast Discovery Properties	
If you will define cluster members using multicast discovery properties, use the properties in this section, instead of those in the section Well-Known Address Properties on page 54 .	
<code>tangosol.coherence.clusteraddress</code>	<p>Use this setting if multicast discovery is used and if you need a non-default value. Specifies the multicast IP address that the socket will listen to or publish on.</p> <p>Possible values are addresses between (and including) 224.0.0.0 and 239.255.255.255.</p> <p>Default value is 224.3.3.1</p>
<code>tangosol.coherence.clusterport</code>	<p>Use this setting if multicast discovery is used and if you need a non-default value. Specifies the port that the socket will listen to or publish on.</p> <p>Possible values are integers between 1 and 65535.</p> <p>Default value is 35463</p>
<code>tangosol.coherence.ttl</code>	<p>Specifies the time-to-live setting for the multicast, that is, the maximum number of "hops" a packet can traverse. A hop is defined as a traversal from one network segment to another via a router.</p> <p>For production use, set this value to the lowest integer value that works. Setting the value too high can use unnecessary bandwidth on other LAN segments and can even cause the operating system or network devices to disable multicast traffic.</p> <p>On a single-host cluster, set to zero (0). On a simple switched backbone, set to 1. On an advanced backbone with intelligent switching, it may require a value of 2 or more.</p> <p>Note: A value of 0 is intended to keep packets from leaving the originating machine. However, some operating systems do not implement this correctly, and the packets may in fact be transmitted on the network.</p> <p>Required for multicast configuration.</p> <p>Possible values are integers between 0 and 255.</p> <p>Default value is 4</p>

Table 10 Cluster Tab — Cache OM — Coherence Properties (Sheet 2 of 4)

Property	Notes
Multicast Discovery Properties	
If you will define cluster members using multicast discovery properties, use the properties in this section, instead of those in the section Well-Known Address Properties on page 54 .	
<code>tangosol.coherence.clusteraddress</code>	<p>Use this setting if multicast discovery is used and if you need a non-default value. Specifies the multicast IP address that the socket will listen to or publish on.</p> <p>Possible values are addresses between (and including) 224.0.0.0 and 239.255.255.255.</p> <p>Default value is 224.3.3.1</p>
<code>tangosol.coherence.clusterport</code>	<p>Use this setting if multicast discovery is used and if you need a non-default value. Specifies the port that the socket will listen to or publish on.</p> <p>Possible values are integers between 1 and 65535.</p> <p>Default value is 35463</p>
<code>tangosol.coherence.ttl</code>	<p>Specifies the time-to-live setting for the multicast, that is, the maximum number of "hops" a packet can traverse. A hop is defined as a traversal from one network segment to another via a router.</p> <p>For production use, set this value to the lowest integer value that works. Setting the value too high can use unnecessary bandwidth on other LAN segments and can even cause the operating system or network devices to disable multicast traffic.</p> <p>On a single-host cluster, set to zero (0). On a simple switched backbone, set to 1. On an advanced backbone with intelligent switching, it may require a value of 2 or more.</p> <p>Note: A value of 0 is intended to keep packets from leaving the originating machine. However, some operating systems do not implement this correctly, and the packets may in fact be transmitted on the network.</p> <p>Required for multicast configuration.</p> <p>Possible values are integers between 0 and 255.</p> <p>Default value is 4</p>

Table 10 Cluster Tab — Cache OM — Coherence Properties (Sheet 2 of 4)

Property	Notes
Localhost and Localport Properties	
These properties are used in these cases:	
<ul style="list-style-type: none"> • When a host has multiple network cards. • For multicast discovery when more than one cluster is running on the same subnet (localhost is required but not localport in this case.) • Add these properties at the PU level when well-known address discovery is use as explained in Configuring Well-Known Address Cluster Member Discovery on page 18. Also see Configuring for WKA Cluster Discovery Configuration on page 115. 	
Default values are provided at the cluster level. However if you need specify these properties at the PU level, add them as Processing Units tab properties and provide the values as needed.	
<code>tangosol.coherence.localhost</code>	<p>Specifies the IP address that the socket will listen to or publish on.</p> <p>As needed, you can set the value of the <code>localhost</code> property to the value <code>localhost</code>. However, if <code>localhost</code> is used as the loop back address (127.0.0.1) you must enter a machine name or IP address.</p>
Default value is <code>localhost</code> .	
<code>tangosol.coherence.localport</code>	<p>Specifies the port that the socket will listen to or publish on.</p> <p>Possible values are 1 to 65535.</p> <p>Default value is 8088.</p> <p>Note If a specified port is not available, the object management layer (by default) increments the port number until it finds an available port. Avoid potential conflicts by choosing a number that is not close to a port used by other software in your environment.</p> <p>Tip To turn off auto-incrementing, add this property:</p> <pre>tangosol.coherence.localport.adjust=false</pre>

Table 10 Cluster Tab — Cache OM — Coherence Properties (Sheet 3 of 4)

Property	Notes
Well-Known Address Properties	
See Configuring Well-Known Address Cluster Member Discovery on page 18 .	
Note If you will discover cluster members using well-known addresses, use the properties in this section, and remove the multicast discovery properties shown in the section Multicast Discovery Properties on page 51 .	
Provision is made for up to six well known addresses, using numbers 1–6. If you need more than six, see Overriding and Extending the Operational Deployment Descriptor on page 278 for information about adding more well-known addresses. However this is seldom needed.	
<code>tangosol.coherence.wkan</code>	
<code>tangosol.coherence.wkan.port</code>	
<p>The addresses and ports for machines used by the well-known address cluster discovery protocol. At least one of these machines must be running at any time so that others can join the cluster.</p>	
For <code>tangosol.coherence.wkan</code> , enter the IP address.	
For <code>tangosol.coherence.wkan.port</code> , enter a value between 1 and 65535.	
For example (in the UI the properties are not entered quite this way):	
<code>tangosol.coherence.wka1</code>	<code>10.97.118.151</code>
<code>tangosol.coherence.wka1.port</code>	<code>8088</code>
<code>tangosol.coherence.wka2</code>	<code>10.97.118.152</code>
<code>tangosol.coherence.wka2.port</code>	<code>8088</code>
Also at the Processing Units tab, configure <code>localhost</code> and <code>localport</code> properties for one processing unit that will be deployed to the WKA machine. Set the <code>localhost</code> value to the value of the <code>wkan</code> property. Set the <code>localport</code> value to the value of the <code>wkan.port</code> property. (See Localhost and Localport Properties on page 51)	
Note You can configure two well-known addresses for the same machine, and use a different port number for each. In this case you would also configure two processing units, each of which matches one set of WKA properties.	
Tip To turn off auto-incrementing, add this property:	
<code>tangosol.coherence.localport.adjust=false</code>	

Table 10 Cluster Tab — Cache OM — Coherence Properties (Sheet 4 of 4)

Property	Notes
Other Coherence Properties	
These properties are used in special situations.	
<code>tangosol.coherence.override</code>	<p>Specifies the location of an Operational Descriptor Override File. A sample value is:</p> <pre>file:/c:/tmp/my_tangosol-coherence-override.xml</pre> <p>Use of an operational descriptor override is not generally required. For details see Appendix D, Advanced Caching Topics, on page 277</p>
<code>tangosol.coherence.cacheconfig</code>	<p>If you have customized the cache configuration file, specify the file path and name of the customized file. The default file is located in the <code>BE_HOME/lib/ext/coherence.jar</code> file and is called:</p> <pre>coherence-cache-config-jdbc.xml.</pre> <p>If a backing store is used, see If a Custom Cache Configuration File is Used on page 254.</p>
<code>tangosol.coherence.localport.adjust</code>	<p>An auto-incrementing feature ensures that a different port is used if one specified is already in use. However in various situations you may want to turn off this behavior. For example, if you use TIBCO BusinessEvents Monitoring and Management, and the MM server runs on the same machine as any of the monitored cluster engines, you must explicitly ensure that all ports used by MM <i>and</i> the monitored cluster are unique. Therefore the auto-incrementing feature may not be appropriate.</p> <p>To turn off auto-incrementing, add this property and set the value to false.</p> <p>Default is true.</p>

Collections Tab — Destinations Settings and Properties

Also available from the Agent Classes Tab.

When you select a destination in the CDD editor, you add deploytime settings and properties as needed (shown below). Doing so creates a *destination configuration*. Each destination configuration is assigned a unique ID. You can select and configure a destination in two places in the CDD: in the Collections tab, or in the Agent Classes tab.

Table 11 Destination Settings (Sheet 1 of 3)

Property	Notes
Destination ID	<p>Uniquely identifies this destination configuration at runtime. Edit as needed to ensure that each destination in the cluster has a unique deployment name.</p> <p>Default value is destination name.</p>
URI	<p>Project path to the destination (that is path to the destination in the design-time project).</p>
Preprocessor	<p>Specifying a preprocessor is optional. If you specify a preprocessor, also specify worker thread settings.</p> <p>Select the rule function that has been configured as this event's preprocessor.</p> <p>For more detailed information about preprocessors and see Event Preprocessors in <i>TIBCO BusinessEvents Architect's Guide</i>.</p>

Table 11 Destination Settings (Sheet 2 of 3)

Property	Notes
Threading Model	<p>If you specified a preprocessor, also specify thread settings. Select one model:</p> <p>Shared Queue Uses the TIBCO BusinessEvents system-wide shared queue and threads. See Properties for Shared Queue and Threads Threading Model on page 57.</p> <p>Caller Uses the thread (and queue size) provided by the channel resource client. There is one thread per destination.</p> <p>Workers TIBCO BusinessEvents creates a dedicated thread pool and set of worker threads in each destination. See Thread Count and Queue Size below.</p> <p>For more information on threading models see <i>TIBCO BusinessEvents Architect's Guide</i>.</p>
Thread Count	<p>If you specified Workers in the Threading Model setting, specify the number of threads for this destination here.</p>
Queue Size	<p>If you specified Workers in the Threading Model setting, specify the queue size for this destination here.</p>
<p>Properties for Shared Queue and Threads Threading Model</p> <p>Add to the property sheet as needed.</p>	
<pre>com.tibco.cep.runtime.scheduler.default.numThreads</pre>	<p>Specifies the number of system-wide shared threads.</p> <p>The default value is the same number as there are processors available to the JVM.</p> <p>In MM Console, this thread appears with the name <code>\$default.be.mt\$</code>.</p> <p>See also notes for <code>com.tibco.cep.runtime.scheduler.queueSize</code></p> <p>Default value is 10.</p>

Table 11 Destination Settings (Sheet 3 of 3)

Property	Notes
<code>com.tibco.cep.runtime.scheduler.queueSize</code>	<p>Specifies the queue size for the system-wide shared queue.</p> <p>For more details, see notes for <code>com.tibco.cep.runtime.scheduler.default.numThreads</code></p> <p>If set to 0 (zero), the queue size is unlimited.</p> <p>By default the queue size is the number of threads multiplied by 128.</p>

Collections Tab — Log Configurations Overview

Each processing unit references a log configuration. The log configurations are defined in the Collections tab. See [Configuring the Collections Tab \(All OM Types\) on page 21](#). This section explains more about log configurations.

In a log configuration, you select a *level* of logging for each *module* within TIBCO BusinessEvents runtime software. You can also enable one or more outputs for the log information and configure them.

Levels

A level corresponds to how much logging is filtered out. They are ordered where all is lowest and off is highest:

Level	Description
Off	Highest possible rank. Filters out all logging messages (turns logging off for the specified module).
Fatal	Logs only severe runtime errors that cause the application to stop running.
Error	Also logs runtime errors that might not cause the application to stop running.
Warn	Also logs potentially harmful runtime events or situations.
Info	Also logs runtime informational events of general interest.
Debug	Also logs detailed runtime informational events, for use in identifying issues.
Trace	Also logs even more detailed runtime information.
All	Lowest possible rank. Turns on all logging including any custom logging levels.

Syntax

Enabling a lower level automatically enables the higher levels. For example, enabling `info` automatically enables `fatal`, `error`, and `warning`.

Assign each module to a level using a space-delimited list. The levels are not case sensitive. The syntax is as follows:

```
module1:level module2:level . . .
```

To assign a certain level of logging to *all* modules, use an asterisk:

```
*:info
```

This syntax means that logging for all modules is at the `info` logging level.

You can use the asterisk syntax and also specify exceptions that use a different logging level. For example:

```
*:info driver.tibrv:debug
```

This syntax means that all modules use logging level `info`, except the module `driver.tibrv` which uses `debug` level.

Collections Tab — Log Configurations Settings

Additional log settings can be added to the processing Unit tab Properties. See [Processing Units Tab — General Settings and Properties on page 70](#).

Table 12 Log Configurations Settings (Sheet 1 of 3)

Property	Notes
Name	Name of this log configuration.
Levels	Space-separated list of levels and modules used in this log configuration. See Syntax on page 59 and other sections in introduction to Collections Tab — Log Configurations Overview on page 59 Default is info
Files Section	
Enable	Check the Enable checkbox to enable log files to be written. Configure the settings in this section to specify details. If this checkbox is unchecked, all other properties in this section are ignored.
Directory	Enter the absolute path to the directory in which you want to store the files. If you do not enter a leading slash, the files are stored relative to the working directory (the directory in which you start the <code>be-engine.exe</code> executable).
Name	Name of the log file. The default value is the engine name. If no engine name is set, then the default value is <code>cep-engine.log</code>

Table 12 Log Configurations Settings (Sheet 2 of 3)

Property	Notes
Max number	<p>Number of log files to keep. When the Max size setting value is reached, a new log file is created for the next log entries. Files are created up to the Max number setting size. The oldest file is deleted when a new file is added after this value is reached.</p> <p>Default is 10.</p>
Max size	<p>Maximum size of one log file.</p> <p>Default is 10000000.</p>
Append	<p>If checked then new entries are added to the end of the file. If not checked, the contents of the file are flushed each time the engine starts.</p>
Send to Terminal Section	
Enable	<p>Check the Enable checkbox to enable the redirections specified in this section. If this checkbox is unchecked, all other properties in this section are ignored.</p>
Output redirection	<p>Specify the directory where the STDOUT stream is written.</p>
Error redirection	<p>Specify the directory where the STERR stream is written.</p>
Custom Line Layout Section	
Enable	<p>Check the Enable checkbox to enable the custom line layout entries to take effect. Configure the settings in this section to specify details of a custom layout. If this checkbox is unchecked, all other properties in this section are ignored.</p>

Table 12 Log Configurations Settings (Sheet 3 of 3)

Property	Notes
Class	The custom line layout class.
Arguments	Arguments used for the custom line layout class, for example values for substitution variables.

Agents Tab Settings and Properties

Different sections in the table below deal with different types of agent classes.



See [Using Properties at Different Levels, page 10](#) to understand the effect of using agent class properties at the cluster level and at the processing unit level to widen the scope of the property.

Table 13 Agents Tab Settings and Properties (Sheet 1 of 6)

Property	Notes
Agent Class Name	Enter a name that is unique across the cluster. Use the NSName standard as in all names in the CDD.
Agent Class Type	<p>Valid types for your cluster depend on object management type, and whether you use any TIBCO BusinessEvents add-on products.</p> <p>Inference Used with all OM types. This agent class has additional functionality with Cache OM than other OM types.</p> <p>Cache Used with Cache OM to provide cache storage and other services.</p> <p>Query Used with queries, part of the TIBCO BusinessEvents Event Stream Processing add-on product.</p> <p>Dashboard Used with the TIBCO BusinessEvents Views add-on product.</p> <p>Monitoring & Management Used only by the Monitoring and Management component. See Chapter 6, Basic MM Configuration, page 85.</p>

Table 13 Agents Tab Settings and Properties (Sheet 2 of 6)

Property	Notes
Inference Agent and Query Agent Configuration Settings	
Max Size (Local Cache)	<p>Specifies the maximum number of objects (entities) in each agent's L1Cache (inference agent) or local cache (query agent). The L1 cache is a local cache used by the inference agent for local access to recently used objects. It is used to optimize access to objects.</p> <p>The query local cache is used in a way similar to the inference agent L1Cache. The query agent's local cache stores cache data locally for efficient reuse. The local cache listens to and synchronizes the locally stored entity instances with those in the main cache, so that the local cache stays up-to-date.</p> <p>When the threshold is reached, oldest entities are removed first.</p> <p>Default is 1024 (unit is objects).</p>
Eviction Time (Local Cache)	<pre>be.agent.query.localcache.maxelements be.agent.query.localcache.evictseconds</pre> <p>Specifies an age limit on the cached entities in seconds. After this period, they are removed from the local cache.</p> <p>Note Age resets each time an entity is accessed by a query engine.</p> <p>Default is 900.</p>
BusinessWorks Repo URL (Inference Agents Only)	<p>If this project will integrate with a TIBCO ActiveMatrix BusinessWorks project, enter the Repo URL for the ActiveMatrix BusinessWorks project repo URL here.</p> <p>Use forward slashes.</p> <p>See Task A, Add the Repo URL for BusinessWorks to the CDD in Chapter 21, ActiveMatrix BusinessWorks Integration of <i>TIBCO BusinessEvents Developer's Guide</i> for more details about the value to use.</p>
Concurrent RTC (Inference Agents Only)	<p>If checked, this agent runs concurrent run to completion (RTC) cycles. Use of the concurrency features affects project design. See <i>TIBCO BusinessEvents Architect's Guide</i> for more on this and other concurrency features.</p>

Table 13 Agents Tab Settings and Properties (Sheet 3 of 6)

Property	Notes
<code>tangosol.coherence.distributed.localstorage</code>	<p>Used only for inference agents and query agents. If set to true, the agent is used for storing cache data and in other respects acts like a cache agent, in addition to its other purposes.</p> <p>Note: Set to true use for test deployments only. Not recommended in production.</p> <p>Default value is false (internally set to true for cache agents).</p>
Inference Agent Properties	
<code>Agent.AgentClassName.maxActive</code>	<p>Specifies the maximum number of active instances of the agent. This value is used for fault tolerance. Deployed agents that are acting as standbys can take over from active instances that fail. In many cases, there is no need to keep standby instances.</p> <p>A value of 0 indicates an unlimited number of active instances.</p> <p>See Load Balancing and Fault Tolerance of Inference Agents in <i>TIBCO BusinessEvents Architect's Guide</i> for more details.</p> <p>Default is 0.</p>
<code>Agent.AgentClassName.recoveryPageSize</code>	<p>Specifies the number of entries per page to be used while recovering objects from the cache.</p> <p>For example, if you set the value to 10,000, then the engine loads handles in blocks of 10,000, instead of trying to load them in a single batch. Smaller batch sizes result in slower recovery. Experiment with batch size to establish the best batch size to use for your environment.</p> <p>A value of 0 means that the objects are recovered in one iteration.</p> <p>Default is 0.</p>
<code>Agent.AgentClassName.checkDuplicates</code>	<p>By default, TIBCO BusinessEvents checks if the external IDs (<code>@extId</code>) of entities are unique within the agent. If you want to check for uniqueness of external IDs across the cluster, set this property to true. Performing this check affects performance.</p> <p>Default is false.</p>

Table 13 Agents Tab Settings and Properties (Sheet 4 of 6)

Property	Notes
<code>Agent.AgentClassName.cacheTxn.updateCache</code>	<p>Used only if cache-aside strategy for database updates is used.</p> <p>If set to true: When a rule action changes the value of any entity's properties, then the entity instance is evicted from the cache. (Updates are saved in the backing store.)</p> <p>Note interactions of this CDD property with the metadata property Evict From Cache on Update:</p> <ul style="list-style-type: none"> • When this CDD property is set to true, the metadata property Evict From Cache on Update is ignored. • When this CDD property is set to false, the metadata property Evict From Cache on Update overrides this CDD property. <p>See also Entity-Level Configuration for Cache and Backing Store on page 30.</p> <p>Use these settings as needed to improve performance and cache memory management.</p> <p>Possible values are true and false.</p> <p>Default is true.</p>
Cache Writer Properties for Use with Cache-Aside (and Recovery)	
Used in <code>\$CacheWriter</code> thread tuning	
<code>Agent.AgentClassName.threadcount</code>	<p>Defines the number of threads performing cache writing jobs.</p> <p>Tip This property is also used to define the number of Recovery threads (used for recovering Cache Plus Memory entity handles at inference engine startup).</p> <p>Default value is 2.</p>

Table 13 Agents Tab Settings and Properties (Sheet 5 of 6)

Property	Notes
Database Writer Properties for Use with Cache-aside	
Used in \$DBWriter thread tuning	
<code>Agent.AgentClassName.dbthreadcount</code>	<p>The number of threads performing database writing jobs (applying entity inserts, updates, and deletes to the database).</p> <p>Used only with cache-aside.</p> <p>Default is 2.</p>
<code>Agent.AgentClassName.dbOpsQueueSize</code>	<p>The size of the queue (a Java blocking queue) for database writing jobs.</p> <p>Zero (0) or a negative value means the queue size is unlimited.</p> <p>Used only with cache-aside.</p> <p>Default is 8.</p>
<code>Agent.AgentClassName.dbOpsBatchSize</code>	<p>The number of Rete transactions included in a single database transaction.</p> <p>Used only with cache-aside.</p> <p>Default is 10.</p>
Query Agent Properties	
<code>be.agent.query.localcache.prefetchaggressive</code>	<p>If set to true, then objects required for a query are prefetched while the query is executing.</p> <p>The prefetch feature improves performance, but CPU and memory usage increases as a result of the aggressive prefetching. You may have to try different values till you find the optimal settings for your environment.</p> <p>Ensure that the cache size is large enough to accommodate objects that are prefetched.</p> <p>Default is false.</p>

Table 13 Agents Tab Settings and Properties (Sheet 6 of 6)

Property	Notes
<code>be.network.mode.standalone</code>	<p>Query agents that perform event stream processing only can be deployed in a standalone mode. Such agents do not use cache objects and do not require cache configuration properties.</p> <p>If true, specifies that the PU containing this query agent is a standalone PU and it does not use any cache features.</p> <p>(For more details see <i>TIBCO BusinessEvents Query Developer's Guide</i> available with the TIBCO BusinessEvents Event Stream Processing add on product.)</p> <p>Default is false.</p>
Cache Agent Properties	
<code>be.engine.cacheServer.channel.disable</code>	<p>By default cache servers connect to channels. In most cases, however, cache servers do not need to connect to channels. To prevent cache servers from connecting to channels, set this property to true.</p> <p>Default value is false.</p>
<code>java.property.tangosol.coherence.distributed.threads</code>	<p>Specifies the number of Coherence daemon threads used by the distributed cache service.</p> <p>Used only if write-behind database writes are used (not used with cache-aside).</p>

Processing Units Tab — General Settings and Properties



Many properties can be defined at the cluster level, processing unit level, or agent level, as appropriate (see [Using Properties at Different Levels on page 10](#)).

For example, for well-known address cluster discovery configuration, you would add `tangosol.coherence.localhost` and `tangosol.coherence.localport` properties at the processing unit level (see [Localhost and Localport Properties on page 51](#)).

Table 14 Processing Units Tab General Settings and Properties (Sheet 1 of 5)

Property	Notes
Name	Enter a name that is unique across the cluster.
Log Configuration	Browse to and select a log configuration, configured at Collections tab.
Hot Deploy	Check the checkbox to enable hot deployment for this processing unit. See Chapter 11, Hot Deployment, on page 195 for details.
Enable DB Concepts	Check the checkbox to enable database concepts functionality for this processing unit. Available only with TIBCO BusinessEvents Data Modeling add-on software.
Agents Section	
Agent	Name of the agent class you selected. Agent classes are defined at the Agent Classes tab.

Table 14 Processing Units Tab General Settings and Properties (Sheet 2 of 5)

Property	Notes
Key	<p>Specifies a value that uniquely identifies an instance of an agent of this class at deploy time.</p> <p>Required for recovery of scorecards. Recommended in all cases, for situations that require an agent instance to be uniquely identified.</p> <p>The value for Key must uniquely identify the agent.</p> <p>Note: In certain TIBCO BusinessEvents Monitoring and Management methods, you may be prompted for a session name. For session name you generally put the agent class name. However, if the agent class also has a key, you must instead use the key value. For this reason the key value must uniquely identify the agent.</p> <p>Default is Coherence node UID.</p>
Priority	<p>Specifies the priority of the agent for load balancing purposes.</p> <p>The priority indicates the order in which standby agents become active, and conversely, the order in which active agents become standbys, when new agents join the cluster.</p> <p>The <i>lower</i> the number, the higher the agent is in the activation priority list. For example, an agent with priority 2 has a higher priority than an agent with a priority of 6.</p> <p>Ensure that inference agents of the same class in different PUs have different values. This value determines the order of each instance of an agent class for startup, as well as failover and failback in fault tolerance situations.</p> <p>Default is 10.</p>

Table 14 Processing Units Tab General Settings and Properties (Sheet 3 of 5)

Property	Notes
tangosol.coherence.log properties	
	<p>The properties in this section are used to configure the Coherence (cache) log. This log is used only by cache servers, and only if cache-aside strategy for database updates is used.</p>
	<p>Standard logging settings are configured in the Log Configuration tab. See Collections Tab — Log Configurations Overview on page 59.</p>
<code>tangosol.coherence.log</code>	
	<p>Specifies the output device used by the logging system.</p>
	<p>Optional.</p>
	<p>Possible values are:</p>
	<ul style="list-style-type: none"> • <code>stdout</code> • <code>stderr</code> • <code>jdk</code> (Requires JDK 1.4 or later) • <code>log4j</code> (Requires log4j libraries to be in the classpath) • A file name
	<p>If you specify <code>jdk</code> or <code>log4j</code> you must also perform appropriate configuration of the JDK or Apache log4J logging libraries.</p>
	<p>Default is <code>stdout</code>.</p>

Table 14 Processing Units Tab General Settings and Properties (Sheet 4 of 5)

Property	Notes
<code>tangosol.coherence.log.level</code>	<p>Specifies which logged messages are output to the log destination.</p> <p>Optional.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> • 0: Only output without a logging severity level specified will be logged • 1: All the above plus errors • 2: All the above plus warnings • 3: All the above plus informational messages • 4-9: All the above plus internal debugging messages (the higher the number, the more the messages) • -1: No messages <p>Default is 5.</p>
<code>tangosol.coherence.log.limit</code>	<p>Specifies the maximum number of characters that the logger daemon processes from the message queue before discarding all remaining messages in the queue.</p> <p>The message that causes the total number of characters to exceed the maximum is not truncated.</p> <p>All discarded messages are summarized by the logging system with a single log entry detailing the number of discarded messages and their total size. When the queue empties, the logger is reset and subsequent messages are again logged.</p> <p>The purpose of this setting is to avoid a situation where logging can itself prevent recovery from a failing condition, for example by contributing to timing issues.</p> <p>Logging occurs on a dedicated low-priority thread to further reduce its impact on the critical portions of the system.</p> <p>Optional.</p> <p>Possible values are positive integers or zero (0). Zero implies no limit.</p> <p>Default is 0</p>

Table 14 Processing Units Tab General Settings and Properties (Sheet 5 of 5)

Property	Notes
tangosol.coherence.management properties	
You must also specify the following JMX-related properties to enable cluster statistics to appear in the monitored objects table in MM:	
<code>tangosol.coherence.management</code>	Default is all
<code>tangosol.coherence.management.remote</code>	Default is true

Processing Units Tab — JMS Server Reconnection Properties

The following properties relate to JMS channels. They enable you to define how TIBCO BusinessEvents attempts to reconnect to a JMS server in the event of a disconnection. See *TIBCO BusinessEvents Developer's Guide* for details about configuring a JMS channel.

You can add these properties at the cluster level if they apply to JMS channels in all processing units in the cluster.

Table 15 Properties for Reconnecting to a JMS Server

Property	Notes
<code>com.tibco.tibjms.connect.attempts</code>	<p>Specifies the number of reconnection attempts, and the interval between each attempt to connect to the JMS server.</p> <p>The value must use the format: <i>attempts, retry interval</i>. For example: 10, 500 means 10 attempts, with a 500 millisecond interval between each retry attempt.</p> <p>This property is used only for channels that have a TIBCO Enterprise Message Service provider.</p> <p>Note: Use <i>either</i> <code>be.jms.reconnect.timeout</code> or <code>com.tibco.tibjms.connect.attempts</code>. If you set both the properties, then <code>com.tibco.tibjms.connect.attempts</code> takes precedence.</p> <p>Default is 2, 500</p>
<code>be.jms.reconnect.timeout</code>	<p>Specifies the retry interval (in seconds) for reconnecting to the JMS server when the connection is broken.</p> <p>A value of zero (0) means do not retry. Any other value means keep retrying (with no limit to number of retries), and use the specified interval between each attempt.</p> <p>Note: Unacknowledged messages (Events) are resent to the TIBCO BusinessEvents engine, which may result in duplicate events.</p> <p>Note: Use <i>either</i> <code>be.jms.reconnect.timeout</code> or <code>com.tibco.tibjms.connect.attempts</code>. If you set both the properties, then <code>com.tibco.tibjms.connect.attempts</code> takes precedence.</p> <p>Default is 0 (zero)</p>

Table 15 Properties for Reconnecting to a JMS Server (Cont'd)

Property	Notes
<code>be.jms.reconnect.msgCodes</code>	<p>Specifies a case-insensitive character pattern that matches all error messages or error codes that will cause a reconnect attempt.</p> <p>This property is used for JMS channels with providers other than TIBCO Enterprise Message Service.</p> <p>Default is * (that is, the wildcard matched by any characters.)</p>

This chapter explains how to configure the CDD file for Berkeley DB OM.



Berkeley DB OM is deprecated in this release It is recommended that you migrate from Berkeley DB OM (known as Persistence OM in TIBCO BusinessEvents 3.x) to another object management option. Migration information is provided in *TIBCO BusinessEvents Installation*.

Topics

- [Configuring a Berkeley DB OM Cluster, page 78](#)
- [Cluster Tab — Berkeley DB OM Settings and Properties, page 79](#)

Configuring a Berkeley DB OM Cluster

This procedure assumes you have picked Berkeley DB in the second page of the new cluster configuration wizard (see [Adding a Cluster Deployment Descriptor on page 14](#)).

Configure the Cluster Tab of the CDD Editor

1. In the Cluster tab click **General**. On the right, specify the following:
 - The cluster name and message encoding.
 - As desired, an Author name, any comment you wish to record. (Version and date are not editable.)

See [Cluster Tab — General Settings on page 36](#) for details.

2. **Object Management** should be set to Berkeley DB. If not you can right-click and choose **Change to Berkeley DB**.

See the note in the section [Adding a Cluster Deployment Descriptor on page 14](#) for implications of switching here.

3. In the left panel, click **Object Management: [Berkeley DB]**. Various options appear on the right. Configure them according to guidelines provided in [Cluster Tab — Berkeley DB OM Settings and Properties on page 79](#).
4. In the left panel, click **Properties** and add additional properties (and property groups) as required and provide values. Configure them according to guidelines provided in [Cluster Tab — Berkeley DB OM Settings and Properties on page 79](#).

Property groups are for your convenience only and have no effect on deployment.

5. Save.

To complete configuration refer to the following sections:

- [Configuring the Collections Tab \(All OM Types\) on page 21](#)
- [Configuring the Agent Classes Tab \(All OM Types\) on page 24](#)
- [Configuring the Processing Units Tab \(All OM Types\) on page 27](#)

Cluster Tab — Berkeley DB OM Settings and Properties

For General settings see [Cluster Tab — General Settings on page 36](#).

Table 16 Cluster Tab — Berkeley DB OM Settings and Properties (Sheet 1 of 5)

Property	Notes
Berkeley DB OM Settings	
Checkpoint Interval	<p>A checkpoint is the point in time at which working memory data is written to disk. The checkpoint interval is the time, in seconds, between writes to disk.</p> <p>The term checkpoint also encompasses all the activities involved in writing the data to disk.</p> <p>Note: No changes can occur in the Rete network during a checkpoint.</p> <p>It is recommended that you schedule checkpoints based on both the Checkpoint Interval and Max Outstanding Database Operations (see Checkpoint Interval and Outstanding Database Operations on page 83).</p> <p>If you want to use only the Outstanding Database Operations setting, set Checkpoint Interval to zero (0).</p> <p>Default is 30 seconds.</p>
Checkpoint Ops Limit	<p>Database operations include object creations, updates, and deletions. An outstanding database operation is one that is held in working memory only (it has not yet been written to disk). When the number of outstanding database operations exceeds the Checkpoint Ops Limit value, a checkpoint occurs.</p> <p>It is recommended that you schedule checkpoints based on both the Checkpoint Interval and Checkpoint Ops Limit (see Checkpoint Interval and Outstanding Database Operations on page 83).</p> <p>If you want to use only Checkpoint Interval, set Checkpoint Ops Limit to zero (0).</p> <p>Default is 1000.</p>

Table 16 Cluster Tab —Berkeley DB OM Settings and Properties (Sheet 2 of 5)

Property	Notes
Property Cache Size	<p>Defines the maximum number of concept properties that are kept in JVM memory for this agent.</p> <p>When the persistence layer performs cleanup, the least recently used (LRU) properties are moved to the persistence store, to reduce the number of properties in memory to the specified number.</p> <p>See Caches Used for Persistence-Based Object Management on page 84.</p> <p>Default is 10000.</p>
Delete Retracted Objects from Database	<p>When objects are retracted (deleted) from the working memory, they are marked with a retraction flag.</p> <p>To delete retracted objects from the database, check the Delete Retracted Objects from Database checkbox.</p> <p>To leave retracted objects the database (flagged with the retraction flag), uncheck the Delete Retracted Objects from Database checkbox.</p> <p>It is recommended that you delete retracted objects to avoid accumulating large numbers of retracted objects in the database. However, you may want to keep retracted objects in the database, for example for reporting or data mining purposes.</p> <p>Default is checked.</p>
Do not Recover on Restart	<p>To use the persistence database to recover from unplanned system shutdowns, uncheck this checkbox.</p> <p>To disable recovery features, check this checkbox. In this case, the persistence database is used as virtual memory only.</p> <p>When recovery features are disabled, performance improves because the processing required to support the recovery features is not done.</p> <p> When you check the Do Not Recover on Restart checkbox, data is lost in the event of a system failure.</p> <p>Default is unchecked.</p>

Table 16 Cluster Tab —Berkeley DB OM Settings and Properties (Sheet 3 of 5)

Property	Notes
Database Environment Directory	<p>Persistence files for the agent are stored under the Database Environment directory on the target machine when the agent is deployed. By default (if you do not specify a directory), persistence files are located under the TIBCO BusinessEvents engine's working directory, in directories named <code>db/session_name</code>.</p> <p>To specify a different directory, enter the file path from the deployed engine to the directory.</p> <p>Note: Each agent must have its own database environment directory. If you will deploy more than one agent, define its directory at the agent level, see Defining the Database Directories for Each Inference Agent on page 84 for details.</p> <p>Tip: If you can't determine the location of a deployed application's persistence files, search for their filenames. The persistence file directory contains one file called <code>je.lck</code> and one or more files called <code>00000000.jdb</code>, <code>00000001.jdb</code>, and so on.</p>
Berkeley DB OM Cluster Level Properties — Add as needed	
<code>be.engine.om.berkeleydb.internalcachepersent</code>	<p>Percentage of JVM memory to set aside for use by the persistence layer's internal cache. This memory is set aside when the engine starts up.</p> <p>For projects with multiple agents, you can also set <code>be.engine.om.berkeleydb.cacheweight.agent</code></p> <p>Default is 20.</p>

Table 16 Cluster Tab —Berkeley DB OM Settings and Properties (Sheet 4 of 5)

Property	Notes
<code>be.engine.om.berkeleydb.cacheweight.agent</code>	<p>For projects with multiple agents, you can provide a weight for one or more agents. The weight enables the system to calculate what percentage of the memory set aside using <code>be.engine.om.berkeleydb.internalcacheppercent</code> to allocate to each agent.</p> <p>Default is 1.</p> <p>The formula is as follows:</p> $\text{Session cache percent} = \text{internalcacheppercent} * (\text{cache.weight.agent} / \text{total of all session cacheweight values}).$ <p>This value is an integer. Any fractional part resulting from the formula is truncated.</p> <p>For example, if you want to provide a lot of the allocated memory to a certain agent, you can add an entry for that agent providing a higher weight value, and the rest of the agents will be assigned the default weight.</p>
<code>be.engine.om.eventcache.defaultmaxsize</code>	<p>Defines the maximum number of events that are kept in JVM memory for an agent.</p> <p>Sets the default maximum event cache size. This default is used if <code>be.engine.om.eventcache.maxsize.agent</code> is not specified for an agent.</p> <p>Default is -1 (which means, do not use this setting. See below for default behavior).</p> <p>If you do not set either of the event cache maximum size properties (<code>be.engine.om.eventcache.defaultmaxsize</code> or <code>be.engine.om.eventcache.maxsize.agent</code>) then the value of the Object Management tab setting is used (see Caches Used for Persistence-Based Object Management on page 84).</p> <p>Note, however, that the property cache size applies to the number of concept <i>properties</i>. Events store their properties inside the event. The event cache maximum size settings refer to the entire event, not its individual properties.</p>

Table 16 Cluster Tab —Berkeley DB OM Settings and Properties (Sheet 5 of 5)

Property	Notes
<code>be.engine.om.eventcache.maxsize.agent</code>	<p>Defines the maximum number of events that are kept in JVM memory for the specified agent.</p> <p>The default value is provided by the <code>be.engine.om.eventcache.defaultmaxsize</code> property.</p> <p>When the persistence layer performs cleanup, the least recently used (LRU) events are moved to the persistence store, to reduce the number of events in memory to the specified number.</p> <p>If your system has sufficient memory, you can improve performance by increasing the number of events kept in memory. When determining how many events to keep in memory, consider the size of the events — some may be quite large. Also consider your other requirements for memory.</p>
<code>be.engine.om.berkeleydb.dbenv</code>	<p>At deploy time, directories are created for each agent (using agent names) under the directory you specify.</p> <p>Default is <code>./db</code></p> <p>(The period indicates the working directory.)</p>

Additional Configuration Notes

Some items in the [Table 4, Cluster Tab —General Settings](#) table require additional explanation, provided below.

Checkpoint Interval and Outstanding Database Operations

You can schedule checkpoints based on the Checkpoint Interval only, or on the Max Outstanding Database Operations only, or on both settings. It is recommended that you use both settings. When you do so, data is written to disk as follows:

- When the Checkpoint Interval passes (even if fewer than the Max Outstanding Database Operations have occurred).
- When the Max Outstanding Database Operations value is exceeded within the Checkpoint Interval. TIBCO BusinessEvents then resets the Checkpoint Interval timer.

For example, assume the checkpoint interval is thirty seconds and the number of outstanding database operations is defined as five. Thirty seconds passes with only three outstanding database operations, so TIBCO BusinessEvents performs a checkpoint. Then ten seconds passes and six database operations occur, so again, TIBCO BusinessEvents performs a checkpoint. TIBCO BusinessEvents also resets the checkpoint interval timer.

Caches Used for Persistence-Based Object Management

If your system has sufficient memory, you can improve performance by increasing the number of concept properties kept in memory for each agent. When determining how many concept properties to keep in memory, consider the size of the properties — some may be quite large. Also consider your other requirements for memory

Two caches are used with the Persistence option: a concept property cache and an event cache. The property cache size controls how many concept properties are kept in JVM memory. You define similar settings for the event cache in the CDD file.

Additional memory management settings also are available. They enable you to control the percentage of JVM memory that is set aside for use by the persistence layer's internal cache. See [Cluster Tab — Berkeley DB OM Settings and Properties on page 79](#).

Defining the Database Directories for Each Inference Agent

If your project has multiple agents (or if you will deploy the same application multiple times on the same machine), you must ensure that each agent has its own database environment directory. To do so, add the following property at the agent level:

```
be.engine.om.berkeleydb.dbenv
```

Specify the location of the database directory relative to the deployed agent.

At deploy time, directories are automatically created for each agent under the directory you specify.

Chapter 6

Basic MM Configuration

This chapter explains how to configure the connection between the MM server and the cluster to be monitored and managed, and define some basic MM properties.

For details about configuring the MM metrics and features, see [Chapter 7, MM Metrics and Features Configuration](#), on page 119.

Topics

- [Overview](#), page 86
- [Install and Configure Software for Remote Start and Deployment](#), page 88
- [Install and Configure TIBCO Hawk for Machine Level Metrics](#), page 90
- [Configure JMX Properties in Monitored Engine TRA Files](#), page 92
- [Site Topology Overview](#), page 93
- [Configure the Site Topology in TIBCO BusinessEvents Studio](#), page 97
- [Site Topology Reference](#), page 100
- [Specify the Site Topology File and Basic MM Settings in MM.cdd](#), page 107
- [MM Agent Basic Configuration Reference](#), page 109
- [Configuring for WKA Cluster Discovery Configuration](#), page 115
- [Configure MM Console Properties](#), page 117

Overview

This section explains at a high level how to configure the TIBCO BusinessEvents Monitoring and Management (MM) component for use with a deployed BusinessEvents cluster. Sections follow provide detailed steps.

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



- MM requires Cache based object management.
- All agents and nodes to be monitored must belong to the same BusinessEvents cluster (that is, have the same cluster name). That cluster must use Cache based object management.
- For monitoring of machine-level metrics TIBCO Hawk is required. Other metrics are available without use of TIBCO Hawk.

Before You Begin

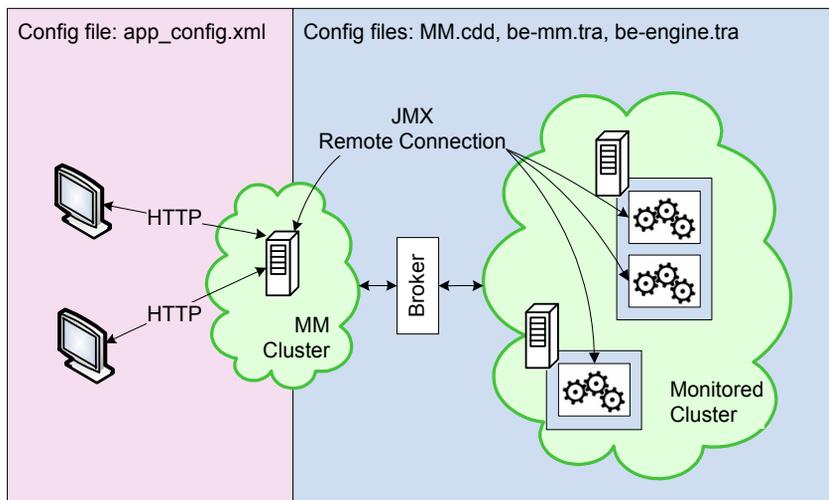
Before you begin, the following monitored cluster project files must be correctly configured and available on the MM server machine.

- The CDD file: See [Chapter 3, CDD Configuration Procedures, page 9](#). Cache OM is required.
- The EAR file: The EAR file contains the project details. See *TIBCO BusinessEvents Developer's Guide*.

MM Runtime Architecture

The following conceptual diagram shows the MM cluster in the center, the web-based MM Console on the left, and the monitored cluster on the right, connected to the MM cluster through JMX, and through a broker.

Figure 1 MM Runtime Architecture

**Legend**

MM Console



Host (for deployed engines)



Engine (processing unit)

JMX remote connections enable MM to use the MBeans exposed in the monitored cluster's engines 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. Configuration is explained in this chapter.

Summary of Configuration Tasks

The configuration tasks are listed below. Specific steps are provided in the referenced procedures.

- [Install and Configure Software for Remote Start and Deployment, page 88](#)
- [Install and Configure TIBCO Hawk for Machine Level Metrics, page 90](#)
- [Configure JMX Properties in Monitored Engine TRA Files, page 92](#)
- [Configure the Site Topology in TIBCO BusinessEvents Studio, page 97](#)
- [Specify the Site Topology File and Basic MM Settings in MM.cdd, page 107](#)
- [Configure MM Console Properties, page 117](#)

Install and Configure Software for Remote Start and Deployment

The following table shows which software utilities can be used to perform remote start and deployment. Information about installation and configuration of each software utility follow. Use the software utility's documentation for more details.

Table 17 Software Options for Deployment, Remote Start, and Remote Method Invocation

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

As you can see, for deployment using MM, SSH is required.

You can use more than one utility for the machines in the cluster. However it is recommended that you 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 on page 106](#).

SSH

Only SSH software enables MM to deploy BusinessEvents software to the predefined hosts, that is, those configured in the monitored cluster's topology file.

SSH can also be used to start remote engines.

The SSH utility is available on UNIX machines by default. No action is required. On Windows machines you must install a server, as explained next.

Installing and Configuring Copssh — Activating a Local User

If you want to use SSH on Windows machines, you must download the software and install it. For machines running Windows OS, TIBCO has tested with Copssh software. Note that Openssh for Windows is not supported.

You must *activate* one of the users available in the host where you install Copssh. To do so, select **Start > All Programs > Copssh > 01 Activate a User**. Select from the list of system users shown, or enter a domain user name.



The credentials of the user you activate are those you specify in the host settings of the site topology file Host Settings User and Password fields. See [Host Settings on page 104](#).



You don't have to add the copSSH bin directory to your system path. However, if you use Copssh, and you also use other Linux bin (executable) tools, add the copSSH bin directory (*yourPath/icw/Bin*) to the system path before all other Linux bin tools.

Restart the Copssh server if you change the path after it is started: In Windows Services, right-click the service **Openssh SSHD** and select restart.

TIBCO Hawk

You can use TIBCO Hawk to start remote engines. It must be installed and running on the MM server machine and on all the client machines that use this method. All machines in the same cluster must use the same TIBCO Hawk domain and use the same transport definitions (server, network, daemon port).

Hawk® is also used for machine-level monitoring. See [Install and Configure TIBCO Hawk for Machine Level Metrics on page 90](#).

The same properties are used for both purposes. Hawk is used for remote start only if specified in the cluster's site topology file. See [Start PU Method Setting on page 106](#).

PsTools

PsTools is an open Windows utility that enables execution 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 you must 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 popup window appears. You must accept the certificate so that PsTools becomes fully functional. Do this once on each host machine.

Install and Configure TIBCO Hawk for Machine Level Metrics

If you want to monitor machine level statistics, you must use TIBCO Hawk software. If you use Hawk, you also need to use TIBCO Rendezvous as the transport.

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

Hawk can also be used for starting remote engines. See [Install and Configure Software for Remote Start and Deployment on page 88](#) for other options. The configuration shown here is used for both purposes.

1. Install TIBCO Hawk and TIBCO Rendezvous software on the MM server machine, and on all client machines in the 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, you must specify them, also in the `mm-class` agent properties list in the `MM.cdd`. 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 in [step 3](#) and [step 4](#) 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 shown in [step 6](#).

Configure JMX Properties in Monitored Engine TRA Files

After the cluster engines are started, they use JMX MBeans to expose monitoring and management information to the MM server, and for 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, and with authentication are not supported.

To Configure JMX Properties

JMX properties are provided in the shipped `BE_HOME/bin/be-engine.tra` file but are commented. Uncomment the `jmxremote` properties shown next. Ensure that the value of the port property is set to `%jmx_port%`.

Do this for all TRA files for all monitored BusinessEvents engines.

```
#java.property.com.sun.management.jmxremote.ssl=false
#java.property.com.sun.management.jmxremote.port=%jmx_port%
#java.property.com.sun.management.jmxremote.authenticate=false
```



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).

How the JMX Remote Port Number is Set at Runtime

When you use the MM UI to start BusinessEvents engines remotely, MM reads the port number from the PU configuration setting in the site topology file. See [\(Processing Unit Settings on page 103.\)](#)

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

Use the same 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 *unpredefined* engine. See [Predefined and Unpredefined Members on page 159](#).

Site Topology Overview

The topology file contains deploytime 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 machines' operating system and IP address.

You also need to know what remote invocation software you'll use to start remote 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.

Editing the Site Topology File in a Text Editor or in the Graphical Editor

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. It is located here:

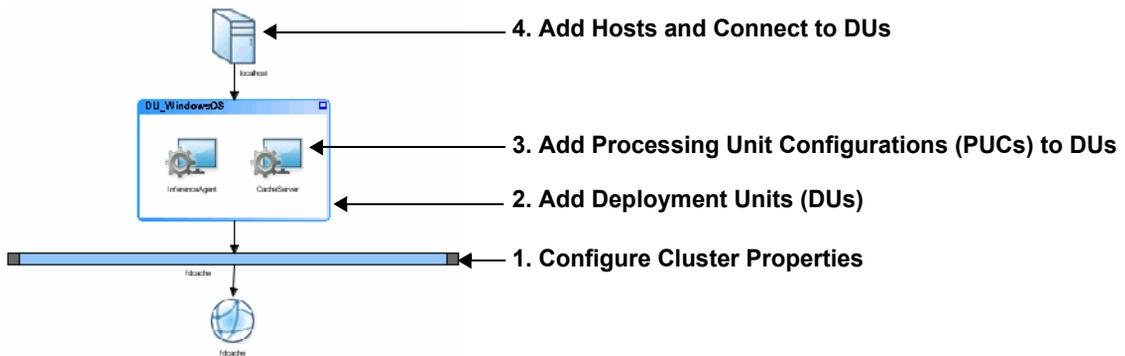
```
BE_HOME/mm/config/site_topology_template.st
```

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

Summary of Site Topology Configuration

Using the canvas-based editor in TIBCO BusinessEvents Studio, you create a visual representation of the desired site topology. Property sheets let you 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. Summary steps are shown below. Detailed steps are provided in [Configure the Site Topology in TIBCO BusinessEvents Studio on page 97](#).



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 on page 95](#) for more on the use of these files.

2. Add Deployment Units (DUs) Add 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.



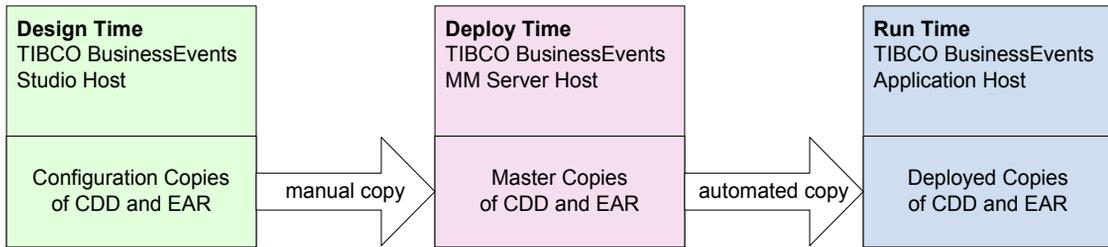
See the note in [Project, Master, and Deployed Locations of CDD and EAR Files on page 95](#) for an important limitation when deploying multiple DUs on one machine.

- One or more *processing unit configurations* (PUCs). You'll 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 deploytime properties such as the JMX ports used by MM to communicate with the deployed engine.

4. Add Hosts Here you specify the machine configuration, including the software used on the remote machines to start remote machines. Connect hosts to the DUs you want to deploy on them. Multiple hosts can use the same deployment unit, as long as the configuration is the same in each case.

Project, Master, and Deployed Locations of CDD and EAR Files



In the topology file, you reference three locations for the CDD and EAR files. The files in each location must be *exact copies*:

- **Project CDD file:** In the cluster configuration tab, you 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:** Also in the cluster configuration tab, you specify the location of the master CDD and EAR files. These copies must be manually copied to the MM server, for use in deployment.
- **Deployed CDD and EAR files:** In the Deployment Unit settings, you 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 are using one machine to configure the topology file and to run MM server. These two sets of fields are available in case you are configuring 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 for the operating system of the host machines.
- **Limitation: One CDD and EAR file per Cluster Machine:** Currently deployment is at the Machine level and each machine can have only one copy of the deployed CDD and EAR files. If you specify multiple DUs for the same host, problems may occur because CDD and EAR files are copied only to the first DU's deployed files location.

Deployment-Specific Processing Units

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.

Note that global variable overrides set in the master CDD are ignored when you deploy using BEMM.

Agent-Instance-Specific Properties

If a processing unit contains agent-instance-specific properties (such as agent key and priority settings), you must 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

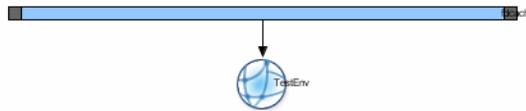
PUs whose `tangoso1.coherence.localhost` property is set to a specific IP value are host-specific. If a deployment unit contains a PUC that references such a PU, you must link it only to the host with that IP address.

Configure the Site Topology in TIBCO BusinessEvents Studio

Before you begin ensure that you have a valid CDD file. The processing units that you will deploy to the various hosts are defined in the CDD. See [Site Topology Overview on page 93](#) for important information. See [Site Topology Reference on page 100](#) for detailed information on the settings.

To Add a Site Topology Diagram

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 are configuring are available for selection.
4. Click **Finish**. You see the site topology editor, showing the cluster bar icon and site globe icon, ready for you to build the site topology diagram:



To Configure the Site Topology

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 selected in the wizard
 - The location of the Master CDD and EAR on the MM server. The MM server 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, you must 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 on page 95](#) 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.

See [Deployment Unit Settings on page 102](#) for details.

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):
 - As desired replace the default PUC name with a name of your choice.
 - As desired, 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 displays. (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, however.

See [Processing Unit Settings on page 103](#) for details, especially on JMX port.

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, IP, and as needed, the user name and password, and operating system.
 - In the Installation tab, ensure the 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.

See [Host Settings on page 104](#) for details.

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 the Delete option.

9. 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 Reference

Site Settings

Table 18 Site Topology — Site Settings

Property	Notes
Site Name	<p>Site name.</p> <p>Default value is the name of the site topology file.</p>
Description	<p>Description of the site, as desired.</p>
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>
Number of Hosts	<p>Read-only field displaying the number of hosts in this topology. (Not present in the site topology XML file.)</p>

Cluster Settings

Table 19 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>

Table 19 Site Topology — Cluster Settings (Cont'd)

Property	Notes
BusinessEvents Version	<p>Read-only field displaying the version of BusinessEvents. Must match the Host Settings field of the same name.</p> <p>Note 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 BusinessEvents Version field in a similar way: both must match.</p>
Project CDD	<p>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 on page 95.</p>
Master CDD	<p>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 on page 95.</p>
Master EAR	<p>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 on page 95.</p>

Deployment Unit Settings

Table 20 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's also a good idea to put the host name in the DU name.</p> <p>Note Paths in different operating systems are specified using different tokens. Even if the DUs are identical in all other respects, you must create different DUs for different operating systems.</p> <p>Default value is DU_# where # 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 used by this DU. See Project, Master, and Deployed Locations of CDD and EAR Files on page 95.</p>
Deployed EAR	<p>Absolute file path to the location where the MM server will deploy the copy of the master EAR used by this DU. See Project, Master, and Deployed Locations of CDD and EAR Files on page 95.</p>
Processing Unit Configurations	<p>Displays a list of processing unit configurations. Adding and configuring PUCs is explained in Configure the Site Topology in TIBCO BusinessEvents Studio on page 97.</p>

Processing Unit Settings

Table 21 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).</p> <p>The processing units settings are configured in the CDD. See Configuring the Processing Units Tab (All OM Types) on page 27.</p>
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>It is recommended that you use the same choice across all processing units in the cluster.</p>
Processing Unit	<p>Select the processing unit 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 How the JMX Remote Port Number is Set at Runtime on page 92 for more details.</p>

Host Settings

Table 22 Site Topology — Host Settings

Property	Notes
General Settings	
Host Name	<p>Machine name of the host. 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: 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 <code>localhost</code>.</p>
IP	<p>IP address of the host machine. Used for remote access. Required.</p> <p>You can use 127.0.0.1 (<code>localhost</code> loop back IP address) for engines running on the same machine as the MM server.</p>
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 runtime, a dialog box pops up to authenticate the user, for example when deploying a PU. If you provide a username and password here, then the dialog is prepopulated with these values. You can enter different values as needed.</p> <p>If you don't 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, you would specify <code>domain\user</code> for domain users.</p> <p>Note If you use <code>copSSH</code>, you must specify the same user credentials here that you use when activating a user for <code>copSSH</code>. See Installing and Configuring Copssh — Activating a Local User on page 88 for details.</p>

Table 22 Site Topology — Host Settings (Cont'd)

Property	Notes
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	
BusinessEvents Version	<p>Read-only field displaying the version of BusinessEvents. Must match the Cluster Settings field of the same name.</p> <p>Note 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 BusinessEvents Version field in a similar way: both must match.</p>
BusinessEvents Home	<p>Install location of TIBCO BusinessEvents on the host machine, for example:</p> <pre>c:/tibco/be/4.0</pre>
TRA File	<p>Location of the be-engine.tra file, for example:</p> <pre>c:/tibco/be/4.0.0/bin/be-engine.tra</pre>

Table 22 Site Topology — Host Settings (Cont'd)

Property	Notes
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 username and password for the remote machines are required for MM to connect (see notes for User Name and Password fields).</p> <p>See Install and Configure Software for Remote Start and Deployment on page 88 for details on each option.</p> <p>Default is SSH. Default SSH port number is 22.</p>

Specify the Site Topology File and Basic MM Settings in MM.cdd

In the CDD file used by MM server, `MM.cdd`, you must specify the location of the topology file for the monitor to be clustered, and perform other basic configuration.

Alerts, health metric rules, and actions are also configured in the `MM.cdd` file. See [Chapter 7, MM Metrics and Features Configuration, on page 119](#) for more details on those tasks.

In order to edit the CDD using the CDD editor, import the eMonitor project into TIBCO BusinessEvents Studio.

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.

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 you must restart the BEMM server so that it uses the updated values.

Import the emonitor Project for CDD Editing

It is not possible to edit this file in TIBCO BusinessEvents Studio outside of its project context. In order to edit the CDD in TIBCO BusinessEvents Studio, you must import the project into your workspace.



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

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, uncheck the Copy the projects into workspace checkbox. (If you do copy the 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.

Specify the Topology File Location and Other Basic Settings

1. Import the emonitor project into your workspace and open the CDD file for editing. See [Import the emonitor Project for CDD Editing on page 107](#) 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 on page 109](#).

You must specify the location of the site topology file configured for the monitored cluster (see [Site Topology Overview on page 93](#)). The monitored cluster's topology file must be located on the MM server. You can copy the file from its design-time location as needed.



If you are running on AIX you must add an additional property to the `mm-agent` class CDD properties. See [For Cache Based Object Management on AIX on page 6](#) for details.

4. Save the CDD file.



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

MM Agent Basic Configuration Reference

See [Specify the Site Topology File and Basic MM Settings in MM.cdd](#) on page 107 for the related procedure.

Table 23 MM CDD mm-class Agent Class Properties

Property	Notes
<code>be.mm.topology.file</code>	<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. See Site Topology Overview on page 93 and sections following for more details.</p> <p>Note The topology file must be located on the same machine as the machine where the MM server is running.</p>
Broker Properties	
<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 you would set this property to true.</p> <p>Default is false</p>
<code>be.metric.cluster.property.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.metric.cluster.property.tangosol.coherence.localhost</code>	<p>IP address of the server machine. Required only if you are using a non-default IP interface.</p>

Table 23 MM CDD mm-class Agent Class Properties (Cont'd)

Property	Notes
<code>be.metric.cluster.property.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>Possible values are 1 to 65535.</p> <p>Default is 9000</p>
<code>be.metric.cluster.property.java.net.preferIPv4Stack</code>	<p>If the server is hosted on an AIX machine, set this property to True. Remember to also add a similar property to the <code>be-mm.tra</code> file. See For Cache Based Object Management on AIX on page 6 for general details (note that the property name is different here than in the TRA file).</p> <p>Default is false</p>
auth section —Authentication Properties	
<p>Authentication is optional. Authentication topics are documented in Chapter 13, Configuring User Authentication, on page 217.</p> <p>To configure for LDAP authentication, add the properties specified in Chapter 13, Configuring User Authentication, on page 217 and remove the file-based authentication properties or remove their values.</p> <p>For file based authentication you can share the same file used by other components that also use file based authentication, as desired and appropriate.</p>	
<code>be.auth.type</code>	<p>Type of authentication used to authenticate users to the MM console.</p> <p>Possible values are File and LDAP.</p> <p>Default value is File</p>
<code>be.auth.file.location</code>	<p>Location of the authentication file used for file-based authentication.</p> <p>Default is <code>BE_HOME/mm/config/users.pwd</code></p>

Table 23 MM CDD mm-class Agent Class Properties (Cont'd)

Property	Notes
<code>java.security.auth.login.config</code>	<p>Location of the JAAS login configuration file. You can substitute a different implementation of the JAAS login module than the one provided.</p> <p>Default is <code>BE_HOME/mm/config/jaas-config.config</code>.</p>
HTTP Properties	
Do not change unless advised by TIBCO.	
<code>be.http.docRoot</code>	<p>The directory from which static HTML content is served.</p> <p>Do not change unless advised by TIBCO.</p> <p>Default is <code>BE_HOME/mm/web-root</code></p>
<code>be.channel.http.propertiesAsXML</code>	<p>Do not change unless advised by TIBCO.</p> <p>Default is true.</p>
Email Properties for Actions Feature	
<code>actions_email</code>	<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 <code>smtp</code> and <code>smtps</code>.</p> <p>Default value is <code>smtp</code></p>
<code>be.mm.email.authentication</code>	<p>Specifies whether the user must authenticate to the email server.</p> <p>Possible values are <code>true</code> and <code>false</code>. If set to <code>true</code> then also configure the <code>username</code> and <code>password</code> properties.</p> <p>Default is <code>false</code>.</p>
<code>be.mm.email.host</code>	<p>Specifies the email host.</p>

Table 23 MM CDD mm-class Agent Class Properties (Cont'd)

Property	Notes
<code>be.mm.email.from</code>	Specifies the From address.
<code>be.mm.email.username</code>	Specifies the username used to authenticate to the email server. Used only if <code>be.mm.email.authentication</code> is set to true.
<code>be.mm.email.password</code>	Specifies the password used to authenticate to the email server. Used only if <code>be.mm.email.authentication</code> is set to true.
Global Variable Overrides — HTTP	
<code>global_variable_overwrite > http</code>	
<code>tibco.clientVar.HTTPHost</code>	Used by the eMonitor project HTTP channel. Overrides the project global variable. Default value is <code>localhost</code>
<code>tibco.clientVar.HTTPPort</code>	Used by the eMonitor project HTTP channel. Overrides the project global variable. Default is 9000
Global Variable Overrides — Frequencies	
<code>global_variable_overwrite > frequencies</code>	
These properties define how frequently certain checks and updates are done. Modify as needed to ensure best performance.	
Overrides global variables in eMonitor project	
<code>tibco.clientVar.TopologyUpdateFreq</code>	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. Default is 30000

Table 23 MM CDD mm-class Agent Class Properties (Cont'd)

Property	Notes
<code>tibco.clientVar.JMXUpdateFreq</code>	<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>
<code>tibco.clientVar.HealthCheckFreq</code>	<p>Specifies the time interval between two consecutive two 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 on page 159 for more details.</p> <p>Default is 30000</p>
<code>tibco.clientVar.SweepFreq</code>	<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 Do not change unless advised by TIBCO.</p> <p>Default is 300000 (that is, five minutes)</p>

Table 23 MM CDD mm-class Agent Class Properties (Cont'd)

Property	Notes
<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: Do not change unless advised by TIBCO.</p> <p>Default is 600000 (that is, ten minutes)</p>
Global Variable Overrides — Hawk	
<code>global_variable_override > hawk</code>	If you use non-default values, specify them here.
<code>tibco.clientVar.Domain=TIBCO Hawk Domain</code>	Name of the TIBCO Hawk domain.
<code>tibco.clientVar.TIBHawkService</code>	Rendezvous service used by Hawk
<code>tibco.clientVar.TIBHawkDaemon</code>	Rendezvous daemon used by Hawk
<code>tibco.clientVar.TIBHawkNetWork</code>	Rendezvous network used by Hawk

Configuring for WKA Cluster Discovery Configuration

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

You must also configure the MM.cdd to work with the cluster to be monitored, but the MM cluster itself does not use well-known address discovery.



Use unique ports — including for MM Server engine If any monitored cluster engines will run on the same machine as the MM server, ensure that all localport values are unique across the MM server and monitored cluster localport values on that machine.

Task A Configure the Monitored Project's CDD

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

For details see [Configuring a Cache OM Cluster — Cluster Tab](#), especially [Configuring Well-Known Address Cluster Member Discovery on page 18](#).

1. Open the project to be monitored in TIBCO BusinessEvents Studio and open its CDD file in the CDD editor.
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.localhost` properties required for basic 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, you can put these properties into a property group, named as desired.

Configure the MM.CDD File

1. Import the `emonitor` project into your workspace and open the CDD file for editing. See [Import the emonitor Project for CDD Editing on page 107](#) for details.
2. In the CDD editor, click the Agent Classes tab and select `mm-class` agent.

3. The broker process must connect to the cluster to be monitored. In the broker properties group, add the following properties:

```
be.metric.cluster.property.tangosol.coherence.localhost
be.metric.cluster.property.tangosol.coherence.localport
```

Set the `localhost` property to the IP address of one of the hosts used by the cluster to be monitored. Set the `localport` property to an available port on that host. Do not use a port that is already in use by the cluster on this host machine.

4. 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
```

As you did with the broker process properties, 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.

Configure the `be-engine.tra` Files for Host Machines with NIC Cards

For all host machines with multiple network cards (NIC), add the following property to the `be-engine.tra` file on that host:

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

The default value is `localhost`.

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

This IP address must match the value specified in other properties where host IP is specified (see [Configure the MM.CDD File on page 115](#):

```
tangosol.coherence.wkan and tangosol.coherence.localhost).
```

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 J-Console.

Configure MM Console Properties

Properties dealing with the Console are configured in the following file:

`BE_HOME/MM/web-root/app_config.xml`

Update the file as needed to set values for the following properties.

Table 24 MM Console Configuration Properties

Property	Notes
<code>debugMode</code>	Set to true to enable more detailed error messages. Default is false.
<code>Demo Mode</code>	If demo mode is enabled, chart updates are faked with random values based on the most recent value. Default is false
<code>updateInterval</code>	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. Default is 5.
<code>failedPaneThreshold</code>	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> . The user clicks OK and is logged out. Default is 0.2.
<code>logoURL</code>	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> . The image displays in the upper left corner. The images size must be no more than 32 by 32 pixels.
<code>chartStyles</code>	You can configure preferences such as colors used for various chart elements. Follow the documentation in the file for each element.

MM Metrics and Features Configuration

This chapter explains how to configure TIBCO BusinessEvents Monitoring and Management alerts, health metrics, and actions.

For basic configuration of the MM component, see [Chapter 6, Basic MM Configuration](#), on page 85.

Topics

- [Overview of Configuring MM Alerts, Thresholds, and Actions](#), page 120
- [Understanding and Configuring Alerts](#), page 121
- [Alert Configuration Reference](#), page 123
- [Specifying the Path to an Alert Metric Value \(and a Reference Value\)](#), page 125
- [Pane Types Reference for Alert Configuration](#), page 128
- [Understanding Health Metric Rules](#), page 130
- [Configuring Health Metric Rules](#), page 135
- [Health Metric Rule Configuration Reference](#), page 138
- [Configuring Actions](#), page 142
- [Action Configuration Reference](#), page 143

Overview of Configuring MM Alerts, Thresholds, and Actions

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:
 - [Understanding and Configuring Alerts on page 121](#)
 - [Configuring Alerts on page 121](#)
 - [Alert Configuration Reference on page 123](#)
 - [Specifying the Path to an Alert Metric Value \(and a Reference Value\) on page 125](#)
 - [Pane Types Reference for Alert Configuration on page 128](#)
- Configure health metric rules:
 - [Understanding Health Metric Rules on page 130](#)
 - [Configuring Health Metric Rules on page 135](#)
 - [Health Metric Rule Configuration Reference on page 138](#)
- Configure actions to take upon triggering of an alert of change in health level:
 - [Configuring Actions on page 142](#)
 - [Action Configuration Reference on page 143](#)

Understanding and Configuring Alerts

The MM server can monitor key performance indicators provided in the metric charts and tables, and generate notifications (alerts) when a specified condition is met.

See [Working with Panels and Panes on page 163](#) for some examples of charts and tables.

Understanding Alerts

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, as desired, 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 on page 165](#) for an example.

The presence of alerts can also be used to define the health level value for any monitored entity. See [Configuring Health Metric Rules on page 135](#).

Configuring Alerts

To Configure an Alert



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.

See [Alert Configuration Reference on page 123](#) 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.

1. Import the emonitor project into your workspace and open the CDD file for editing. See [Import the emonitor Project for CDD Editing on page 107](#) 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)` > **Alert Configurations**.
4. Click **Add** or select an existing alert configuration.
5. Configure the fields as explained in [Alert Configuration Reference on page 123](#)
6. Save.
7. If you have finished configuration, start the emonitor project using the modified CDD file. This starts the MM server. See [Starting MM Server and Logging On to the Console on page 146](#).

Alert Configuration Reference

See [Understanding and Configuring Alerts on page 121](#) for the related procedure.

Table 25 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 on page 130 for details on specifying this value.
Alert Value	<p>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:</p> <p><i>PaneType/SeriesName/CategoryValue/ValueIndex</i></p> <p>See Specifying the Path to an Alert Metric Value (and a Reference Value) on page 125 for details on this setting.</p>
Reference Value	<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 <code>/memory/used</code></p> <p>Then Reference might specify <code>max</code>, which is another series name in the pane type <code>memory</code>.</p>

Table 25 MM CDD mm-class Agent Class Alert Configuration (Cont'd)

Property	Notes
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 <i>Threshold</i> percentage of the value of <i>Reference</i>, the alert is triggered.</p>
Projection Properties	
Severity	<p>The severity of the alert. Possible values are: <code>critical</code>, <code>warning</code>, and <code>normal</code>.</p>
Message	<p>Message to display when this alert is triggered. The message string can optionally contain variables. For example:</p> <pre data-bbox="297 737 1249 789">"{0}'s garbage collection time {2} for {1} has crossed 10% of {3} up time"</pre> <p>See Specifying the Alert Message on page 126 for more details.</p>

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

When you configure an alert, you specify two values to compare, and 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 don't have to provide the full path. The path for reference is structured as follows

SeriesName/CategoryValue/ValueIndex

The parts of the path shown above are explained in [Table 26](#).

Table 26 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. See Table 28, Pane Type Details, on page 128 to find the pane type you want to use.</p>
Series Name	<p><i>PaneType/SeriesName/...</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>

Table 26 Elements Used to Specify the Path to an Alert Metric Value

Path element	Notes
Category Value	<p><i>PaneType/SeriesName/CategoryValue/...</i></p> <p>To set an alert on a specific category of information in a chart or table, specify its name in the <code>CategoryValue</code> position of the path.</p> <p>In a chart, 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, each row represents a category of information. The value in the first (that is, 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. (See Table 28, Pane Type Details, on page 128 for a listing of which panes are time based.)</p> <p>To use all category values, you can use an asterisk (*) as a wild card character in the <code>CategoryValue</code> 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). (See Table 28, Pane Type Details, on page 128 for a listing of which panes use charts and which use tables.)</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). The 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).

Table 27 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> . See Table 26, Elements Used to Specify the Path to an Alert Metric Value, on page 125 .
{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

The following table shows the names of the pane types at each level of the cluster hierarchy (that is, the element types).

Table 28 Pane Type Details

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

Table 28 Pane Type Details (Cont'd)

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

Understanding 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 a information about the overall health of the cluster, and of each of the cluster members. See [Cluster Overview on page 165](#) for an example.

Health is defined in terms of the following three health 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 don't 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

You define the scope of a cluster a metric using a cluster path, for example:

```
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 cluster node: site, cluster, machine, process. Below the process level, you can specify types of agent:

Path specifying a type of cluster member. 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
```

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, you can 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

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.

Which Type to Use for Different Cluster Members

In general these are the guidelines for use of these methods:

- 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.

Using 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.

Using 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

Below are some examples to help you think about the way you want to configure your health metric rules.

Examples Using Child Cluster Member Health Metrics

These examples show how rules can be configured to display a health level indicator on a cluster member based on the health levels its child members.

Note These rules could 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 are inactive

- 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

- 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

- 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

- 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

These examples show how 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

- 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 5 or more critical alerts are received within a window of 5 minutes, for a query agent

- 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`

Configuring Health Metric Rules

Health metric rules have two levels. First you specify the cluster member for which a set of rules will apply. Then you 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.

See [Understanding Health Metric Rules on page 130](#) for more details.

To Set 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 are configuring all three levels, you would put `critical` first, then `warning`, and finally `normal`.

See [Health Metric Rule Configuration Reference on page 138](#) 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.

1. Import the `emonitor` project into your workspace and open the CDD file for editing. See [Import the emonitor Project for CDD Editing on page 107](#) 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**. You see a configuration panel.
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**. You see fields to define a health metric rule for this cluster member.
8. You can configure health metric rules in two ways. See [Two Types of Thresholds on page 131](#) 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:
 - [To Configure a Health Metric Rule Using Child Member Health Status on page 136](#)
 - [To Configure a Health Metric Using Cluster Member Alerts on page 137](#)

To Configure a Health Metric Rule Using Child Member Health Status

This procedure flows from the procedure [To Set Up a Health Metric Rule on page 135](#). See [Health Metric Rule Configuration Reference on page 138](#) for more details on the properties referenced in this 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 **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), as desired.
8. Save.

9. If you have finished configuration, start the emonitor project using the modified CDD file. This starts the MM server. See [Starting MM Server and Logging On to the Console on page 146](#).

To Configure a Health Metric Using Cluster Member Alerts

This procedure flows from the procedure [To Set Up a Health Metric Rule on page 135](#). See [Health Metric Rule Configuration Reference on page 138](#) for more details on the properties referenced in this 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. Save
8. If you have finished configuration, start the emonitor project using the modified CDD file. This starts the MM server. See [Starting MM Server and Logging On to the Console on page 146](#).

Health Metric Rule Configuration Reference

See [Configuring Health Metric Rules on page 135](#) for the related procedure.

Table 29 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	<p>Path that defines the scope of this metric. Scope is defined in terms of cluster levels. See Cluster Member Paths on page 130 for details on specifying this value.</p> <p>Specify a cluster member only once, then specify all the health metric rules for that member in one set.</p>
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	<p>From the drop-down list, select one of the following health levels: critical, warning, or normal.</p> <p>For each cluster member, you can define up to three rules, one for each health level.</p>

Table 29 MM CDD mm-class Agent Class Health Metric Rule Configuration (Cont'd)

Property	Notes
Condition Type	<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 on page 131 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 on page 130 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. <p>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 active property) and optionally: that are at the specified health level.</p>

Table 29 MM CDD mm-class Agent Class Health Metric Rule Configuration (Cont'd)

Property	Notes
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	<p>When used for a Child Cluster Members Condition Type 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 Defines a number of alerts. When the number of alerts notification/property@severity alerts for the enclosing cluster member 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. Note 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>
<h3>Health Metric Rule Configuration Properties</h3>	
<p>The active and health level properties are used only when computing the health level using child cluster members. You can use both active and healthLevel properties in one rule, although there may be few use cases for using both properties.</p>	
active	<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>
healthLevel	<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>

Table 29 MM CDD mm-class Agent Class Health Metric Rule Configuration (Cont'd)

Property	Notes
severity	Used only when the Condition Type is set to Notification . Set to the alert severity that you want to use for the health metric rule calculation.

Configuring Actions

The MM server can perform actions when alerts are triggered or when health level indicators change. An action can be execution of a command. An action can also be sending an email. You must configure email settings in order to use the email feature (see [Specify the Site Topology File and Basic MM Settings in MM.cdd on page 107](#) for details).

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

To Configure an Action

See [Action Configuration Reference on page 143](#) 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.

1. Import the emonitor project into your workspace and open the CDD file for editing. See [Import the emonitor Project for CDD Editing on page 107](#) 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 on page 143](#)
6. Save.
7. If you have finished configuration, start the emonitor project using the modified CDD file. This starts the MM server. See [Starting MM Server and Logging On to the Console on page 146](#).

Action Configuration Reference

See [Configuring Actions on page 142](#) for the related procedure.

Table 30 MM CDD mm-class Agent Class Action Configuration

Property	Notes
Action ID	An ID for this action.
Trigger Condition	
Trigger Condition	<p>Select the type of condition that triggers this action:</p> <p>Health Level A specified health level of the specified cluster members.</p> <p>Alert An alert of a specified severity raised on any of the specified cluster members.</p>
Path	Enter a cluster path. This path defines the cluster members whose health level or alerts trigger this action. See Cluster Member Paths on page 130 for details on specifying this value.
Severity or Health Level	<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>

Table 30 MM CDD mm-class Agent Class Action Configuration (Cont'd)

Property	Notes
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 Specify the Site Topology File and Basic MM Settings in MM.cdd on page 107.</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>

Deploying and Managing Engines with MM

After you have configured your project for deployment and configured MM to connect to the deployment, you are ready to deploy the project, and manage what engines are running in the monitored project.

Topics

- [Starting MM Server and Logging On to the Console, page 146](#)
- [Overriding Global Variables at Deploy Time, page 148](#)
- [Deploying and Managing Engines in the MM UI, page 149](#)
- [Deploying or Starting Engines with the Command Line Utility, page 151](#)

Starting MM Server and Logging On to the Console

After you have completed all the configuration steps explained in [Chapter 6, Basic MM Configuration, on page 85](#), you are ready to use MM.

To Start MM Server



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.

At a command prompt, navigate to `BE_HOME/mm/bin`, and type:

```
be-mm.exe -c MM.cdd -u default -n mm MM.ear
```

Type `be-mm.exe /help` to view usage information.

Or on Windows you can select the following:

Start > All Programs > TIBCO > *TIBCO Env* > TIBCO BusinessEvents 4.0 > Start Monitoring and Management Server.

To Log On to MM Console

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

1. In a web browser, enter the URL for the console. By default the URL is:

```
http://localhost:9000/index.html
```

The hostname and port are configured in the `MM.cdd`. See [Global Variable Overrides — HTTP on page 112](#) for details.

2. Log in using the user credentials that were configured in the authentication file or other authentication mechanism you configured for TIBCO BusinessEvents. As shipped, the default credentials are `admin/admin`.

See [Chapter 13, Configuring User Authentication, page 217](#)

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

See [Cluster Explorer on page 158](#) 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. [OK]

See [Configure MM Console Properties on page 117](#) 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.

Overriding Global Variables at Deploy Time

Global variables are defined in the TIBCO BusinessEvents Studio project. If they are defined as deployment settable and service settable, their default values can also be overridden at deploy time. You can override the default values at the machine level. These global variable overrides are appended to the CDD file that is deployed to a machine.



Notes and Limitations in This Release

- One CDD file is used by all deployed engines on a machine. Therefore global variable overrides are used by all engines deployed on that machine.
- Global variable overrides in the master CDD (that is, the design-time CDD) are not visible in MM Console. Only the global variables set in the TIBCO BusinessEvents Studio Global Variable editor are visible in MM Console and can be overridden there.
- 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 the Refresh button (or starts a new console session).
- While the MM server is running, the current override values are visible in the Global Variables editor. When the MM server shuts down and is restarted, only the default values set in the TIBCO BusinessEvents Studio Global Variables editor are shown.

To Override Global Variable Values at Deploy Time

The overrides affect all engines deployed to the selected machine.

1. Log on to MM Console. See [To Log On to MM Console, page 146](#).
2. In Cluster Explorer, select a machine and select the **Global Variables** tab in the panel on the right. The global variable names and default values display.
3. In the Current Value column, replace the current value with the desired override value.
4. Click **Save**.

MM appends the global variables and values to the deployed CDD file on the target machine.

5. It's a good idea to click **Refresh** to ensure that your value was the last entered. If another user enters an override just after you do, their value wins.

Deploying and Managing Engines in the MM UI

Manually Copy Custom Function and Third-Party Jars at Deploy-time

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, then update the classpath in the TRA file to include the location.

Deploying Cluster Engines

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

1. Log on to MM Console. See [To Log On to MM Console, page 146](#).
2. From the Cluster Explorer, select the machine node you want to deploy.
3. If you want to override any global variables, select the Global Variables tab and do so. See [Overriding Global Variables at Deploy Time on page 148](#) for details.
4. Select the icon of the host machine where you want to deploy and click the Deploy button.

The deployment unit that you configured to deploy on that machine in the site topology file deploys.

5. Verify the login details or provide them (see [Host Settings on page 104](#) 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 Hover the pointer over its name in the explorer panel. A tooltip shows if it is deployed or undeployed.

Starting, Stopping, Pausing, and Resuming Cluster Engines

To Start, Stop, Pause, or Resume an Engine

1. From the Cluster Explorer, select the engine you want to start, stop, pause, or resume. (You resume a paused engine.) Note that you can't start an unpredefined engine.
2. Click the appropriate icon: **Start, Stop, Pause, or Resume**

3. Verify the login details and click **OK**.

Hot Deployment

See [Chapter 11, Hot Deployment, on page 195](#) for details on hot deployment. You can hot deploy to a running engine deployed by MM if the following are true:

- You have modified the TIBCO BusinessEvents Studio project and built the EAR file, following the limitations shown in [Modifications Allowed in Hot Deployment on page 197](#).
- The deployed processing units that you want to hot deploy to were enabled for hot deployment before they were deployed (see [Enabling Hot Deployment on page 199](#))
- 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.

Deploying or Starting Engines with the Command Line Utility

This section explains how to deploy engines and start remote engines at the command line using the `mm-tools` utility.

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

Before you can use the `mm-tools` utility you must configure it.

Configuring the MM Tools Utility TRA File

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

The following setup is prerequisite to use of MM tools:

- 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 [Install and Configure Software for Remote Start and Deployment on page 88](#) for details.
- The site topology file (and its prerequisites) must also be correctly configured. See [Site Topology Overview on page 93](#) and sections following for details.

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

To Configure the `mm-tools.tra` File

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 pathToTopology/toplogy.st
```

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 Hawk is also used for machine level metrics, the values specified would be the same as those specified in the MM.cdd See [Install and Configure TIBCO Hawk for Machine Level Metrics on page 90](#).

4. Save the file.

Deploying or Starting a Remote Engine

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 -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
```

mm-tools Options Reference

Table 31 *mm-tools Options*

Option	Description
-help	Displays this help.

Table 31 *mm-tools Options (Cont'd)*

Option	Description
--propFile	<p>When you execute <code>mm-tools</code>, by default it looks in the working directory (the directory where you execute the command) for a property file of the same name (<code>mm-tools.tra</code>). 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>
-op	<p>Specifies the operation. Two operations are available:</p> <p>deploy: The deploy operation is done through SSH</p> <p>start: The start operation is done using the mechanism defined in the site topology file <code>start-pu-method</code> setting.</p>
-m	<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>
-puc	<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>
-user	<p>Optional. If not specified, the username in the site topology file is used.</p> <p>For the <code>deploy</code> option: The username used by SSH on the remote machine.</p> <p>For the <code>start</code> option: The user name used to log into the remote machine for remote start.</p>
-pwd	<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 username used by SSH on the remote machine.</p> <p>For the <code>start</code> option: The password for the username used to log into the remote machine for remote start.</p>

Monitoring and Managing a TIBCO BusinessEvents Cluster

This chapter explains how you can monitor the health of a TIBCO BusinessEvents deployment using the TIBCO BusinessEvents Monitoring and Management (MM) component and manage the deployment accordingly.

Topics

- [Overview of Monitoring a TIBCO BusinessEvents Cluster, page 156](#)
- [Cluster Explorer, page 158](#)
- [Working with Cluster Explorer, page 161](#)
- [Working with Panels and Panes, page 163](#)
- [Cluster Overview, page 165](#)
- [Machine Overview, page 167](#)
- [Process Overview, page 168](#)
- [Agent Overview, page 170](#)
- [Inference Agent Overview, page 173](#)
- [Query Agent Overview, page 174](#)
- [Ontology \(Cache Objects\) Overview, page 175](#)
- [MM Method Reference, page 177](#)

Overview of Monitoring a TIBCO BusinessEvents Cluster



TIBCO BusinessEvents Views Dashboard agents 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.

See also

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

- For information on configuring MM see [Chapter 6, Basic MM Configuration, on page 85](#).
- For information on configuring the metrics used by the charts and tables, see [Chapter 7, MM Metrics and Features Configuration, on page 119](#)
- For information on starting MM and logging on to the MM Console, see [Starting MM Server and Logging On to the Console on page 146 in Chapter 8, Deploying and Managing Engines with MM, on page 145](#)

Reference to the Metrics

For each level of the cluster hierarchy, various metrics are provided using charts and tables. Later sections 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.

Note the following:

Charting Period 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).

Updates Monitoring and Management polls for updates and refreshes the display periodically. The interval is configurable. See the `updateInterval` property in [Specify the Site Topology File and Basic MM Settings in MM.cdd on page 107](#).

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.

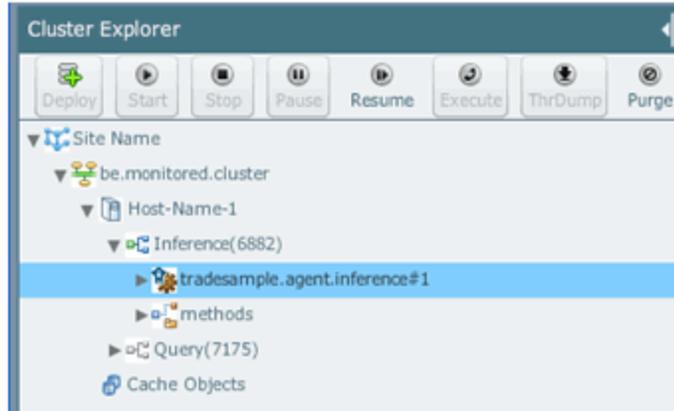
You can also configure alerts to bring specific problem situations to the attention of system users.

These health indicators and alerts are shown in the Cluster Overview, providing a dashboard where you can read the health of the entire cluster at a glance.

See [Chapter 7, MM Metrics and Features Configuration, on page 119](#) for configuration details.

Cluster Explorer

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



The Cluster Explorer figure above 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, without regard to their physical location in 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 can't 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.

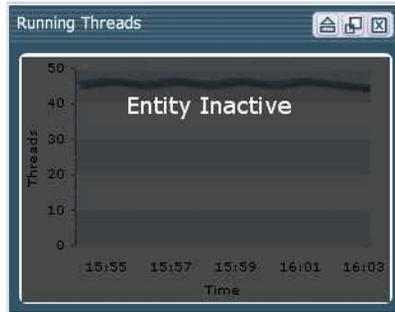
How Inactive Members Display

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:



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 on page 109](#) for details.



You can also configure health thresholds and alerts at any level of the cluster hierarchy. See [Chapter 7, MM Metrics and Features Configuration, on page 119](#) for all configuration options

Working with Cluster Explorer

You can do any of the following:

- Expand Cluster Explorer and select the member 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.

To Purge Inactive Unpredefined Processes

To purge, click the **Purge** icon. The Cluster Explorer view is cleaned. For configuration related to the purge feature see [Chapter 7, MM Metrics and Features Configuration, on page 119](#).

To View Monitored Objects

To view the Monitored Objects details, select the Monitored Objects node from the Cluster Explorer. The Cache Overview page is displayed in the right pane.

The details of various processes are displayed. For example, name of the entity, count, gets, puts, and so on.

To Execute a Method

For a table listing all the available methods, see [MM Method Reference on page 177](#).

1. From the Cluster Explorer, select the method from the running instance of the engine that you want to execute.
2. Click the **Execute** icon.
3. Specify the argument or arguments and click **Execute**.

Generating 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 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.

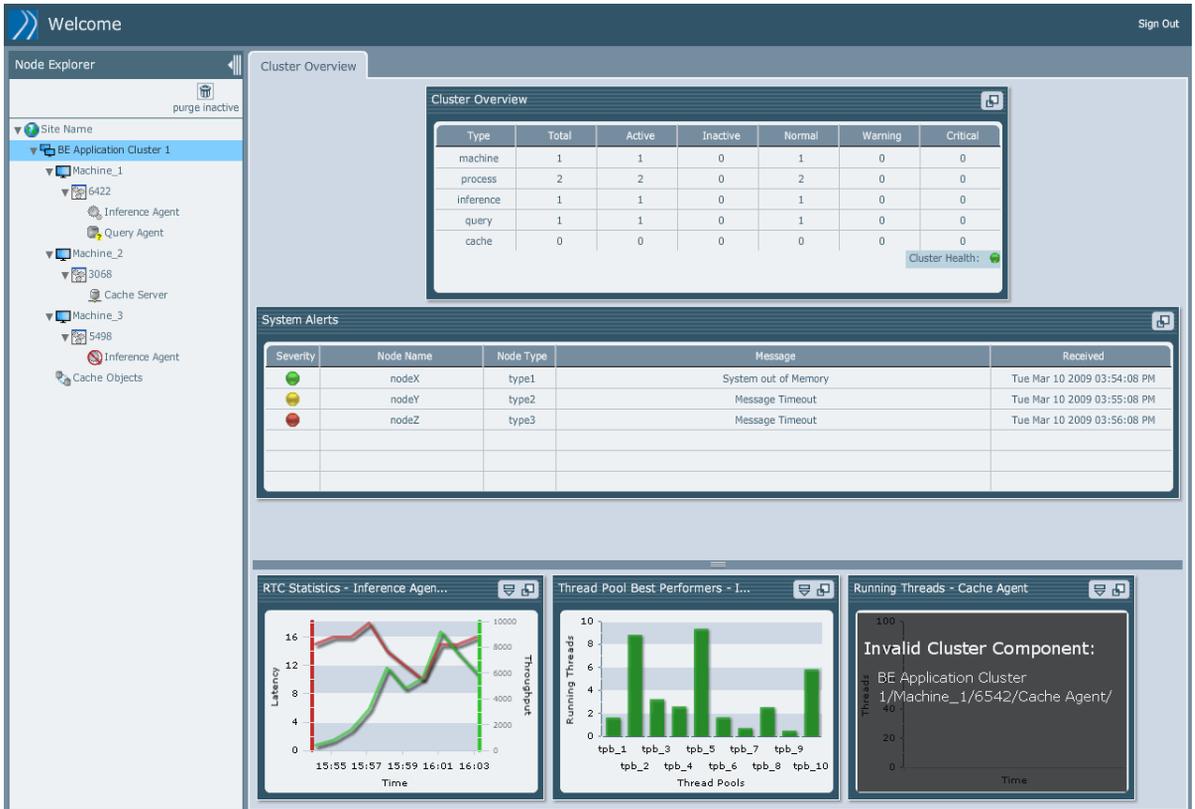
The *x* element is used if you stop and start again. It is a number used to distinguish each set of reports, when multiple sets of reports are generated for the same host.

To Generate Thread Analyzer Reports

1. From the Cluster Explorer, select the process whose threads you want to analyze.
2. Click the **ThrDump** icon.
3. Enter the details as shown below:
 - **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.
 - **Username and password** Enter the credentials (if any are required) used to connect to the JMX server running on the target machine. Note that these are not the JMX credentials and not those used to log into 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: `BE_HOME/mm/logs/thread-analyzer/`.
 - **Time Interval:** The interval in seconds between thread dumps. For example, if you enter 10, a thread dump is obtained every 10 seconds.
4. Click **OK**.

Working with Panels and Panes

The figure below shows the Cluster Overview panel (also called a tab) with three panes promoted from a lower level display (as explained below). One of the promoted panes indicates an agent is inactive.



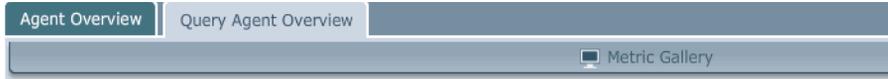
Working with MM Metric Panes

When you navigate to different levels of the cluster hierarchy using Cluster Explorer, appropriate sets of panes display. To learn more about the metrics in the panes, see [Reference to the Metrics on page 156](#).

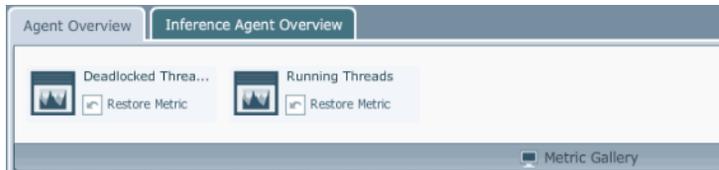
This section explains how to use the controls in a pane's button bar. This information applies to panes in general, not just those that display in the cluster overview.

- 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:



- To restore a removed pane, click the Metric Gallery bar in the appropriate tab and click **Restore Metric**. (Not available on the cluster overview tab.)



Cluster Overview

Cluster Overview

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: 

System Alerts

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

Cluster Overview Pane

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

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 [Table 32, Cluster-Level Metrics](#).

Thresholds for normal, warning, and critical health metrics are configurable. See [Configuring Health Metric Rules on page 135](#).

Table 32 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.

Table 32 Cluster-Level Metrics (Cont'd)

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

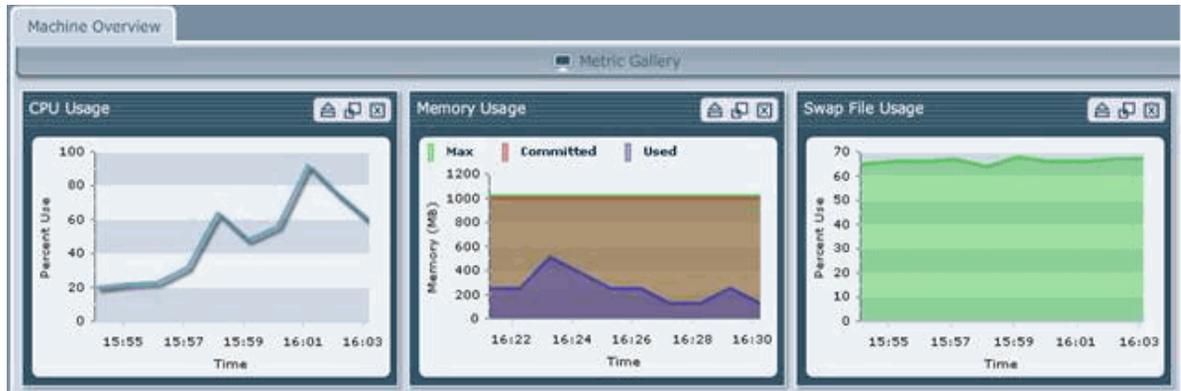
Member Type displays the type of element, for example, inference agent, process, machine, and so on.

You define and configure the alerts you are interested in. For each alert, you specify the metric value of interest, the threshold that triggers the alert, the severity level and a message. See [Understanding and Configuring Alerts on page 121](#).

Machine Overview



Machine level metrics are visible only if a TIBCO Hawk domain has been configured. See ([Install and Configure TIBCO Hawk for Machine Level Metrics on page 90](#)).

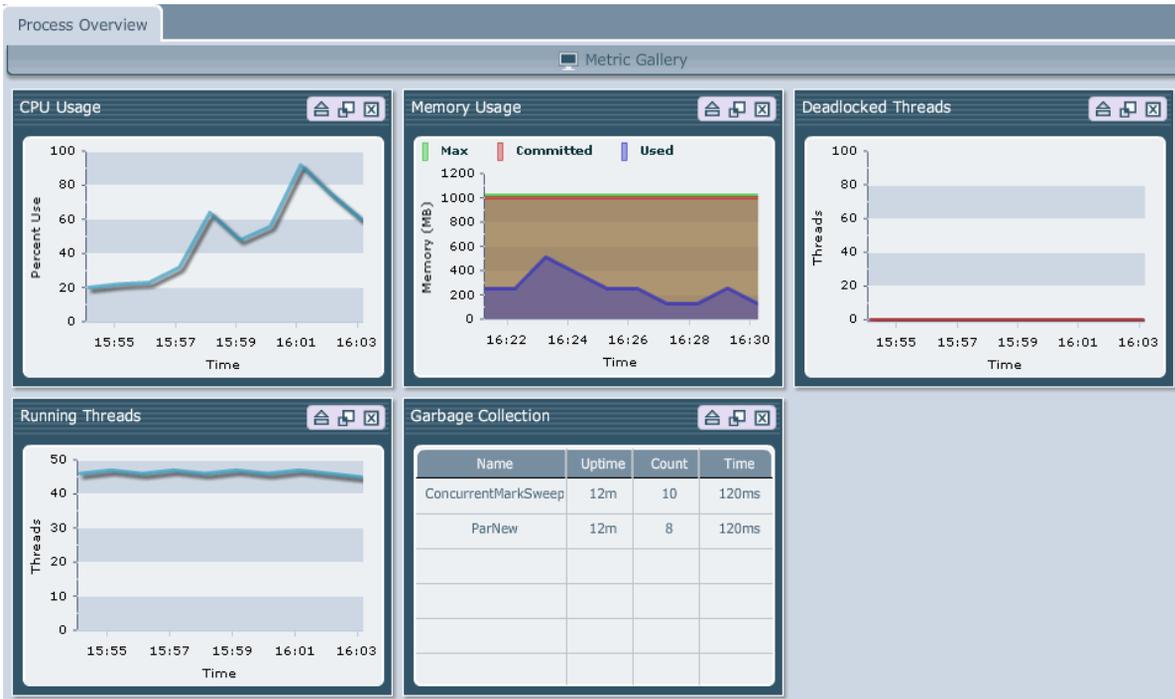


The Metric Gallery (shown as a gray bar above the panes) holds any panes you remove from the panel to keep your display uncluttered. See [Working with Panels and Panes on page 163](#) for details. Machine level metrics shown are as follows:

Table 33 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



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. Metrics shown are as follows:

Table 34 Process- Level Metrics

Metric	Notes
CPU Usage	Percentage of CPU used by this process.

Table 34 Process- Level Metrics (Cont'd)

Metric	Notes
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



Note that the Running Threads, Deadlocked Threads, and Garbage Collection are process-level metrics, shown here to provide a convenient overall view.

The panel common to all agent types shows the agent overview:

Table 35 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.
Garbage Collection	For each garbage collector running in the process, metrics shown are: <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	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. (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.)
Thread Pool Worst Performers	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. (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.)
Thread Pool Usage	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><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 <code>Scheduler</code> created with a single thread uses this default thread pool.</p> <p><code>CommonWorkManager</code>: Any <code>WorkManager</code> created with a single thread uses this default thread pool.</p>

Table 35 Agent Overview (Common) Metrics (Cont'd)

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

Inference Agent Overview



Shown for inference agents only, the Inference Agent Overview panel shows the following metrics:

Table 36 Inference Agent Metrics

Metric	Notes
Locks Held	The number of locks held by the agent.
RTC Statistics	RTC performance in terms of latency and throughput: <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.
Best Rule Performers	Rules sorted by average execution time. The shorter the execution time, the better the rule performance.

Query Agent Overview



Shown for query agents only, the Query Agent Overview panel shows the following metrics:

Table 37 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.
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

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.ConceptMi	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.ConceptMi	1769	0	1769	0	0	0	2147483647	1610612735	0	
be.gen.ConceptMi	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.

The following attributes are shown for each object:

Table 38 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.

Table 38 Cache Objects Metrics (Cont'd)

Metric	Notes
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. This value is specified in the coherence properties.
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. This value is specified in the coherence properties.

MM Method Reference

TIBCO BusinessEvents Monitoring and Management provides the following groups of methods you can use to manage a deployed cluster:

- [Inference Agent Methods](#)
- [Process Methods](#)
- [Query Agent Methods](#)

See the following table for details.

Inference Agent Methods

Table 39 *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.

Table 39 Inference Agent Methods

Property	Notes
GetScorecards	<p>If URI is not provided: Returns a table with of all of the scorecards in this agent.</p> <p>If URI is provided: Retrieves the agent's scorecard information.</p>
Profiler Group	
<p>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 Chapter 22, BusinessEvents Performance Profiler in <i>TIBCO BusinessEvents Developer's Guide</i>.</p>	
StartFileBasedProfiler	Turns on the profiler and starts collecting data for the specified duration. When 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.
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.

Process Methods

Table 40 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.
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).
ActivateTraceRole	Enables a specified trace role.
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.
Object Management Group	
GetEvent	Retrieves an event with the specified Id from the specified agent or from every agent.

Table 40 Process Methods

Property	Notes
GetInstance	Retrieves an instance with the specified ID from the specified agent or from every agent.
GetScorecards	If URI is not provided: Returns a table with of all of the scorecards in the specified agent or in every agent. If URI is provided: retrieves scorecard information in the specified agent or in every agent.
Profiler Group	
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 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.

Table 40 Process Methods

Property	Notes
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.

Query Agent Methods

Table 41 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.

Deployment Using TIBCO Administrator or at the Command Line

This chapter explains how to deploy using TIBCO Administrator and at the command line.

Deployment is also available using TIBCO BusinessEvents Monitoring and Management.

Topics

- [Deployment Overview, page 184](#)
- [Starting a TIBCO BusinessEvents Engine at the Command Line, page 185](#)
- [Deployment in a TIBCO Administrator Domain, page 188](#)
- [Before Deploying a Project in a TIBCO Administrator Domain, page 190](#)
- [Deploying a Project in a TIBCO Administrator Domain, page 191](#)

Deployment Overview

The output of a design-time project is one or more Enterprise Archive (EAR) files and one or more Cluster Deployment Descriptor (CDD) 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 Monitoring and Management options. This is the recommended way. See [Chapter 8, Deploying and Managing Engines with MM, on page 145](#).
- At the command-line. See [Starting a TIBCO BusinessEvents Engine at the Command Line on page 185](#).
- To a TIBCO Administrator domain. See [Deployment in a TIBCO Administrator Domain on page 188](#).



For details about deploying TIBCO BusinessEvents Decision Manager classes (implemented virtual rule functions) see *TIBCO BusinessEvents Decision Manager User'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 a *hot deployment*. See [Chapter 11, Hot Deployment, on page 195](#).

Manually Copy Custom Function and Third-Party Jars at Deploy-time

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/tpc1` directory. If you choose a location that is not in the classpath, then update the classpath in the TRA file to include the location.

Starting a TIBCO BusinessEvents Engine at the Command Line

When testing a project it can be helpful to run it at the command line. Here is the format to use (see [Table 42, TIBCO BusinessEvents Command Line Engine Startup Options](#) for details):

```
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]
```

Table 42 TIBCO BusinessEvents Command Line Engine Startup Options

Option	Description
-h	Displays this help.
--propFile	<p>When you execute <code>be-engine</code>, by default it looks in the working directory (the directory where you execute the command) for a property file of the same name (<code>be-engine.tra</code>). 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 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 global variable. This value overrides any other design-time value. The format is <code>--propVar varName=value</code> , for example, <code>--propVar jmx_port=4567</code> .
-p	Allows you to pass one or more supplementary property files to <code>be-engine</code> . Specify the path and filename. Not often needed. See Supplementary Property Files on page 186 .
-n	Allows you to provide a name for the TIBCO BusinessEvents engine.
	The name provided here is used in the console and in log files. If you do not provide a name, the host name of the machine is used.
-d	Starts the debugger service on the engine for remote debugging.

Table 42 TIBCO BusinessEvents Command Line Engine Startup Options (Cont'd)

Option	Description
-c	Specify the path and filename for the cluster deployment descriptor (CDD) file you want to use. TIBCO BusinessEvents looks first in the file system, and then in the EAR file. The default is <code>default.cdd</code> .
-u	Specify the processing unit ID you want to use for this engine. This ID must exist in the CDD file you reference in the -c option. The default is <code>default</code> .
<i>EAR filename and path</i>	Specify the path and filename for the EAR file you want to use. If you don't 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 alternate file you specified using `--propFile`). See the -p option in Table 42 for basic information. This section provides more details.

In TIBCO BusinessEvents 4.0 and higher, 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 (*property=value*).

Note: 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`.

Deployment in a TIBCO Administrator Domain

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

Within an EAR file, a TIBCO BusinessEvents ARchive (BAR) file contains the compiled agent files for all agents. 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 enterprise archive resource.



- The only supported transport option in this release is the **local** option.
- TIBCO BusinessEvents encoding is ISO8859-1. This is not configurable. This encoding is also the default encoding of TIBCO Administrator.

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, you must use the username and password entered during installation of the TIBCO Administrator software. You can then create additional users and passwords as needed.

If TIBCO Administrator was already installed before you installed TIBCO BusinessEvents, you may 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). See [Order of Precedence at Runtime on page 11](#) for more details.

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*.

Overriding Global Variable Values in TIBCO Administrator

Global variables you define in a project appear automatically in TIBCO Administrator. As shown in [Deploying a Project in a TIBCO Administrator Domain on page 191](#), you can override variables at the deployment level or at the service level. Values set at the deployment level are used in all deployed engines. Values set at the service level are used for the specific engine you are deploying.

For details on setting and using global variables at design time, see Working with Global Variables in *TIBCO BusinessEvents Developer's Guide*.

Before Deploying a Project in a TIBCO Administrator Domain

You must update the `be-engine.tra` files to include the TIBCO Hawk information. You must build the EAR file. Do other tasks as needed.

Update Engine TRA Files for TIBCO Hawk Information

Update the `be-engine.tra` files on all machines where you will deploy TIBCO BusinessEvents engines as follows.

1. Open the `BE_HOME/bin/be-engine.tra` file for editing.
2. Set the value of `tibco.env.HAWK_HOME`.
3. Append the following to the value of `tibco.env.STD_EXT_CP`:
`%PSP%HAWK_HOME/lib%PSP%`
4. Append the following to the value of `tibco.env.PATH`:
`%PSP%HAWK_HOME/bin%PSP%`
5. Save the file.

Set Default or Specific CDD File and Processing Unit Names

If you want to use default values, name the CDD file `default.cdd` and keep it in the root of the project folder. In that 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 [Configuring the Processing Units Tab \(All OM Types\) on page 27](#).

Enable Service-Settable Global Variable Overrides and Build the EAR

Service settable global variables are only available if the "Include All Service Level global variables" checkbox in the Build Enterprise Archive dialog is checked. Then build the EAR. See *Building an Enterprise Archive (EAR File) in TIBCO BusinessEvents Developer's Guide* for details.

Deploying a Project in a TIBCO Administrator Domain

After you do any actions explained in [Before Deploying a Project in a TIBCO Administrator Domain on page 190](#), and build the EAR file, you are ready to deploy.

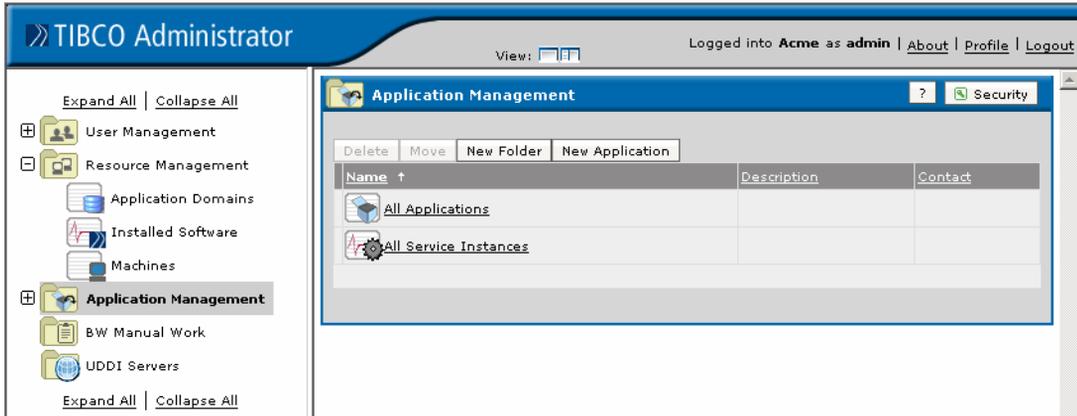


Do not use the fault tolerance features of TIBCO Administrator. Instead use the `Agent.AgentClassName.maxActive` property (see [Table 13, Agents Tab Settings and Properties, on page 64](#)). To maintain one active and one standby agent, deploy two agents of the same class and set the `maxActive` property to 1. (You can also deploy more than two agents and set the property to a larger number for different use cases.)

To Deploy a Project EAR in a TIBCO Administrator Domain

1. As needed, ensure that all 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 during installation, 8080 by default)
3. Select the administration domain for the application and provide the username and password assigned during installation, or other administrator user credentials.

4. If you are deploying a project for the first time, do the following:
 - a. Click **Application Management** (in the left panel).
 - b. Click the **New Application** button.

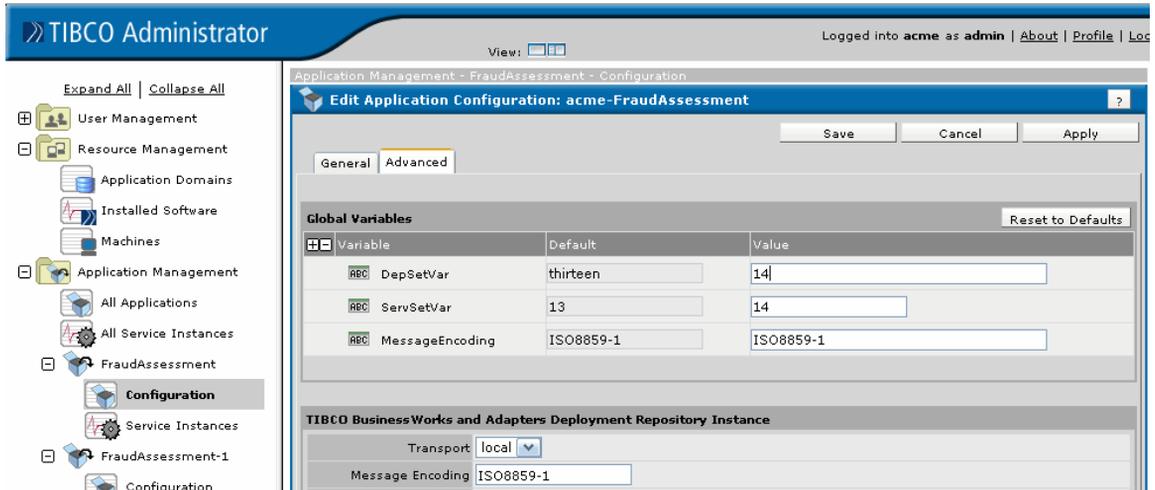


- c. At the Upload EAR File dialog, click **Browse**, select the EAR file you want to deploy, and click **OK**.
 - d. At the New Application Configuration dialog, set the Application Parameters and Services settings as desired (click Help for details). You can change default names:
 - Name: Set by default to the TIBCO BusinessEvents Studio project name
 - Deployment Name: Set by default to the TIBCO BusinessEvents Studio project name prepended with the domain name.
 - e. Click **Save**.

If the application does not appear in the list of applications, check [Before Deploying a Project in a TIBCO Administrator Domain on page 190](#) and ensure you have met all prerequisites.

You see the Configuration Builder panel. (For existing application, you can navigate here by expanding the explorer nodes on the left to **Application Management** > *application_name* > **Configuration**.)
5. To set deployment wide and engine level settings, such as the CDD and processing unit, and global variable overrides, do the following.
 - a. In the Configuration Builder panel, select the application (which is at the top level in the hierarchy) and select the **Advanced** tab. Ensure that the

Transport field is set to **local**. This is the only supported transportation option.



Deployment
Settable Variables

- b. At the Advanced tab, you can also override any deployment-settable global variable values as desired.
- c. Click **Save** when you are done. The Configuration Builder panel appears again.

PUID and CDD

- d. At the Configuration Builder panel, select the *application-name*.bar entry. It's one level below the top level in the hierarchy. Then select the **Advanced** tab.
- e. Enter the name of the CDD file and processing unit (PUID) you want to use for this engine.

Service Settable
Variables

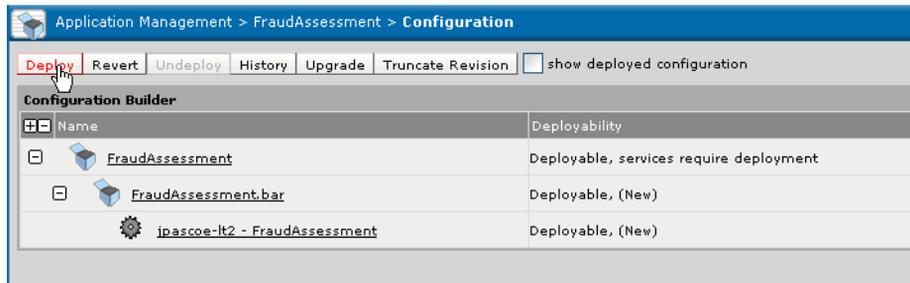
- f. At the Advanced tab, you can also override any service-settable global variable values as desired.
 - g. Click **Save** when you are done. The Configuration Builder panel appears again.
6. As desired, select the machines in the administration domain to which you will deploy the application:
 - a. In the Configuration Builder panel, click the service name. The service is nested under the application name. It has the same name as the

application, with the extension .bar). In the General tab, Target Machines panel, the current machine is available by default.

- b. As desired, select **Add to Additional Machines** to deploy the application to multiple machines. Select the machines to which you will deploy. You can also select the same machine more than one time, if you want to deploy the application more than once on a machine.
- c. Click **Save**.

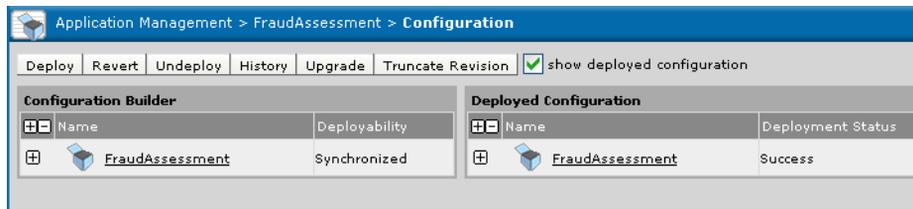
By default an engine starts when you deploy it. You can also start and stop engines as separate actions.

7. Navigate to the application's main Configuration Builder dialog and click the **Deploy** button.



8. At the Deploy Configuration dialog, configure settings if desired then click **OK**. The application deploys, and the Configuration dialog displays again.

You can check the **Show deployed configuration** checkbox to display the Deployed Configuration panel and verify success:



Chapter 11 **Hot Deployment**

This chapter explains how to use the hot deployment feature to deploy changes to a running application.

Topics

- [Hot Deployment Overview, page 196](#)
- [Modifications Allowed in Hot Deployment, page 197](#)
- [Enabling Hot Deployment, page 199](#)
- [Performing Hot Deployment in a TIBCO Administrator Domain, page 200](#)
- [Performing Hot Deployment Outside a TIBCO Administrator Domain, page 202](#)

Hot Deployment Overview

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 on page 199](#).

The changes you can make depend on the type of object management in use. The permitted changes are listed in the section [Modifications Allowed in Hot Deployment on page 197](#). 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 runtime. This is made possible using the `-javaagent` option. The `-javaagent` option is provided in the `be-engine.tra` file as shipped.

This chapter explains how to hot deploy with TIBCO Administrator, and also to an engine that was started at the command line. For details about hot deployment with TIBCO BusinessEvents Monitoring and Management, see [Hot Deployment on page 150](#).

Modifications Allowed in Hot Deployment

You can make only certain changes during a hot deployment. Supported changes are listed in [Table 43, Hot Deployment Supported Modifications](#). Supported modifications for In Memory and Berkeley DB object management are the same; for Cache OM they are more limited.



In some cases, you can work around a limitation. For example, you cannot hot deploy event modifications, so you can't change which rule function is used for an event's expiry action. However you can change the code of the rule function that is already specified.



Global Variables When using the hot deployment feature to deploy a project, TIBCO BusinessEvents ignores any changes to global variables in the project.

Table 43 Hot Deployment Supported Modifications

Resource	New	Modify	Delete
Rules*	Yes	Yes	Yes
Rule Functions*	Yes	Yes	Yes
Concepts	Yes		
Global Variables	Yes		
Simple Events	Yes		
Score Cards	Yes		
Time Events	Yes		
State Machines	Yes**		
State Machine Transitions	Yes**		
State Machine Timeout Expressions	Yes**		
State Machine Timeout Actions	Yes**		
State Machine Entry Actions	Yes**		
State Machine Exit Actions	Yes**		

Table 43 Hot Deployment Supported Modifications (Cont'd)

Resource	New	Modify	Delete
Channels and Destinations			

* **Cache object management** When Cache object management is used, hot deployment is available only for rules and rule functions. For state machines, you can change an existing action or condition, but you can't add anything (including changing an empty action to non-empty).

** **For state machine hot deployment** You can only hot deploy new state machines (and state machine components) 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.

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.

To Enable Hot Deployment

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 check the Hot Deploy check box. Repeat for all processing units you want to enable for hot deployment.
See [Chapter 3, CDD Configuration Procedures, on page 9](#) for more details about configuring CDD files.
4. Start the TIBCO BusinessEvents application Using the CDD file you updated. See the following sections as appropriate:
 - [Starting a TIBCO BusinessEvents Engine at the Command Line on page 185](#)
 - [Deployment in a TIBCO Administrator Domain on page 188](#)

For details on performing a hot deployment, see [Performing Hot Deployment in a TIBCO Administrator Domain on page 200](#) and [Performing Hot Deployment Outside a TIBCO Administrator Domain on page 202](#).

Performing Hot Deployment in a TIBCO Administrator Domain

This section explains how to perform hot deployment when the TIBCO BusinessEvents project has been deployed to a TIBCO Administrator domain.



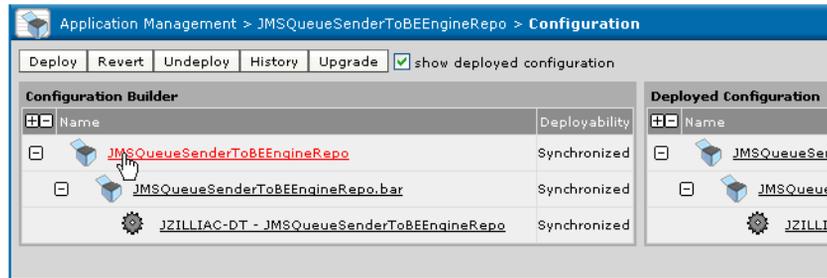
This procedure assumes the following requirements are met:

- The deployed processing units that you want to hot deploy to were enabled for hot deployment before they were deployed (see [Enabling Hot Deployment on page 199](#)).
- Your project complies with the requirements for deploying to TIBCO Administrator. See [Deployment in a TIBCO Administrator Domain on page 188](#).
- You have modified the TIBCO BusinessEvents Studio project and built the EAR file, following the limitations shown in [Modifications Allowed in Hot Deployment on page 197](#).
- The new EAR file has the same name as the existing one.

To Perform Hot Deployment in a TIBCO Administrator Domain

1. As needed, ensure that all the following are started on the machine running the processing unit or units you want to hot deploy to:
 - TIBCO Administrator service for the administration domain
 - TIBCO Hawk service for the administration domain
2. Start the TIBCO Administrator GUI:
 - Windows: Start > Programs > TIBCO > TIBCO Administrator Enterprise Edition *version* > TIBCO Administrator
 - Web browser: `http://host-name:port/` (where *host-name* is the machine name and *port* is the HTTP port specified during installation, 8080 by default)
3. Select the administration domain for the application and provide the username and password assigned during installation, or other administrator user credentials.
4. Expand to **Application Management** > *application_name* > **Configuration**.

- In the Configuration Builder panel, select the application (at the base of the tree).



- In the Edit Application Configuration dialog, click **Upload New EAR File**.
- At the Upload EAR File dialog, click **Browse**, select the EAR file you want to deploy, and click **OK**.
- Confirm the upload by clicking **OK** again, then click **Save**. Verify that the Deployability column displays *Deployable*.
- Click **Deploy**. You see the Deploy Configuration dialog.
- Uncheck** these checkboxes (if they are checked):
 - **Stop running services before deployment**
 - **Start successfully deployed services**
 - **Force redeployment of all services**
 (When the Stop running services before deployment checkbox is checked, you see an additional setting, Kill services that haven't stopped after (seconds). It is removed when you uncheck the checkbox.)
- 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

This section explains how to perform hot deployment when the TIBCO BusinessEvents project has not been deployed to a TIBCO Administrator domain.

You can only perform hot deployment if the deployed application was enabled for hot deployment before it was deployed (see [Enabling Hot Deployment on page 199](#)).

Modify the Project as Needed and Build the EAR File

In TIBCO BusinessEvents Studio, modify the TIBCO BusinessEvents project according to your needs. See [Table 43, Hot Deployment Supported Modifications](#) for a list of modifications you can make. Then rebuild the project EAR file.



The new EAR file must have the same name as the existing one.

Replace the EAR File that was Used to Start the Engine

Replace the EAR file used to start the engine with the modified EAR file. 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.

Chapter 12 **Threading Models and Tuning**

This chapter explains the different threading models you can use, and mentions the performance tuning controls available for each. The chapter focuses on Cache OM and backing store threading options.

Topics

- [Threading Models Overview, page 204](#)
- [Event Preprocessor and Rete Worker Thread Options, page 206](#)
- [RTC Options — Single-Threaded or Concurrent, page 210](#)
- [Post RTC Options — Cache-aside and Write-behind, page 212](#)

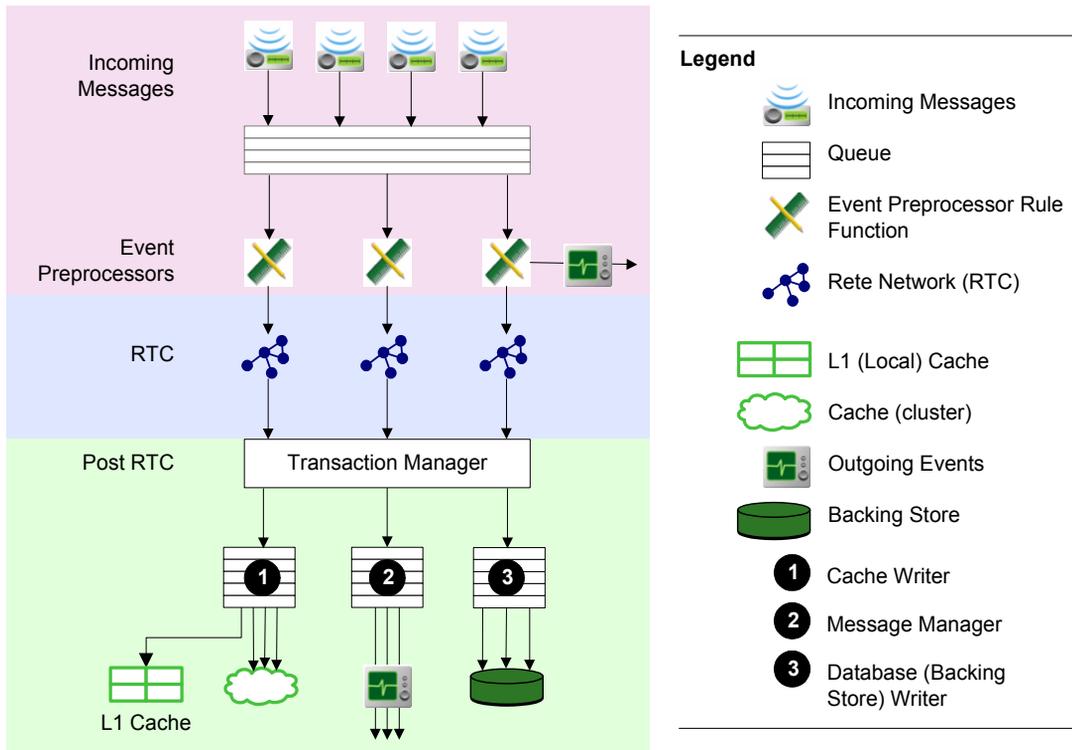
Threading Models Overview

When you begin to scale up messaging the following are the potential bottlenecks:

- Either messages are coming in too fast into the inference engine(s).
- Or the inference engines are not handing off objects fast enough to the cache agents (if write-behind strategy is used) or to the backing store (if cache-aside strategy is used)
- Or Cache agents are not accepting the objects fast enough.
- Or backing store is not accepting the objects fast enough.

These points are related. You can add more inference engines and more cache servers to address these issues, depending on where the bottlenecks are occurring. Below is a representative example flow for an inference agent. Later sections show more detail for options available at each phase.

Figure 2 Agent threading example — shared threads, concurrent RTC, cache-aside



- Event preprocessing is multithreaded. For each destination you choose a threading option: shared queue and threads; dedicated worker threads; caller's threads. These threads are released at the end of the RTC (post-RTC phase uses different threads).
- For the RTC phase, you can choose single or concurrent RTC options.
- For the post-RTC phase, you can choose cache-aside or write-behind thread management. Cache-aside is shown in the diagram above.
- Events can be sent out (and acknowledged) in the event preprocessor. Otherwise they are sent out in the post-RTC phase.

Event Preprocessor and Rete Worker Thread Options

This aspect of the threading model deals with messages that arrive at destinations. For each destination you choose a threading model type:

- Shared Queue and Threads
- Dedicated Workers
- Caller's Thread

The thread used carries execution to the post RTC phase, at which point execution is handed off to one of another set of threads, for writing to cache and backing store, if one is used.

In all threading models, event preprocessing is multi-threaded for high performance.

Diagrams in this section use as an example EMS messages, arriving at JMS destinations.

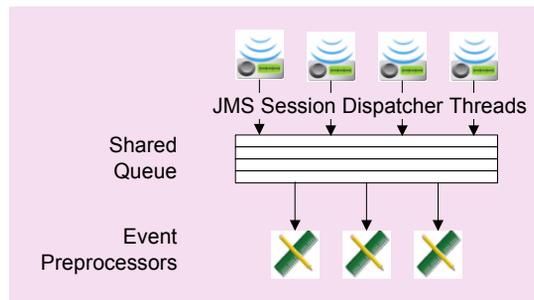
Each JMS destination creates a separate JMS Session internally, and creates a JMS thread and a dedicated JMS connection for itself.

All queues in all options are blocking, and FIFO (first in first out).

Additional tuning may be possible at the event level, depending on options available. For example the Enterprise Message Service server queue can be sized appropriately. This topic is outside the scope of TIBCO BusinessEvents documentation.

Note that events sent out in the event preprocessor phase are sent immediately and not in the post RTC phase.

Shared Queue and Threads



With this option, all destinations of all types share the TIBCO BusinessEvents system-wide queue and threads (JMS destination shown as an example). One pool of worker threads picks jobs from the shared queue. Execution continues on this thread through to post-RTC.

This is the default choice. It is most straightforward option The number of threads is not configurable, but the queue size is.

Property	Notes
----------	-------

CDD Editor > Collections > Destinations > Threading Model: Shared Queue — Queue Size	
--	--

Specifies the queue size for the system-wide shared queue.

For more details, see notes for

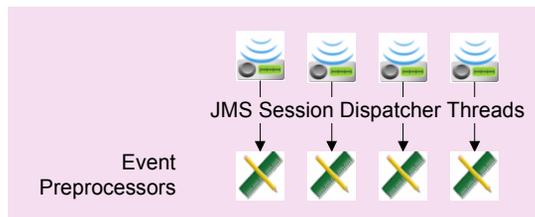
`com.tibco.cep.runtime.scheduler.default.numThreads`

If set to 0 (zero), the queue size is unlimited.

Default is 0.

- | | |
|---------------|--|
| Advantages | <ul style="list-style-type: none"> • Good for multi-core machines, which can make good use of a heavily threaded set-up. |
| Disadvantages | <ul style="list-style-type: none"> • Too many threads create context switching. • One single queue can become a bottleneck in the case of a sudden increase in incoming messages. • Correlation of events arriving on different queues at different rates can be problematic, as can request-reply situations. • It can be harder to tune performance with only one queue and one set of threads for all destinations. |

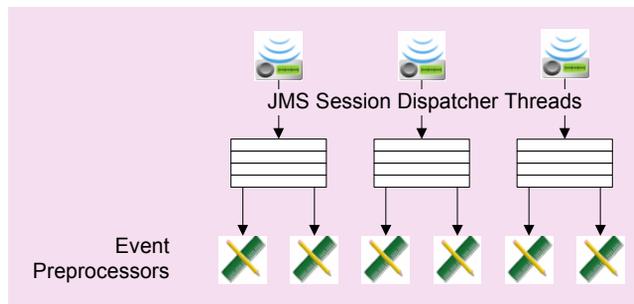
Caller's Thread



Uses the thread (and queue size) provided by the channel resource client (the Enterprise Message Service client, for example). There is one caller's thread per destination. The same thread executes the RTC phase.

- Advantages**
- The messaging library's thread does the message delivery, pre-processing and the Rete operations, so there is less context switching.
 - The messaging system cannot push events faster than the rate at which it can get consumed, so the system is self-throttling.
 - Best option for request-reply situations.
- Disadvantages**
- To scale up, many destinations have to be created in order to create that number of caller threads.
 - Because each destination creates a JMS session, a session might be under used. On some operating systems, sockets and sessions could be very under-used.

Dedicated Workers



This option is similar to the Shared Queue option except that each destination has a dedicated thread pool and set of worker threads to process messages.

Each destination has its own queue (of a configurable size) and its own set of dedicated worker threads (also configurable).

Property	Notes
CDD Editor > Collections > Destinations > Threading Model: Workers — Thread count	Specifies the number of dedicated worker threads for each destination
CDD Editor > Collections > Destinations > Threading Model: Workers — Queue size	Specifies the size of the queue used for each destination

- Advantages**
- Each destination can be configured differently, to deal with correlation of events arriving at different rates in different destinations, or events that are correlated in different ratios, such as correlation of every tenth event from destination one with every other event from destination two.

- If you use priority queues in Enterprise Message Service, you can use dedicated queues to service them efficiently.

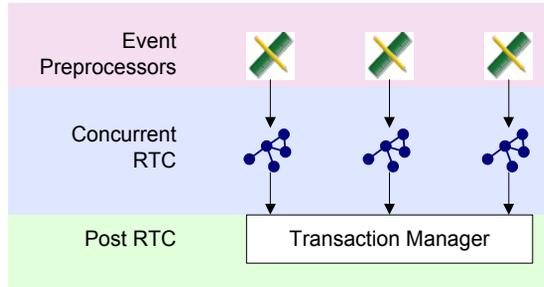
Disadvantages

- More complex to manage multiple queues and sets of threads.

RTC Options — Single-Threaded or Concurrent

RTC can be set to single-threaded or concurrent.

Concurrent RTC



One RTC executes simultaneously on each thread. All threads fill post RTC queues.

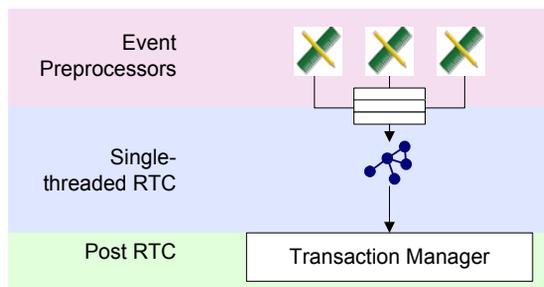
Advantages

- Can confer performance benefits, given correctly sized hardware and JVM configuration. Best on large high-capacity, high-performance machines.

Disadvantages

- When many smaller CPUs are used, then concurrent agents may give better performance than concurrent RTC.
- Requires the same kind of locking as for multi-agent concurrency to protect integrity of the data. The cost of locking negates some of the performance benefits of concurrency.
- If you also use concurrent agents, the system can become complex.

Single-Threaded RTC



Each worker thread waits to execute its RTC in turn.

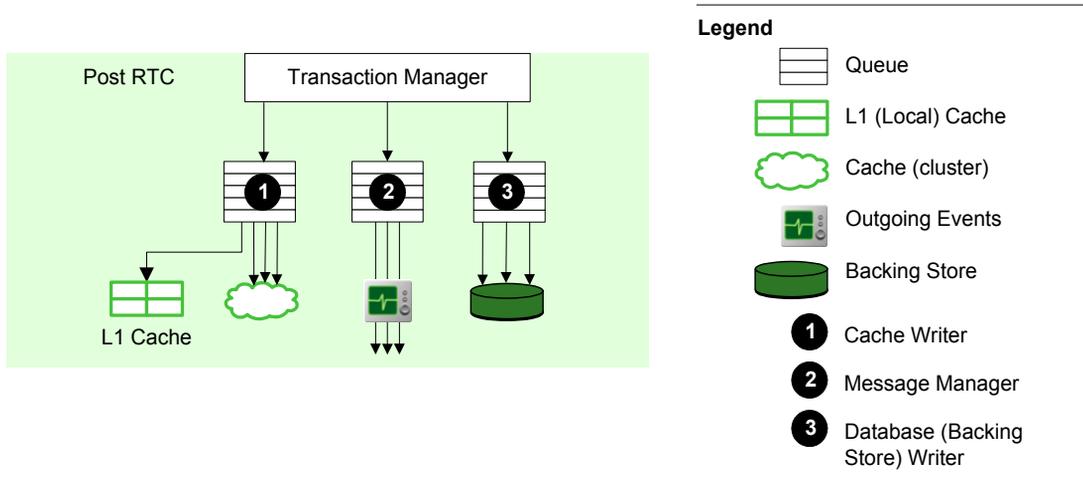
- Advantages
- Simpler. Does not require locking (unless concurrent agents are used).
- Disadvantages
- Less performant than concurrent RTC (depending on hardware used).

Post RTC Options — Cache-aside and Write-behind

In the post RTC phase, the cache is updated, outgoing events are sent, the backing store is updated, and locks (if any) are released. TIBCO BusinessEvents handles outgoing events on a separate set of threads, in both the write-behind and cache-aside options. However, each option handles writing to cache and to the backing store differently, as shown next.

Cache-aside

With cache-aside, inference agents manage writes to the cache, the local L1 cache, and the backing store, simultaneously in the post RTC phase.



When writing to the database, TIBCO BusinessEvents does the following actions in a single transaction:

```
deleteConcepts
deleteEvents
insertConcepts
insertEvents
modifyConcepts
removeObjectTable
saveObjectTable
```

Multiple transactions can be committed in one batch. The size of the batch is controlled by the `Agent.AgentClassName.dbOpsBatchSize` property as shown in the section [Tuning Properties for Cache-aside Strategy on page 213](#).

Tuning Properties for Cache-aside Strategy

Add to a CDD property sheet, at the level of Agent Classes, Processing Units, or Cluster, depending on the scope of the setting. For example to apply the settings to all agents in a cluster, set the property at the cluster level. You can also override the value at a lower level.

Property	Notes
<code>Agent.AgentClassName.threadcount</code>	<p>Defines the number of threads performing cache writing jobs.</p> <p>Tip This property is also used to define the number of Recovery threads (used for recovering Cache Plus Memory entity handles at inference engine startup).</p> <p>Default value is 2.</p>
<code>Agent.AgentClassName.dbOpsQueueSize</code>	<p>Defines the size of the Java <code>BlockingQueue</code> for database writing jobs.</p> <p>Zero (0) or any negative value means the queue size is unlimited.</p> <p>Default value is 8.</p>
<code>Agent.AgentClassName.dbthreadcount</code>	<p>Defines the number of threads performing database writing jobs.</p> <p>Default value is 2.</p>
<code>Agent.AgentClassName.dbOpsBatchSize</code>	<p>Defines the number of Rete transactions (RTCs) included in a single database transaction.</p> <p>Default value is 10.</p>
Advantages	<ul style="list-style-type: none"> • Provides transaction control, making sure transactions, including deletes, are performed following an RTC. • Allows batching of Rete transactions (RTCs). • Provides thread and queue size controls. • Offers the ability to use the database as the primary storage, and to use cache secondarily, to pass the objects between the Rete network and the database. This strategy is useful in some usage scenarios.

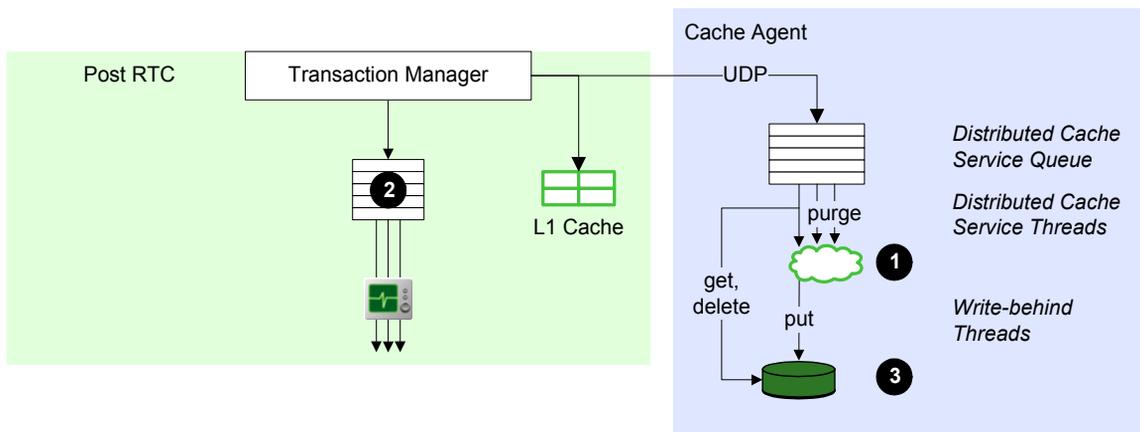
- Disadvantages
- There are really no disadvantages in comparison with write-behind. Cache-aside is a later implementation to offer improved controls based on experiences with the earlier write-behind method.

Write-behind

With write-behind strategy, cache agents handle writes to cache and to the backing store. First the cache agent writes data to the cache and then to the backing store.

For inserts and updates, one write-behind thread is used for each entity type. Deletes are performed by the distributed cache threads (configurable) and they are synchronously deleted from the database.

You can configure a write-delay property to define whether the write is synchronous or asynchronous. Write operations from multiple writers to the cache are batched.



- Advantages
- Batches writes during the delay period which increases database call efficiency and minimizes network traffic.
 - Offloads cache and database work to the cache agent.

- Disadvantages
- Does not offer transaction controls.
 - Can be slower than cache-aside.
 - If enough cache agents fail, the cache management layer won't be able to persist a write that was done previously, resulting in an inconsistent DB (This risk can be minimized by using a short write-behind delay or synchronous writes).

Cache Agent Thread tuning

Uses the following cluster-wide property:

```
java.property.tangosol.coherence.distributed.threads
```

Specifies the number of Coherence daemon threads used by the distributed cache service. If zero, all relevant tasks are performed on the service thread.

Chapter 13 **Configuring User Authentication**

This chapter explains how to set up user authentication using a file-based system, and using integration with an LDAP system. It also explains how you can use your existing JAAS login module instead of the one provided.

Topics

- [User Authentication Overview, page 218](#)
- [Configuring User Authentication, page 219](#)
- [Authentication Property Reference, page 221](#)

User Authentication Overview

When configuring user authentication, you choose which authentication option you want to use and configure that option. You can also use a different login module if you don't want to use the module shipped with the product.

Authentication is used for applications provided with BusinessEvents and BusinessEvents add-on products, such as TIBCO BusinessEvents Views and TIBCO BusinessEvents Decision Manager).

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. You can substitute a different implementation of the JAAS login module than the one provided. As another option, you may want to add the provided login module to your existing JAAS login configuration file (thus providing multi-stage authentication).

Authentication Options

Authentication options are set using CDD file properties. You can choose between the following options for 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.

LDAP Authentication This method authenticates users against a directory server using LDAP as a protocol. BusinessEvents applications can leverage this information to authenticate users. The role information is configured through an LDAP attribute (this is directory server specific) like the `nsrole` attribute in Sun Java System Directory Server.

Configuring User Authentication

This section explains how to select file-based authentication or LDAP-based authentication, and how to configure each authentication option. It also explains how to use a different JAAS login module, if you prefer to use a different implementation to the provided one. See [User Authentication Overview on page 218](#) for more background information.

To Use a Different JAAS Login Module

You can substitute a different implementation of the JAAS login module than the one provided. To configure the product to use your implementation, specify the location of your JAAS login configuration file using the following property at the appropriate level in the CDD file:

```
java.security.auth.login.config
```

For the TIBCO BusinessEvents Decision Manager BRMS application, the property is located in the cluster properties.

For the TIBCO BusinessEvents Views application, add the property in the `dashboard-class` agent properties for the TIBCO BusinessEvents Studio project.

For the TIBCO BusinessEvents Monitoring and Management component, the properties are found in the `mm-class` agent class, in the `mm/auth` property group.

To Configure the Authentication Type

If you use the provided JAAS login module, choose the authentication type and then configure it.

1. Open the project in TIBCO BusinessEvents Studio. Open the CDD file in the CDD editor.
2. As needed add the property `be.auth.type` and set the value to `file` or `ldap` as desired. Below are the property locations and values used in different cases:
 - RMS (BRMS project): In the Cluster tab properties, in the `RMS` property group. The value is `file`.
 - BEMM (emonitor project): In the `mm-class` agent class, in the `mm/auth` property group.
 - For TIBCO BusinessEvents Views applications, add the property to the `dashboard-agent` class agent properties.

3. If you enter `file`, then do the following to complete configuration:
 - a. As needed add the property `be.auth.file.location` and set the value to the location of your password file. Below are the property and file locations used in different cases:
 - RMS (BRMS project): In the Cluster tab properties, in the RMS property group. The shipped file location is `BE_HOME/rms/examples/users.pwd`.
 - BEMM (emonitor project): In the `mm-class agent` class, in the `mm/auth` property group. The file location is `BE_HOME/mm/config/users.pwd`
 - TIBCO BusinessEvents Views: Add the authentication properties in the `dashboard-agent` class properties for the BusinessEvents project. To see a configured example project, import the following example project into TIBCO BusinessEvents Studio and open its CDD file:
`BE_HOME/examples/views/TickeTracker`.
 - b. Configure the password file. See [To Configure the Password File for File Based Authentication on page 220](#) for details.
4. If you enter `ldap`, add and configure the LDAP properties shown in [Table 44, Authentication Configuration Properties, on page 221](#). Prior familiarity with LDAP is required. Details are not provided in this guide.

To Configure the Password File for File Based Authentication

In file-based authentication, you define a list of user names, passwords, and roles in a file called (by default) `users.pwd` file. This file is commonly referred to as the password file.

1. Locate and open the password file. Its location is specified in the CDD file (see [step 3a in To Configure the Authentication Type on page 219](#) for details).
2. Add each user on a separate line using this format:

```
Username:password:role,role,role;
```

The roles are used for authorization (access). However access control is currently used only in TIBCO BusinessEvents Decision Manager and is documented in *TIBCO BusinessEvents Decision Manager User's Guide*. For other applications, omit the roles. In all cases, do not use spaces. For example:

```
Mark:A31405D272B94E5D12E9A52A665D3BFE:BUSINESS_USER,APPROVER;
James:21232f297a57a5a743894a0e4a801fc3:RULE_ADMINISTRATOR;
```



You must hash the password with MD5 (Message-Digest 5) hashing algorithm.

Authentication Property Reference

Add these to the appropriate CDD file. See [Configuring User Authentication on page 219](#) for details.

Table 44 Authentication Configuration Properties

Property	Notes
<code>java.security.auth.login.config</code>	<p>Provides the absolute location for the login module configuration used by JAAS. See Pluggable JAAS Login Module on page 218 for more details. The value as shipped is:</p> <p><code>BE_HOME/rms/config/security/jaas-config.config</code></p>
<code>be.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 the LDAP Authentication Properties section of this table to work with your LDAP setup.</p> <p>Default is <code>file</code></p>
<code>be.auth.file.location</code>	<p>Specifies the absolute filepath to and name of the password file. This file is used for file-based authentication.</p>
LDAP Authentication Properties	
<code>be.auth.ldap.host</code>	<p>Specifies the domain name of the host for LDAP authentication.</p>
<code>be.auth.ldap.port</code>	<p>Specifies the port for LDAP authentication.</p>
<code>be.auth.ldap.adminDN</code>	<p>Specifies the base distinguished name (DN) for admin login. For example, <code>cn=Directory Administrators, dc=na, dc=tibco, dc=com</code>.</p>

Table 44 Authentication Configuration Properties (Cont'd)

Property	Notes
<code>be.auth.ldap.adminPassword</code>	Specifies the password for the LDAP administrator DN.
<code>be.auth.ldap.baseDN</code>	Specifies the base tree in LDAP under which users can be searched. For example, <code>dc=na, dc=tibco, dc=com</code> .
<code>be.auth.ldap.roleAttr</code>	Specifies the name of the attribute used by the LDAP server for role information of a user. Default value is <code>nsroleDN</code> (for Sun Java Directory Server).

Chapter 14 **Configuring Access Control for a Project**

This chapter describes how the access control system works, the types of resource specifications that are permitted, and how to grant permissions to different users.



In this release, access control is currently used only for RMS projects (available with the TIBCO BusinessEvents Decision Manager add-on).

Topics

- [Configuring Access Control—Overview, page 224](#)
- [Working with Access Control Files, page 227](#)
- [Resource Types and Corresponding Action Types, page 231](#)

Configuring Access Control—Overview

For each RMS project, you set up an access control file. In the access control file, you group the decision project resources as desired, giving each group (or individual resource) an ID, and then you assign permissions to each user role, using those IDs.

Establishing the User Roles

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 add one entry element for each user role. See [Structure of the Access Control File on page 225](#) for details.

See [User Authentication Overview on page 218](#) for details.



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.

Guidelines for Configuring Access Control

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 for using those resources, for a role named `CallCenter`:

```
<permission reref="#AllP">
  <action type="create">ALLOW</action>
</permission>
```

```
<permission reref="#CID">
  <action type="create">DENY</action>
</permission>
```

With these settings, you give users with the CallCenter role the create permission for all properties in the /Concepts directory except the custID property.



An example of an access control file giving full permissions is provided in the credit card application example, located in the following directory:

```
BE_HOME\rms\examples\CreditCardApplication\
CreditCardApplication
```

Deny everything
and specify
exceptions

The other 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.

Mixing these
approaches

You can mix 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 is an XML file with the following elements:

```
<acl xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ACL.xsd">
<resources>
  <resource id="id" type="ReType"/>
  <resource id="id" name="ProjectPath" type="ReType"/>
  .
  .
  .
</resources>

<entries>
  <entry>
    <role name="RoleName" />
    <permissions>
      <permission reref="#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 `reref` 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 resources in the specified project path are included in the permission.
- The action `type` attribute specifies an **action type**, for example, `create`.

See [Table 45, Resource Types and Action Types, on page 231](#).

Permissions—ALLOW and DENY

The value of the `action` element is one of the key words `ALLOW` or `DENY`. It 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. Use all capitals.

Working with Access Control Files



In this release access control is used only by TIBCO BusinessEvents Decision Manager. This section is tailored for use by that component.

Access control settings are created using XML files with the extension `.ac`. 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 RMS and Decision Manager.

Examples shipped with the product contain access control files you can use as models.

Creating and Modifying an Access Control File

Use any of the following methods to work with access control files:

- You can create or modify an *RMSProject.ac* file using any XML editor.
- If your user role has permission to do so, you can also create or modify an access control file in Decision Manager.
- To create an access control file in Decision Manager, select **Access > New**.
- To modify an existing access control file, select **Access > Open**.

Required Location of Access Control Files

The access control file for an RMS project must be placed in the project's `config` directory and it must be named using the format *ProjectName.ac*.

See *TIBCO BusinessEvents Decision Manager User's Guide* to understand where the project directory or directories are located.

Specifying and Grouping Decision Project Resources

In the `resources` element, you group the project resources in whatever way supports the permissions you want to set. You give each grouping or individual resource an ID that is used when defining the permissions.

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, you associate an ID with a resource type, and you do not use the name attribute:

```
<resource id="ID" type="ReType"/>
```

For example: `<resource id="C" type="CONCEPT"/>`

See [Table 45, Resource Types and Action Types, on page 231](#) for a list of resource types, and the action types that are valid for each resource type.

Defining Permissions

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 delete permission. All privileges are mutually exclusive.

For each resource type there is a predefined set of action types such as create, read, modify, and delete (see [Table 45, Resource Types and Action Types, on page 231](#)). For each resource ID defined in the resources section, you must grant permission separately for each action type. For example, you would add four permission elements to give a user role permissions to create, read, modify, and delete a specified group of resources of a certain type.



In the Decision Manager application, you cannot create, modify, or delete properties, rules, concepts, or rule functions.

Defining Permissions for Domain Models

In addition to permissions to work with a resource, you can also create permissions to work with that resource's domain model. Domain models are used only with resources of type PROPERTY.

The method is the same as for permissions to use resources. To specify that the permission is for domain models, and not for the resources themselves, you use the permission type, DOMAINMODEL.

Just as you create a set of permissions of each type that defines what a user role can do with the resource itself, you create a set of resources that defines what a user role can do with the domain model for the resource. For example, the following gives broad permissions for a user to work with all resources of type PROPERTY:

```
<permission reref="#PR">
  <action type="create">ALLOW</action>
</permission>
<permission reref="#PR">
  <action type="modify">ALLOW</action>
</permission>
<permission reref="#PR">
  <action type="delete">ALLOW</action>
</permission>
```

DOMAINMODEL
Permission Type

```
<permission reref="#PR" type="DOMAINMODEL">
  <action type="read">ALLOW</action>
</permission>
<permission reref="#PR" type="DOMAINMODEL">
  <action type="create">ALLOW</action>
</permission>
<permission reref="#PR" type="DOMAINMODEL">
  <action type="modify">ALLOW</action>
</permission>
<permission reref="#PR" type="DOMAINMODEL">
```

```
<action type="delete">ALLOW</action>  
</permission>
```

Resource Types and Corresponding Action Types

For each resource type, you define permissions for the action types available for that type of resource. Permissions to work with domain models for a property are similarly given, using the permission type `DOMAINMODEL`, as explained in [Defining Permissions for Domain Models on page 229](#).

Table 45 Resource Types and Action Types

Resource Type	Allowable Action Types	Notes
PROJECT	checkout, update, commit	Applies to actions taken at the project level (and not to all resources within a project).
FOLDER	read	Controls access to specified folders (or all folders if none are specified).
PROPERTY	read,	Controls access to the properties of the specified resource or resources (or all properties if no resources are specified).
CONCEPT	read	Controls access to specified concepts (or all concepts if none are specified).
EVENT	read	Controls access to specified events (or all events if none are specified).
RULESET	read	Controls access to specified rule sets (or all rule sets if none are specified).
RULE	read	Enables users to view the source for specified rules (or all rules if none are specified).
RULEFUNCTION	read, add_impl, del_impl, invoke	<p>The read permission controls read access to specified rule functions (or all of them if none are specified).</p> <p>The add_impl permission enables a user to add decision tables (VRF implementations).</p> <p>The del_impl permission enables a user to delete decision tables.</p> <p>The invoke permission enables a user to invoke specified rule functions (or all of them if none are specified) in decision tables.</p>

Table 45 Resource Types and Action Types (Cont'd)

Resource Type	Allowable Action Types	Notes
CATALOGFUNCTION	invoke	Enables a user to invoke specified catalog functions (or all of them if none are specified) in decision tables. Validation errors are thrown if other functions are written manually.
RULEFUNCTIONIMPL	read	Controls access to the specified decision tables (or all decision tables if none are specified). (Other permissions for decision tables are set on the resources used in the decision table.)
DOMAIN		Controls access to the specified domain models.

Chapter 15 JDBC Backing Store Configuration

A backing store enables persistent backup of the objects generated and modified at runtime. Use of a backing store enables recovery in the event of a system-wide failure.

Oracle-Only Backing Store The Oracle-only backing store available in earlier releases is deprecated in this release. It is documented in [Appendix A on page 259](#). Alternatively you can migrate to the JDBC backing store feature. See *TIBCO BusinessEvents Installation* for details on migration.

Topics

- [JDBC Backing Store Configuration Overview, page 234](#)
- [Special Cases, page 235](#)
- [Resources Required for Setting Up the Database, page 237](#)
- [JDBC Backing Store Database Configuration Tasks, page 241](#)
- [Updating an Existing Backing Store Database Schema](#)
- [Adding a JDBC Connection Resource to the Studio Project, page 252](#)
- [Configuring Backing Store Properties in the CDD Editor, page 253](#)
- [Backing Store Table Reference, page 255](#)

JDBC Backing Store Configuration Overview

The backing store feature requires use of Cache object management. Before you add a backing store, develop your caching solution and test it.

Task Summary

The main tasks in setting up a backing store are as follows:

- Ensure that you have access to a supported database management product and driver, and can create a user with permission to connect to the database and create tables and views. See [Resources Required for Setting Up the Database on page 237](#).
- Configure the backing store database using the provided utilities. See [JDBC Backing Store Database Configuration Tasks, page 241](#).
- Configure a JDBC Connection in your TIBCO BusinessEvents Studio project. See [Adding a JDBC Connection Resource to the Studio Project, page 252](#). You'll reference the connection in CDD settings.
- Configure CDD settings. See [Configuring Backing Store Properties in the CDD Editor, page 253](#)

Maintenance and Related Tasks

- If your project ontology changes, you must update the backing store schema. See [Updating an Existing Backing Store Database Schema on page 248](#).
- For instructions on migrating from a 3.x backing store to the JDBC backing store, see *TIBCO BusinessEvents Installation*, which has a migration chapter.

Special Cases Certain aspects of a project ontology may require special handling. See the following sections for details on how to handle these issues:

- [Names that Exceed the DBMS Maximum Column Length, page 235](#)
- [String Properties That Exceed the Maximum Database Column Length, page 236](#)
- [Ontology Identifiers That Use Database Key Words, page 236](#)

Special Cases

This section explains some cases that may require special handling.

Names that Exceed the DBMS Maximum Column Length

Entity names and entity property names are used by backing store scripts to generate database table and column identifiers. (Entity names can be overridden by the entity metadata property called Table Name.)

DBMSs put different limits on the length of a database identifier name. For example, in Oracle the maximum length is 30 characters, and in SQL Server the limit is 128 characters.

Note that the generated database identifiers are longer than the TIBCO BusinessEvents identifiers because they contain characters in addition to the TIBCO BusinessEvents identifier.

You can deal with this potential situation in different ways. One way is to configure project settings before you generate the EAR, and the other is configured by running the `be-jdbcdeploy` utility without modifying the project first. These are explained next.

Allowing the Utility to Generate Aliases for Long Names

When you run the `be-jdbcdeploy` utility, it generates short aliases for long names. For every entity, property, or state machine whose database identifier name exceeds the maximum length, an entry is created in the generated `yourname.aliases` file (For example, `acme.aliases`). This file has no entries if all names are within the DBMS limit.

You can edit the file to specify different short aliases. If you edit the aliases file, you must then run the `be-jdbcdeploy` utility again, so that your names are used.

The advantage of allowing the scripts to generate aliases is that it is automatic and you are in no doubt that the names will be of a correct length. However these auto-generated names might make it hard to identify what entity they represent, by simply looking at the table names.

The procedure is explained in [Task F, Check the Aliases File and Modify Aliases as Desired, on page 245](#).

Providing Short Names in the Project, Before Generating the EAR

You can avoid the problem of long names before you begin to configure the backing store; you can specify short database identifiers using metadata properties.

The advantage of this method is that you can choose meaningful names. The disadvantages are that you may not know ahead of time which entities and entity properties require short names, and that you must also ensure that the metadata names are unique across all entities in the ontology.

If entity names are repeated, table names are appended with dollar (\$) characters as necessary, for example, `D_ORDER`, `D_ORDER$`, `D_ORDER$$` and so on.

The procedure is explained in the section [Set Metadata Properties for Long Identifiers, as Desired on page 242](#), which is part of [Task A, As Needed, Set Entity Metadata Properties, on page 242](#)

String Properties That Exceed the Maximum Database Column Length

The default column size for String type attributes is 255 characters. If you expect the data length of an entity property to exceed that value, then before you begin backing store setup, open the editor for each such property and set the metadata Max Length field as needed. The utility changes the data type of String attributes with long lengths to CLOB, as appropriate.

Ontology Identifiers That Use Database Key Words

As well as database names that are too long, ontology terms that are key (reserved) words in your DBMS product must also be mapped to an alias. If errors occur when you run the SQL scripts due to key word clashes, examine the errors and add the appropriate words to the key word mapping file.

A provided file (`BE_HOME/bin/dbkeywordmap.xml`) ships with some basic mappings: `start`, `end`, and `schema`. You can use it as a model.

Unlike the Aliases file, the key word mapping file is not a project-specific file. It is intended to be generally useful across different projects. However, keyword mappings are also added to the aliases file when you run the SQL scripts, so you can also provide project-specific aliases for the generic mappings, if you want to. See [JDBC Backing Store Database Configuration Tasks on page 241](#) for full details.

Resources Required for Setting Up the Database



At time of writing, Oracle Database and Microsoft SQL Server are supported. Any later additions would be noted in the product readme file.

DBMS Software and Installation Requirements

The following sections contain a few relevant notes about the database products. See the product readme file for information about supported database versions.

Database Location	Instructions in this chapter assume you are working with a local database for testing purposes. Adapt the instructions if you are working with a remote database. For example, in production environments, you might have to ask a database administrator to create a database user for you. You should then be able to run the other SQL scripts yourself, logged on as the user created by the administrator.
-------------------	---

Minimum User Permissions

By default the TIBCO BusinessEvents user permissions are set to DBA privileges. At a minimum, the user must be able to create tables and views. For example for an Oracle database you could use the following:

```
DROP USER BE_ORCL CASCADE;
CREATE USER BE_ORCL IDENTIFIED BY BE_ORCL;
GRANT CONNECT TO BE_ORCL;
GRANT resource TO BE_ORCL;
GRANT CREATE ANY VIEW TO BE_ORCL;
GRANT CREATE ANY TABLE TO BE_ORCL;
```

SQL Server

Here are a few helpful points about SQL Server:

- For non-production purposes, it is convenient to use SQL Server authentication so you can create database users as needed. Select this option when you install Microsoft SQL Server. With Windows Authentication, on the other hand, you may have difficulties creating users without help from others in your enterprise.
- The datetime datatype in SQL Server 2005 has the following range: 1/1/1753 to 12/31/9999.

- Microsoft SQL Server 2008 has added a new data type, `datetime2`, which has a date range of 0001/01/01 through 9999/12/31. Therefore, if you are using Microsoft SQL Server 2008, then you can manually change the generated SQL script (DDL) for your backing store, and replace any affected columns' data type from `datetime` to `datetime2`.
- Use the SQL Server JDBC driver, `sqljdbc4.jar`. You can download this driver from:

<http://msdn.microsoft.com/en-us/data/aa937724.aspx>

Oracle

Here are a few helpful points about Oracle:

- You can download the Oracle JDBC driver from:

http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/jdbc_111060.html

— For Oracle Database 10g use `ojdbc14.jar`

— For Oracle Database 11g use `ojdbc6.jar`

- Maximum length for an Oracle table name or column name is 30 characters.

Provided Configuration Resources

The table below lists resources required and sections following explain the procedures for setting up backing store tables.

Table 46 Resources Required for JDBC Backing Store Implementation

Resource	Default Location and Notes
<code>cep-backingstore.jar</code>	<code>BE_HOME/lib</code> A JAR file required for backing store functionality (for both Oracle and JDBC store features).
Provided Files in <code>BE_HOME/bin</code>	
<code>base_types.xml</code>	The <code>base_types.xml</code> file is used by the deployment utility. Do not edit this file. (The <code>base_types.sql</code> file in the same location is used by the Oracle-only backing store and is not needed here.)

Table 46 Resources Required for JDBC Backing Store Implementation (Cont'd)

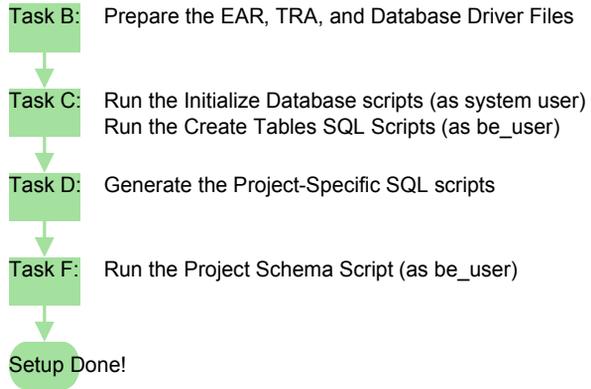
Resource	Default Location and Notes
be-jdbcdeploy executable be-jdbcdeploy.tra	Executable and property files used to generate SQL scripts (see Generated SQL Scripts below).
create_tables_oracle.sql create_tables_sqlserver.sql	Use the appropriate SQL (DDL) script for your DBMS. This script creates the tables that are used to maintain the metadata. The script drops any existing tables and recreates them.
dbkeywordmap.xml	This file contains mappings to handle words used in the TIBCO BusinessEvents project that are database reserved words. See Ontology Identifiers That Use Database Key Words on page 236 for details.
initialize_database_oracle.sql initialize_database_sqlserver.sql	Use the appropriate script for your DBMS. Note This script drops the user (and therefore all the tables) and adds the user again. By default the user is called be_user with the password be_user and the user has DBA rights. Edit the script if you want the user to have a different name or different rights. For SQL Server, this script also creates the default database, with the name be_user and makes it the default database for the user be_user.
Generated SQL Scripts	
These scripts are generated when you run the be-jdbcdeploy executable. Provide a value for <i>yourname</i> when you generate the scripts. They are located in the same directory where you run be-jdbcdeploy.	
<i>yourname</i> .sql	This SQL (DDL) script creates schema tables and types.
<i>yourname</i> .aliases	This script has entries if the database table identifiers are longer than the DBMS maximum character limit. See and Task F, Check the Aliases File and Modify Aliases as Desired, on page 245 .

Table 46 Resources Required for JDBC Backing Store Implementation (Cont'd)

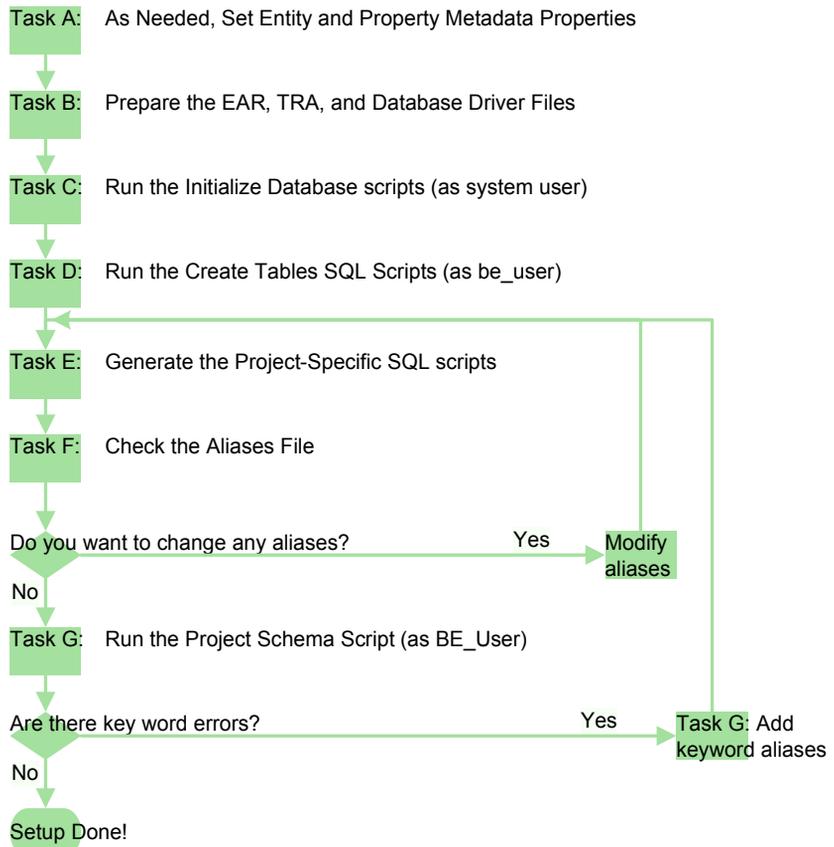
Resource	Default Location and Notes
<code>yourname_alter.sql</code>	The <code>yourname_alter.sql</code> script is for use in schema migration. See Updating an Existing Backing Store Database Schema on page 248 .
<code>yourname_remove.sql</code>	For use as needed. This script removes the database schema. You can use it to reset the project.
<code>yourname_cleanup.sql</code>	For use as needed. This script truncates the tables.

JDBC Backing Store Database Configuration Tasks

Simple Case



Complex Case



Task A As Needed, Set Entity Metadata Properties

You may need to modify your TIBCO BusinessEvents Studio project before you generate the EAR file. Check the project and, as needed, do the following:

- [Set Metadata Properties for Long Identifiers, as Desired](#)
- [Set the Maximum Column Length for Long String Values, as Needed](#)

Set Metadata Properties for Long Identifiers, as Desired

You can let the `be-jdbcdeploy` utility create aliases for long identifiers, or you can set short database identifier names before generating the EAR, using metadata properties. For more details about this choice, see [Names that Exceed the DBMS Maximum Column Length on page 235](#).

To set short database identifiers for affected entities and properties do the following.

1. In TIBCO BusinessEvents Studio, open the entity editor for each affected item.
2. For an entity name that is too long, in the Metadata section of that entity's editor, set the Table Name field to the desired value.



Note that the entity property metadata field Column Name is not used in this release.

It is recommended that you specify table names that start with "D_" to match the standard naming convention.

3. Save the resource.

Set the Maximum Column Length for Long String Values, as Needed

For properties whose value could be longer than 255 characters, do the following. See [String Properties That Exceed the Maximum Database Column Length on page 236](#) for more details.

1. In TIBCO BusinessEvents Studio, open the entity editor for each affected item.
2. Right-click the row for each affected property. From the shortcut menu, select **Metadata**. You see a pop-up dialog.
3. Set the Max Length field to the expected maximum length of a value.
4. Save the resource.

Task B Prepare the EAR, be-jdbcdeploy.tra, and Database Driver Files

As desired, you can configure the JDBC Connection resource for your backing store before creating the database, if you know the values to use. After you have created the database, it's a good idea to open your project again and test the connection. If you need to make corrections to the connection, do so and rebuild the EAR before deploying the project. The procedures in this guide assume you will configure the resource after setting up the database.

1. Ensure that you have a valid EAR for your project, with up-to-date ontology definitions. Model information in the EAR will be used to build tables in the database.
2. Ensure that `cep-backingstore.jar` is located in `BE_HOME/lib` (or other location in your class path).
3. As needed, copy your JDBC drivers file to `BE_HOME/lib/ext` (or other location in your class path).
4. Open the `be-jdbcdeploy.tra` file for editing
 - a. Specify which type of database you are using. The default value is `oracle`. Do not change the other configuration properties shown:

```
# BE base type schema file
java.property.jdbcdeploy.bootstrap.basetype.file %BE_HOME%/bin/base_types.xml
java.property.jdbcdeploy.bootstrap.keyword.file %BE_HOME%/bin/dbkeywordmap.xml
java.property.jdbcdeploy.database.type [oracle | sqlserver]
```

- b. Also set the environment variables to appropriate values, as needed.

Task C Run the Initialize Database Script as the System User

This script creates the TIBCO BusinessEvents user and initializes the database.



Running the `initialize_databaseYourDBMS.sql` script deletes the user before creating it again. Running the `create_tables_YourDBMS.sql` drops all database tables before creating them again. This means you can run these scripts again during test phases of your project development, without having to take extra cleanup steps.

The first time you run the scripts, you see harmless error or warning messages because there is nothing to delete.

If you are updating the schema for an existing backing store, see [Updating an Existing Backing Store Database Schema on page 248](#).

1. As desired, change the default TIBCO BusinessEvents user credentials: Open the `initialize_databaseYourDBMS.sql` script for editing and change the default username and password. The documentation uses the default username (`be_user`) and password (`be_user`)
2. Navigate to the location of the scripts (by default in the `BE_HOME/bin` directory) and open an SQLPlus or OSQL prompt. (For example, open a command window, type `SQLPlus` then provide the `system` user credentials.)
3. For SQL Server, type a command like the following at the prompt:

```
osql -S Your-Server-Name -U system_user -P sys_user_password -n -i
initialize_database_sqlserver.sql
```

For Oracle you would use a command like this:

```
sqlplus system_user/sys_user_password@SID @ initialize_database_oracle.sql
```

This script creates the TIBCO BusinessEvents database user. This user must be used to run the other scripts. You see messages like the following:

```
DROP USER be_user CASCADE
*
ERROR at line 1:
ORA-01918: user 'BE_USER' does not exist

User created.
Grant succeeded.
SQL>
```



Using your database product, you can configure additional users to access the database, in addition to this user.

Task D Run the Create Tables Scripts as the TIBCO BusinessEvents User

Next you log on as the TIBCO BusinessEvents user, `be_user` by default and run a script to create non-project specific tables.

1. Navigate to the location of the scripts (by default in the `BE_HOME/bin` directory) and open an SQLPlus or OSQL prompt. (For example, open a command window, type `SQLPlus` then provide the `system` user credentials.)
2. For SQL Server, type a command like the following at the prompt:

```
osql -S Your-Server-Name -d Your-DB-Name -U be_user -P be_user -n -i
@create_tables_sqlserver.sql
```

For Oracle you would use a command like this:

```
sqlplus be_user/be_user@SID @ create_tables_oracle.sql
```

Use the credentials defined in the `initialize_database_oracle.sql` or `initialize_database_sqlserver.sql` files. By default those are: username `be_user`, with password `be_user`.

Task E Generate the Project-Specific SQL Scripts

Open a command window and navigate to `BE_HOME\bin`. Run `be-jdbcdeploy.exe` using a command with the following format:

```
be-jdbcdeploy [-p property file] [-o schema output file] [EAR Path] [-h]
```

For example:

```
be-jdbcdeploy -o acme c:/BEProjects/MyEar.ear
```

The generated scripts appear in the directory where you run the executable. For example, if you provide the schema output filename `acme`, you would see files called `acme.sql`, `acme.aliases`, and `acme_remove.sql`.

The user-defined part of the database schema is in the schema output file (`yourname.sql`) as schema definition commands. In [Task G](#) you run this script (together with provided scripts) to build the schema in the database.

The options are explained in the following table:

Option	Description
<code>-p, /p,</code> <code>-property, or</code> <code>/property</code>	Specifies the property file. If not specified, the default property file is used, that is, <code>be-jdbcdeploy.tra</code> in the current directory.
<code>-o</code>	Specifies the schema output filename for deployment.
<code>-h, /h, or /help</code>	Displays this help.

Task F Check the Aliases File and Modify Aliases as Desired

The aliases file contains any table names that are longer than the database maximum length for table names, and auto-generated short aliases for them. (For example, the Oracle limit is 30 characters, and the SQL Server limit is 128 characters.)

It's a good idea to check the aliases file for entries, even if the TIBCO BusinessEvents names are not very long. The length of the generated database table names is not easy to predict.

Optionally, you can edit the file to provide more meaningful names.



It is recommended that you keep the aliases file for future reference. If the project ontology changes after the backing store has data in it, you must also update the database schema to match (as explained in [Updating an Existing Backing Store Database Schema on page 248](#)). If you modified the generated aliases, you must use the same aliases again when you update the schema, to preserve those columns and their data.

Key word
mapping file

Entries in the key word mapping file are also added to the aliases file so you can replace the key word aliases with project-specific ones, as desired (generally in a second pass). For details see [Task H, If Needed — Map Key \(Reserved\) Words to Aliases, on page 247](#).

1. Open the `yourname.aliases` file for editing.
2. Replace any aliases as desired with more meaningful short names. Make sure that each name is unique. It's a good idea to leave any system generated prefixes or suffixes in place for consistency of names across the database.
3. Perform [Task E, Generate the Project-Specific SQL Scripts, on page 245](#), again. This time, the aliases you created are used.

Task G Run the Project Schema Script (as `be_user`)

In this step, you log on as the user you created and run a script to create the project related part of the database schema.

The schema combines the definitions in `base-types.sql`, `create-tables.sql`, and the generated schema file, `yourname.sql` (`acme.sql` as an example).

1. Login to the database server as `be_user`, password `be_user`—or whatever username and password you set in the script in [Task C](#).
2. Navigate to the location of the scripts and open an SQLPlus prompt (Oracle) or OSQL prompt (SQL Server). Identify yourself as the `TIBCO BusinessEvents` user again.
3. At the SQL prompt, type the following to run the script:

```
@yourname.sql (for example, @acme.sql )
```

If there are no errors, your database tables are now configured for use. If there are errors you may need to add some mappings to the key word mapping file.

Task H If Needed — Map Key (Reserved) Words to Aliases

Complete this task only if you saw errors after completing [Task G, Run the Project Schema Script \(as be_user\)](#). Such errors are caused when your project ontology uses terms that are key words (reserved terms) in the DBMS you are using. You must map these terms to an alias in the keyword mapping file.

1. Edit the `BE_HOME/dbkeywordmap.xml` file to add entries. Below is the format followed by an example:


```
<keyword name="dbKeyWord" mapname="nonDbKeyWord" />
<keyword name="start" mapname="start_" />
```
2. Repeat [Task E, Generate the Project-Specific SQL Scripts, on page 245](#), and tasks following as needed.



Providing Project-Specific Key Word Aliases When you repeat [Task E](#), the new key words are added to the `yourname_aliases` file. You can create project-specific aliases for the key word mappings as desired. Then repeat [Task E](#) again and continue.

Note that you must generate the SQL scripts a total of three times if you add keyword mappings to the aliases file — a summary of the whole procedure is as follows:

1. *Generate the SQL scripts and run them (as explained in the procedures).*
2. *Errors occur due to key word clashes, so you add the appropriate key word mapping entries to the key word mapping file.*
3. *Generate the SQL scripts again.*
4. *To use project-specific aliases for the keyword mappings (Optional):*
 - a. *Edit the aliases file entries for the key word mappings.*
 - b. *Generate the SQL scripts again.*
5. *Run the SQL scripts to create the backing store.*

Next Step

After you have set up the database, perform project configuration activities as explained in See [Adding a JDBC Connection Resource to the Studio Project on page 252](#) and [Configuring Backing Store Properties in the CDD Editor on page 253](#) for details.

Also, remember to update your schema if your ontology changes. See [Updating an Existing Backing Store Database Schema on page 248](#)

Updating an Existing Backing Store Database Schema

If you change the project ontology, that is, if you create, alter or delete a concept or an event, you must update the backing store schema so it matches the updated ontology. You must do this before you deploy the updated project.

What the Schema Update Utility Can and Can't Handle Automatically

You must examine the alter script before you run it. Entries that could result in data loss are commented. Decide what changes to make manually and what changes to make using the script, taking into account the kind of data in the tables. Remove or comment entries for changes you will make manually.

Adds

The schema migration utility handles addition of entity types and attributes. New entity types and attributes are added to the database schema.

Changes (Drop and Add) — Assess individually

The utility handles changes to attributes (entity properties) as `DROP` and `ADD` operations. However, `DROP` operations are commented in the script to avoid data loss.

If a column is empty, or you don't want to keep the data they contain, you can enable the `DROP` operation and let the utility handle the change.

If the column contains data that you want to keep, then make the change manually using an appropriate database tool. For example, you can change the data type of a column from string to double without loss of data, as long as all the column values are numeric values.

Entity Deletions

If an entity is deleted from the Studio project, the corresponding tables are not dropped from the database schema. Existing data is not lost. Deleted entities are not mentioned in the alter script. Manually keep track of and delete such tables as needed.

Attribute Deletions

The schema update utility does handle deletion of entity attributes. SQL statements for deleted attributes are generated but they are commented. Examine the alter script and enable these commands if you want to execute them. Note that existing data is lost when you drop an attribute.

Example Alter Script

Below is an example *yourname_alter.sql* script.

```

Property type change  -- ##### WARNING : Non-alterable Ontology changes found. Please
                        see following errors. Manual schema-migration is required.

                        --* For Concept Concept1 field PROPERTY_1 type changed from
                        VARCHAR2 to LONG

                        -- ALTER TABLE D_Concept1 DROP ( Property_1 );

                        ALTER TABLE D_Concept1 ADD ( Property_1 numeric(19) );

New table              DROP TABLE D_Book_rrf;

                        CREATE TABLE D_Book_rrf (pid numeric(19), propName char
                        varying(255), id$ numeric(19) not null);

New property          -- ALTER TABLE D_MyConcept DROP ( FOLDER_1 );

                        ALTER TABLE D_MyConcept ADD ( Folder_0 char varying(255) );

```

The Procedure

To Update an Existing Backing Store Database Schema

1. To prepare for the update, do the following:
 - Gracefully shut down the deployed application (all agents and cache servers).
 - Back up your existing database.
 - Generate the updated EAR file for the modified project.
 - If you modified aliases when you created the schema, locate the *yourname.aliases* file you used. It will help you to modify those aliases in the newly generated file, so they match.
2. Open the `be-jdbcdeploy.tra` file for editing and set the following properties:

```
be.jdbc.schemamigration.url=DbURL
be.jdbc.schemamigration.user=username
be.jdbc.schemamigration.pswd=password
```

- Use the database URL that points to the existing backing store. See [Adding a JDBC Connection Resource to the Studio Project on page 252](#) for example URLs.
- Use the same username and password you used when setting up the backing store. See [Task C, Run the Initialize Database Script as the System User, on page 243](#).

These properties enable the program to compare the schema of the existing database with the ontology in the project EAR file, and generate the alter script.

3. Log on as the user name you specified in [Task C, Run the Initialize Database Script as the System User, on page 243](#).
4. Run the `be-jdbcdeploy.exe` utility as explained in [Task E, Generate the Project-Specific SQL Scripts, on page 245](#), using the *updated* EAR file.
5. If any of the new or changed definitions result in entries in the *yourname.aliases* file, and you want to change the provided aliases, follow instructions in [Task F, Check the Aliases File and Modify Aliases as Desired, on page 245](#). If you modify aliases, remember to generate the scripts again so the modified aliases are used.



You must use the same aliases that you used before. If any were modified when the schema was created, you must modify them the same way when updating the schema. It can be useful to refer to the original aliases file.

6. Examine the generated `yourname_alter.sql` script and modify as needed so you only run statements for changes you want to make. See [What the Schema Update Utility Can and Can't Handle Automatically on page 248](#) for details.
7. Run the `yourname_alter.sql` script.

Your database tables are now configured for use.

Adding a JDBC Connection Resource to the Studio Project

Add a JDBC Connection resource to your project and configure it to connect to the backing store. Details below use Oracle 10g database as an example. Adapt the instructions as needed for your database. The main section for the JDBC Connection Resource is provided in *TIBCO BusinessEvents Developer's Guide*. See the section JDBC Connection in that guide for a reference to the fields.



- The value of the CDD Cluster tab > Backing Store > Connection > Max Size field overrides the value of the JDBC Connection Resource Max Connections setting.
- You must first add your DBMS product's libraries to the TIBCO BusinessEvents Studio classpath. See Adding External Libraries to the BusinessEvents Studio Classpath in *TIBCO BusinessEvents Developer's Guide*.

1. In TIBCO BusinessEvents Studio, open your project, and open the folder where you keep shared resources and select **New > Other. > TIBCO Shared Resources > JDBC Connection**.

2. In the JDBC Driver field, select the driver for your database, for example, **oracle.jdbc.OracleDriver**. The driver appears in the JDBC Driver field and a Database URL format appears according to the driver you selected. For the Oracle thin driver, the format is:

```
jdbc:oracle:thin:@<host>:<port#>:<db_instancename>
```

Check the product readme file to ensure you are using the correct database and driver versions.

3. In the Database URL field, configure the provided format. For example:

```
jdbc:oracle:thin:@localhost:1521:ORCL
```

where 1521 is the default port, and ORCL is the default instance name for Oracle Database 10g.

4. In the User Name and Password fields, enter the username and password of the database user (see [Task C, Run the Initialize Database Script as the System User, on page 243](#)).

5. Click **Apply**, then click **Test Connection**. If the database is running and the details are correct, you see a success message. (True only if you add the JDBC driver to the project libraries in TIBCO BusinessEvents Studio).

6. Save the resource.

7. Rebuild the EAR file.

Configuring Backing Store Properties in the CDD Editor

This section summarizes CDD configuration related to use of a backing store. It assumes that you are adding backing store support to an existing project configured for cache OM.



You can also control how data is loaded from the backing store to the cache at startup. See [Cluster Tab — Cache OM — Domain Objects Settings on page 46](#) for details.



JDBC backing store requires use of cache-aside write management.

Enabling Backing Store and General Settings

For details on the following, see [Cluster Tab — Cache OM — Backing Store Settings, page 38](#) and related content in [Chapter 3, CDD Configuration Procedures, on page 9](#):

- Enabling or disabling backing store.
- Selecting a backing store implementation and database product.
- Choosing whether to use Oracle strategy, if you choose the Oracle Database product.

Note that the cache-aside method of database write management is enabled internally for JDBC backing store independently of any setting.

Database Connection Settings

For details on the following, see [Chapter 3, CDD Configuration Procedures, on page 9](#):

- The JDBC connection resource to use.
- Whether to enforce pool settings.
- The max and initial pool settings.

- If Oracle Strategy is used then:
 - Min setting is available.
 - Max and initial pool settings use Oracle properties.
 - Additional properties are available. See [Cluster Tab — Cache OM — Backing Store Settings, page 38](#) and [Cluster Tab — Cache OM — Backing Store Properties, page 42](#)



Configuring Exceptions When backing store is enabled, you can configure individual entity types can be set to not use a backing store. To do so use the entity metadata property, `hasBackingStore`. See [Entity-Level Configuration for Cache and Backing Store on page 30](#).

If a Custom Cache Configuration File is Used

If you have customized the cache configuration file, `coherence-cache-config-jdbc.xml`, add the following property to the cluster tab property sheet: `tangosol.coherence.cacheconfig` and provide the filepath and name of the custom file as the value.

To correctly enable JDBC backing store functionality, you must also update your customized file as follows. Change instances of this string:

```
com.tibco.be.oracle.BECoherenceOracleStore
```

To this string:

```
com.tibco.be.jdbcstore.becoherencejdbstore
```

In the standard cache configuration file, you would replace six instances.

Specifying Limited Cache Size

When you use a backing store you may want to use a limited size cache. Use of the limited cache properties require backing store to be used.

If you want to limit the size of the cache, set the `isCacheLimited` property to true.

```
be.engine.cluster.isCacheLimited=true
be.engine.limited.cache.back.size.limit=10000
```

You only need to set the size limit if you want to use a size other than the default.

Backing Store Table Reference

The backing store uses relational tables and SQL data types for ease of maintenance. The SQL (DDL) scripts use ANSI SQL type definitions (where supported by the target DBMS product).

Each ontology type in the backing store has its own primary table and zero or more second-level tables. There are only two levels of tables, which makes the database easier to manage and easier to understand. Because the backing store adheres to SQL standards and a straight-forward structure, standard database tools can be used to view backing store data.

Primary Tables

Primary tables contain only primitive properties such as the following:

Property	Note
cacheId	Entity version number (starts with 1)
time_created\$	Time when the entity was created
time_last_modified\$	Time when the entity was last modified
parent\$_id\$	Id of the parent for contained concepts
id\$	Unique Id of the entity (must be unique across all entities)
extId\$	Unique (or null) extId assigned
state\$	Always set to 'C' meaning 'Created' (reserved for future use)

Secondary Tables

Secondary tables are used for complex properties, that is, arrays, properties with history, and concept relationship properties. Each array and history-enabled property has a separate table. Only primitive properties are stored in the primary table.

Table 47 Secondary table structure

property Type	Column	Description
Array	pid\$	Parent ID
	valPid\$	Array index
	val	Item's value
History	pid\$	Parent ID
	howMany	Number of history items
	timeIdx	Item's time stamp
	val	Item's value
Array with History	pid&	Parent ID
	valPid\$	Array index
	howMany	Number of history items
	timeIdx	Item's time stamp
	val	Item's value

Reverse Reference Tables

Each concept also has a reverse reference table. This table's name contains the concept name and ends with the characters `_rrf$`. It has these columns:

Column	Description
pid\$	Parent ID from the main concept table
propertyName\$	Property name (field) from the referencing concept.
id\$	Identifier (id\$) of the referencing concept.

Class-to-Table Mapping

This table contains the mapping between class names and table names, and the mapping between complex property field names and secondary table names.

for example:

```
'be.gen.Ontology.DeleteVerifyEvent', 'D_DeleteVerifyEvent'  
'be.gen.Ontology.Treatment', 'D_Treatment'  
'be.gen.Ontology.Treatment', 'rrf$', 'D_Treatment_rrf$'  
'be.gen.Ontology.BaseAlert', 'treatments',  
'D_BaseAlert_treatments'
```

Appendix A **Setting up an Oracle-Only Backing Store**

A backing store enables persistent backup of data in the cache. Use of a backing store enables recovery in the event of a system-wide failure.

This appendix covers setting up the backing store database using the Oracle-only backing store feature.

You must also configure backing store settings for the Oracle-only backing store.



The Oracle-only backing store was deprecated in the 4.0 release. If you are setting up a backing store for the first time, it is recommended that you use the current implementation, the JDBC backing store, which works with more DBMS products. It also has a more human-readable schema. See [Chapter 15, JDBC Backing Store Configuration](#), on page 233.

For instructions on migrating data from a 3.x backing store to a newly created JDBC backing store, see [Migrating Data to a Current Backing Store Implementation](#) in *TIBCO BusinessEvents Installation*.

Topics

- [Oracle-Only Backing Store Database Setup Overview](#), page 260
- [Resources Required for Setting Up the Oracle Database](#), page 262
- [Oracle-Only Backing Store Database Configuration Tasks](#), page 264
- [Updating an Existing Oracle Database Schema](#), page 268
- [Project Configuration for Oracle-Only Backing Store](#), page 270
- [Oracle-Only Backing Store Runtime Behavior](#), page 271
- [Handling Deleted Entities](#), page 272

Oracle-Only Backing Store Database Setup Overview

You can implement an Oracle-only backing store for use with any Cache object management option. At system startup, data is loaded into the cache from the backing store. During regular operation, the cache persists the data that is written to it in the backing store. This happens at the end of each RTC.

If you use a limited-size cache, you generally use a backing store so that data evicted from the cache is not lost.

This overview provides summary information about setting up the database, with references to the sections that provide detailed information.

You must also configure various properties to enable and configure backing store functionality.

Oracle-Only Backing Store Requirements

Backing store functionality has been tested with Oracle 10g Enterprise Edition and Oracle Database 10g Express Edition (see the readme file for specific version information). You can download the Express Edition for development use from the Oracle web site.

Instructions in this chapter assume you are working with a local database for testing. For production deployments, you might have to ask a database administrator to create a database user for you. You should then be able to run the other SQL scripts yourself, logged on as the user created by the administrator.

Before You Begin Database Setup

- Develop your caching solution and test it. See [Chapter 3, CDD Configuration Procedures, on page 9](#).
- Ensure that you have access to a supported database management product and can create a user and tables.

Oracle-Only Backing Store Database Setup Tasks

Backing store configuration tasks for each backing store, and background information are summarized below.

- [Resources Required for Setting Up the Oracle Database on page 262](#) outlines all the resources you need to set up a backing store.

- [Oracle-Only Backing Store Database Configuration Tasks on page 264](#) explains several tasks you must complete in order to configure the database schema, including generating SQL scripts for your project and running them to create the database user and tables.

Extra Procedure to Handle Long Database Identifier Names

A known limitation in Oracle means that each identifier name cannot exceed 30 characters in length. If you have longer names, remember to complete [Task C, Shorten Long Names Using the Aliases File, on page 265](#), to give an alias to each long identifier name.

Every entity, property, or state machine whose name exceeds 30 characters in length has an entry in the generated *yourname.aliases* file (For example, *acme.aliases*). This file has no entries if all names have 30 characters or less.

Note that you must run the `be-oradeploy` utility again, after updating the *yourname.aliases* file (as mentioned in the procedure.)

After You Finish Database Setup

You must also configure various properties to enable and configure backing store functionality.

Maintaining a Backing Store — If Ontology Object Definitions Change

If you add, change, or delete ontology object definitions, the backing store schema will no longer match your ontology. You must update the backing store schema so it still matches the ontology. See [Updating an Existing Oracle Database Schema on page 268](#) for details

Resources Required for Setting Up the Oracle Database

The table below lists resources required and sections following explain the procedures for setting up backing store tables.

Table 48 Resources Required for Backing Store Implementation

Resource	Default Location and Purpose
Oracle 10G	This feature has been tested with Oracle 10g Enterprise Edition and Oracle Database 10g Express Edition (see the readme file for specific version information).
Oracle JDBC Driver	Oracle JDBC drivers are not provided. Download the client from the Oracle web site or find the drivers in your Oracle Client installation.
Oracle Thin driver recommended	Copy the JAR files (for example, ojdbc14.jar) to <i>BE_HOME/lib/ext</i> or elsewhere in your class path. Backing store functionality has been tested with Oracle Thin driver.
be-oracle.jar	<i>BE_HOME/lib</i> A JAR file required for backing store functionality.
be-oradeploy.exe be-oradeploy.tra	<i>BE_HOME/bin</i> Executable and property files used to generate SQL scripts (see below).
Provided SQL Scripts	<i>BE_HOME/bin</i>
initialize_database.sql	The initialize_database.sql script drops the user (and therefore all the tables) and adds the user again. By default the user is called be_user with the password be_user and has DBA rights. Edit the script if you want the user to have a different name or different rights.
base_types.sql	The base_types.sql script defines the base types, corresponding to the TIBCO BusinessEvents object data structure.
create_tables.sql	The create_tables.sql script creates the tables that are used to maintain the metadata. It does not drop any existing tables.

Table 48 Resources Required for Backing Store Implementation (Cont'd)

Resource	Default Location and Purpose
Generated SQL Scripts	These scripts are generated when you run the <code>be-oradeploy</code> executable, and they are located in the same directory where you run <code>be-oradeploy</code> .
<code>yourname.sql</code>	
<code>yourname.aliases</code>	You provide the value of <code>yourname</code> when you generate the scripts.
<code>yourname_remove.sql</code>	The <code>yourname.sql</code> script is executed after the provided scripts are executed, as explained in the procedures below. It creates schema tables and types.
<code>yourname_cleanup.sql</code>	
<code>yourname_alter.sql</code>	
<code>yourname_delete.sql</code>	The <code>yourname.aliases</code> script has entries if your project has names longer than 30 characters. You must perform a procedure to provide aliases for long names, and then regenerate the SQL scripts again using the <code>be-oradeploy</code> utility. The procedure is explained in Task C, Shorten Long Names Using the Aliases File .
	The <code>yourname_remove.sql</code> script can be used as needed. It removes the database schema. You can use it to reset the project
	The <code>yourname_cleanup.sql</code> script can be used as needed. It truncates the tables.
	The <code>yourname_alter.sql</code> script is for use in schema migration. Generated only if properties are added to <code>be-oradeploy.tra</code> , to identify the existing database and its user and password. See Updating an Existing Oracle Database Schema on page 268 .
	The <code>yourname_delete.sql</code> script is for use as needed to delete entities from the backing store that have been deleted from the cache. By default deleted entities are removed from the backing store at system startup. See Handling Deleted Entities on page 272 .

Oracle-Only Backing Store Database Configuration Tasks

As with any procedure that modifies your data, ensure that you have made backups before you begin.



Existing Backing Stores: After Ontology Object Definition Changes

See [Updating an Existing Oracle Database Schema on page 268](#) for the impact of different kinds of changes and how you can update the backing store schema.

Task A Prepare Files

1. Open your project in TIBCO Designer, and build the EAR file. Model information in the EAR will be used to build tables in the database.
2. Ensure that `be-oracle.jar` is located in `BE_HOME/lib` (or other location in your class path).
3. Copy your JDBC drivers file to `BE_HOME/lib/ext` (or other location in your class path). These files are part of the Oracle Client software.

Task B Generate the SQL Scripts

Open a command window and navigate to `BE_HOME\bin`. Run `be-oradeploy.exe` using a command with the following format:

```
be-oradeploy [-p property file] [-o Oracle schema output file] [EAR Path] [-h]
```

For example:

```
be-oradeploy -o acme c:/BEProjects/MyEar.ear
```



If you are not running from the default folder (`BE_HOME/bin`) or if you are not using the default files, you must provide both the `--propFile` and the `-p` parameters, and pass them the fully qualified name of the TRA file. For example:

```
be-oradeploy --propFile c:\mypath\myfile.tra -p  
c:\mypath\myfile.tra
```

The options are explained in the following table:

Option	Description
<code>-p, /p,</code> <code>-property, or</code> <code>/property</code>	Specifies the property file. If not specified, the default property file is used, that is, <code>be-oradeploy.tra</code> in the current directory. See note above.

Option	Description
-o	Specifies the Oracle schema output filename for deployment.
-h, /h, or /help	Displays this help.

In the command window, you see various messages as schema definition commands are created in the generated scripts and the scripts are created.

The generated scripts appear in the directory where you ran the executable. For example, if you provided the schema output filename `acme`, you would see files called `acme.sql`, `acme.aliases`, `acme_cleanup.sql`, `acme.delete.sql` and `acme_remove.sql`.

The user-defined part of the database schema is created in `yourname.sql` as schema definition commands. In [Task E](#) you run this script (together with provided scripts) to build the schema in the database.

Task C Shorten Long Names Using the Aliases File

If the aliases file has entries, do the following to provide short aliases for all names longer than 30 characters (see [Extra Procedure to Handle Long Database Identifier Names on page 261](#) for more details):

1. Open the `yourname.aliases` file for editing.
2. For each entity, property, and state machine name that exceeds 30 characters in length, provide an alias using a name that is shorter than 30 characters. Ensure the name is unique.

For example, you would modify the following entry:

```
TABLE.D_NewConceptNewConceptNewConceptNewConcept.alias=
D_NewConceptNewConceptNewConceptNewConcept
```

With a short name such as:

```
TABLE.D_NewConceptNewConceptNewConceptNewConcept.alias=
D_NewConceptNewConceptNewCon
```

3. Perform [Task B, Generate the SQL Scripts](#), again. This time, the aliases you created are used.

Task D Run the Initialize Database Script to Create the Oracle User



Running the `initialize_database.sql` script drops all existing backing store tables. If you have data you need to retain, contact TIBCO support for assistance.

In the `initialize_database.sql` script, the Oracle user is set to `be_user`, with password `be_user`. You can edit the script as needed to change these default settings. The documentation uses the default username and password.

1. Login to Oracle Server as the system user.
2. Navigate to the location of the scripts (or copy them all to the `BE_HOME/bin` directory) and open an SQLPlus prompt. (Open a command window, type `SQLPlus` then provide the `system` user credentials.)
3. Type `@initialize_database.sql` to run the provided script, `initialize_database.sql`. You see messages like the following:

```
DROP USER be_user CASCADE
      *
ERROR at line 1:
ORA-01918: user 'BE_USER' does not exist

User created.
Grant succeeded.
SQL>
```

Note that the script assumes that it has been run before. It deletes the user before creating it again. This means you can run the script again without having to take extra steps. This is useful for testing purposes.



Using your database product, you can configure additional users to access the database, in addition to this user.

Task E Login as the Oracle User and Run SQL Scripts

In this step, you run scripts to create the database schema under the user you created. The schema combines the definitions in `base-types.xml`, `create-tables.sql`, and the generated schema output file (`acme.sql` as an example).

Note that these scripts also perform cleanup before creating the schema. The first time you run the scripts, you see harmless error or warning messages because there is nothing to delete.

1. Login to the Oracle server as `be_user`, password `be_user` (or whatever username and password you set in the script in [Task D](#)).
2. Navigate to the location of the scripts and open an SQLPlus prompt. Identify yourself as `be_user` with password `be_user`.

3. At the SQL prompt, type the following to run each script in turn:
 - a. `@base_types.sql`
 - b. `@create_tables.sql`
 - c. `@yourname.sql` (for example, `@acme.sql`)

Your database tables are now configured for use.

Updating an Existing Oracle Database Schema

If you change the project ontology, that is, if you create, alter or delete a concept or an event, you must update the backing store schema so it matches the updated ontology. You must do this before you deploy the updated project.

Not all changes can be automatically migrated. Manual migration is required for such changes.

What the Schema Migration Utility Can and Can't Handle Automatically

The migration utility handles the following:

- Addition of entity types and attributes. New entity types and attributes are added to the database schema.
- Deletion of entity types and attributes. Deleted entity types and attributes are dropped from the database schema.

The utility can handle only certain changes to existing entities and attributes, depending on the datatype and on Oracle functionality. The utility does not handle changes to the data type of an existing attribute, for example, changing a String attribute to a number attribute.



If the utility encounters any change in the schema that cannot be migrated automatically, then the migration script is not generated. In this case, migrate the data and the schema manually. Contact TIBCO support for assistance about handling of specific changes.

To Update an Existing Backing Store Database Schema

Before you begin:

- Gracefully shut down the deployed application (all agents and cache servers).
 - Back up your existing database.
1. Generate the updated EAR file for the modified project.
 2. Save existing copies of *yourname.sql* and *yourname.aliases* so you can compare them with the files you will generate for the changed project.
 3. Open the `be-oradeploy.tra` file for editing and set the following properties:

```
be.oracle.schemamigration.url=DbURL
be.oracle.schemamigration.user=username
be.oracle.schemamigration.pswd=password
```

Where the username and password are those you set up in [Task D, Run the Initialize Database Script to Create the Oracle User, on page 265](#).

The properties enable the program to compare the schema of the existing database with the schema of the project EAR file, to generate the schema alteration script.

4. Run the `be-oradeploy.exe` utility as explained in [Oracle-Only Backing Store Database Configuration Tasks on page 264, Task B, Generate the SQL Scripts, using the updated EAR file](#).
5. If any of the new or changed definitions result in entries in the `yourname.aliases` file, follow instructions in [Task C, Shorten Long Names Using the Aliases File, on page 265](#). You must use the same aliases again for entity types (definitions) that used aliases before. Remember to generate the scripts again, as instructed in [Task C](#).
6. Run the newly-generated script `yourname_alter.sql` script.

Your database tables are now configured for use.

Project Configuration for Oracle-Only Backing Store

Project configuration of an Oracle-Only backing store is similar to project configuration for a JDBC backing store. You must add a JDBC Connection resource with details for your database connection, and configure various settings in the CDD editor. See the following for details:

- [Adding a JDBC Connection Resource to the Studio Project, page 252](#)
- [Configuring Backing Store Properties in the CDD Editor, page 253](#)

Oracle-Only Backing Store Runtime Behavior

During regular operation, cache data is written to the backing store by the cache cluster.

With write-behind database writes management, only cache servers write to the backing store. Inference agents and query agents are not involved in writes to the database.

At system startup, the data is loaded into the cache from the backing store, using any one of the nodes (including nodes running inference agents and query agents).

Options to configure these behaviors are explained next.

Reading From the Backing Store at Startup

At system startup, after the minimum number of cache servers have started, the cache is loaded with backing store data. This occurs before the agents begin to process events or execute startup functions.



Any node in the cluster can load data from the backing store to the cache at startup.

You can configure the system so that only certain objects are loaded into the cache from the backing store at startup. See [Cluster Tab — Cache OM — Domain Objects Settings on page 46](#).

Reading From the Backing Store At Runtime

At runtime, if an object is not in working memory it is requested from the cache. If it is not in the cache, the cache server retrieves that object from the backing store.

At runtime, only the cache servers interact with the backing store.

Writing to the Backing Store

After an RTC, data is written to the cache, and then the cache writes the data to the backing store.

Handling Deleted Entities

When entities are deleted from the cache, they are added to a script that is run at system startup. The script removes the deleted entities from the backing store.

Depending on the number of deleted entities, the cleanup step can make system startup unacceptably slow.

Manual Deletion of Deleted Entities

To deal with slow startup, you can run the `yourname_delete.sql` script manually, to reduce the number of entities that have to be processed at startup. This script is generated by the `be-oradeploy` utility. See [Generated SQL Scripts on page 263](#) for a list of scripts generated by the utility.

Disabling Deleted Entity Cleanup at System Startup

You can also disable the cleanup action performed at system startup and perform all cleanup manually, using the deletion script.

Add the following property to the TRA file and set it to false:

```
be.engine.cluster.cleanup
```

Appendix B Determining the Engine Name

When establishing the engine name, TIBCO BusinessEvents software searches for a value from one of the settings in the order shown, accepting the first value it finds:

1. For deployment using MM, the name specified in the Processing Unit Configuration Name field in the site topology file. See [Site Topology Reference on page 100](#).
2. 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.
3. 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.
4. 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.
5. The engine name set in the CDD file, in the Processing Unit tab Name field.
6. The name of the TIBCO Hawk microagent instance. This name exists if TIBCO Hawk is enabled at runtime. The microagent name can also be set in the `be-engine.tra` file using the property `Hawk.AMI.DisplayName`.
7. The host name.
8. This string: `engine`.

Appendix C Engine Startup and Shutdown Sequence

This section outlines the main actions that occur during engine startup and shutdown (in normal circumstances). In any particular project only some of the actions may be required. For example, a project may have no startup rule functions.

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

Startup Sequence



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 local storage enabled.)

In a situation where all cache agents are stopped but engines running other types of agents are running, you must restart all engines.

During engine startup the following actions occur:

1. System information displays (in consoles) and is recorded in the log file:
 - The property file and EAR file that were used to start the engine.
 - The version of the JAR files it is using, and the version of the JAR files that the EAR file was built with.
 - If persistence OM is used, the location of the Berkeley DB software it is using, and information about what was recovered from the database.
2. Cache OM with backing store only: Recovery stage. When the minimum number of cache servers 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 Recovery state ends, the cluster enters Ready state.
3. All inference agents build their Rete networks by evaluating conditions against all Cache Plus Memory objects (if any).

(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:
 - The clock starts for repeating time events and they are created and asserted at the specified intervals.
 - 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. Finally, inbound channel listeners activate and accept incoming events and the system is now fully started up.

Shutdown Sequence

During engine shutdown the following main actions occur:

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

System Startup and Shutdown Sequence

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

- Start engines running cache agents before starting other engines.
- Stop other engines before stopping engines running cache agents.

Appendix D **Advanced Caching Topics**

This appendix provides details about caching topics that are useful only in some circumstances or provide background information for interested readers.

Topics

- *Overriding and Extending the Operational Deployment Descriptor, page 278*
- *Specifying Operational Override File Locations, page 280*
- *Understanding Entity Caches, page 283*

Overriding and Extending the Operational Deployment Descriptor

The operational deployment descriptor is called `tangosol-coherence.xml`. This file is provided in `BE_HOME/lib/ext/coherence.jar`. In order to configure certain cache-related settings, you may need to override the values of certain elements in the operational deployment descriptor. This is explained in [Overriding Element Values in Engine Property Files on page 278](#).

You may also need to define additional elements, and system properties for existing elements that lack them.

Overriding Element Values in Engine Property Files

TIBCO BusinessEvents overrides the values of elements in the operational descriptor using their `system-property` attributes. The `system-property` attribute values are used in Java command line options, and those options are included in some standard TIBCO BusinessEvents engine properties.

For example, the `system-property` attribute assigned to the element `<cluster-name>` is `tangosol.coherence.cluster`. TIBCO BusinessEvents uses this attribute to specify a value for `<cluster-name>` in the engine property (TRA) files as follows:

```
java.property.tangosol.coherence.cluster myclustername
```

System properties are predefined for many operational elements. You can override the value of any element that has a `system-property` attribute by adding a Java command line option in the engine property files, as shown in the example above.

For detailed documentation of this feature, and a list of elements for which `system-property` attributes are predefined, see the section "Line Setting Override Feature" in the online reference, *TIBCO BusinessEvents Cache Configuration Guide*.

Defining Additional Elements and System Properties in Override Files

If there is no `system-property` attribute for a setting you want to override, or if you need to add more elements than are available in the operational deployment descriptor, you must create an override file (or a series of override files) to contain the `system-property` attributes and any new elements you require. For example, you might add more elements to provide well-known addresses for the servers in a cache cluster.

You can store default values in an override file, and you can also create engine properties (using the `system-property` attributes) that override those file values. The settings you enter in engine property files override settings in the override file, and settings in the override file override those in the operational descriptor.

Override file elements that also exist in the operational descriptor must use the same structure as the elements in the operational descriptor.

Example: Adding Well Known Address Attributes

You would need to add elements if you use the well-known address method of defining a cache cluster and need more than six well-known addresses. (Six well-known addresses are declared in the default file, `tangosol-coherence.xml`.) Here is an example definition:

```
<well-known-addresses>
  <socket-address id="1">
    <address system-property="tangosol.coherence.wka"></address>
    <port system-property="tangosol.coherence.wka.port">8088</port>
  </socket-address>
</well-known-addresses>
```

In the override file, for example, `tangosol-coherence-override-prod.xml`, you add similar entries to the one provided in `tangosol-coherence.xml`, each specifying a unique `system-property` setting. For example:

```
<well-known-addresses>
  <socket-address id="7">
    <address system-property="tangosol.coherence.wka7"></address>
    <port system-property="tangosol.coherence.wka.port">8089</port>
  </socket-address>
  <socket-address id="8">
    <address system-property="tangosol.coherence.wka8"></address>
    <port system-property="tangosol.coherence.wka1.port">8089</port>
  </socket-address>
</well-known-addresses>
```

You can then use the `system-property` values in engine properties, and (in the case of the example shown) define the well known address values.

Specifying Operational Override File Locations

As explained in [Overriding and Extending the Operational Deployment Descriptor on page 278](#), in order to define additional elements or add system properties for existing elements, you use an override file.

You can actually use two or more tiers of override files depending on your needs:

- The first tier settings override and extend the operational deployment descriptor
- The second tier settings override and extend the first tier overrides.
- More tiers can be added as needed, though this is unlikely.

The reason for using multiple tiers is to enable all (or many) nodes to use the first tier overrides, while providing additional second tier overrides to selected nodes. For example, you may want to enable verbose logging on only one or two nodes for diagnostic purposes. To do so you would provide those settings in a second tier override file that you then reference in the selected nodes' engine property files.

How the First Tier Override File Default Location is Specified

The operational descriptor file, `tangosol-coherence.xml` (located in `BE_HOME/lib/ext/coherence.jar`) contains an `xml-override` property configured to point to a default override file:

```
<coherence xml-override="{tangosol.coherence.override
/tangosol-coherence-override-{mode}.xml}">
```

The structure of the property is:

```
xml-override={property default-property-value}
```

where *property* is the name of the override property, and *default-property-value* is the filepath and name of the file.

How TIBCO BusinessEvents Uses the Override Property

The object management layer looks for the specified override file in the classpath.

The default location of the file specified by `tangosol.coherence.override` is a file at the root level of the class path. TIBCO BusinessEvents uses the first instance of this file that it finds at the root level of the classpath. For example, if you put an instance of this file at the root level of a different JAR file that is located closer to the beginning of the classpath, then that file is used instead of the one in `coherence.jar`.

If a user overrides the default override location, TIBCO BusinessEvents looks for the file specified in the same way.

Optional User-Defined Property for Flexibility in the Filename

You can use a user-defined property such as `{mode}` to define name patterns that provide flexibility. You can place multiple files that use the name pattern in the specified override location. For example, each of the following has a different value for `{mode}`: `tangosol-coherence-override-dev.xml`, `tangosol-coherence-override-test.xml`, and `tangosol-coherence-override-prod.xml` files. Then you can easily switch between files that have values appropriate for those environments at engine start-up.

You specify the value to use for the user-defined property in your TRA file, or at the command line.

For example, using `{mode}` as the user defined property, to specify the value in the TRA file, use:

```
java.property.mode Mode_Value
```

To override the value at the command line, use

```
-Dmode Mode_Value
```

Where (still using the example defined above) the *Mode_Value* is one of `test`, `prod`, or `dev`.

The default value for `{mode}` is `prod` (defined in the `tangosol-coherence.xml` descriptor). Therefore the default value for the first tier override file is `/tangosol-coherence-override-prod.xml`.

How to Specify a Different Location for the First Tier Override File

To override the default location of the first tier override file, you provide a value for the `tangosol.coherence.override` property either at the command line or using an engine property. You can do this one time only, on engine startup. The value and file contents must be the same on all nodes in a cluster.

The value of the property can be a file path or a JAR URL. It must be a location in the classpath.

To specify the override using an engine property, add the property name (specified in the `xml-override` property) and its value to the TRA files. All nodes in a cache cluster must specify the same file. For example:

```
java.property.tangosol.coherence.override=file:/c:/tmp/my_tangosol-coherence-override.xml
```

For URL locations inside any JAR, specify the path as in the following example:

```
java.property.tangosol.coherence.override=file:/home/jsmith/tmp/client/lib/coherence.jar!/my_tangosol-coherence.override.xml
```

You can alternatively specify the override as a system property, that is, a command line parameter at engine start-up. For example,:

```
-Dtangosol.coherence.override=file:/C:/tmp/my_tc-override.xml
```

How a Second Tier Override File Default Location is Specified and Overridden

The first tier override file can itself specify the default location and name of a second tier override file, again using the `xml-override` property:

```
xml-override={property default-property-value}
```

You would configure the property using different values, but the mechanism is the same. For example, you might use the property name `be.coherence.override`. You can also use another user property for part of the filename, if you want to provide that flexibility, for example:

```
<coherence xml-override="{be.coherence.override  
/be-coherence-override-{be.coherence.environment}.xml}"
```

As explained above, if you want to specify a file of this pattern, for example, `be-coherence-override-dev`, you could specify the value in the TRA files or at the command line on engine start-up. Here is an example showing how the system property value is defined at the command line:

```
-Dbe.coherence.environment=dev
```

Similarly, if you want to specify a different location and filename for the second-tier override file, you would use the `be.coherence.override` property, for example:

```
-Dbe.coherence.override=file:C:/tmp/tango-coherence-override.xml
```

Understanding Entity Caches

This appendix is provided for those who want to understand the internal structures of the caches used in Cache object management. This information is not required for configuration tasks.

For each entity in working memory, a corresponding cache exists in the cache cluster. Internal entities also have caches for various purposes, explained in this section.

Entity Cache Names Format

Each entity cache has a name, which uses the following format:

cache-type.cluster-name.AgentClassName.entity-name

The elements of the above name are explained below

Cache Type (Caching Scheme)

Cache type is the type of caching scheme (as defined by its cache name in the `coherence-cache-config.xml` descriptor), for example, `dist-unlimited-bs`.

Cluster Name

Cluster name is the value of the following property:

```
java.property.tangosol.coherence.cluster
```

Agent Name

This field of the cache name is blank because TIBCO BusinessEvents does not support agent-specific entity caches.

All entities are globally scoped and available to all agents.

Entity Name

Two types of entities have caches:

- Internal entities
- Ontology entities

Internal entity names and caches are listed and described in [Table 49, Internal Entity Caches, on page 284](#).

The ontology entity field of the entity cache name uses the entity's generated class name, which is similar to its design-time folder path and name, prefixed by `be.gen`. For example:

```
be.gen.Concepts.LargeConcepts.ThisLargeConcept
```

Caches for Ontology Objects

These caches are used to store the objects of types defined in the ontology of the project.

The types of caches created for ontology objects depend on the caching scheme used. If the `dist-unlimited-bs` caching scheme is used, then the cache names look like this:

```
dist-unlimited-bs$foo$be.gen.Order
```

Where `foo` is the cluster name.

Caches for Internal Entities

The following internal caches use a pre-defined scheme in the cache configuration file. Do not change this scheme. This information is provided for reference only.

Table 49 Internal Entity Caches

Entity (Cache) name	Purpose of the Cache
Master	Maintains the cluster state and is shared by all nodes.
Catalog	Maintains a cached copy of all ontology definitions shared by all nodes.
TypeIDs	Stores the mapping between type IDs and class names. All ontology objects are tagged with a unique integer ID. Use of IDs avoids the need to serialize and send class name strings between nodes.
ObjectTableIDs	Stores the key mapping for all objects in the cluster. The objects themselves are stored in their respective caches.
ObjectTableExtIDs	Stores the external key mapping for all objects that have an external ID (<code>extId</code>).
AgentTable	Stores all the agents and their respective states across all cluster nodes and identifies the currently active and standby nodes.

Table 49 Internal Entity Caches

Entity (Cache) name	Purpose of the Cache
<i>AgentTxn-agentId</i>	Each agent in the cluster has an <i>AgentTxn-agentId</i> cache. The <i>agentId</i> is internally generated. It stores the change list for the agent. The change list is used to replicate changes between active-active and active-passive sets of agents in the cluster so that they stay synchronized.
TimeQueue	Maintains all entries that are time bound, for example, state machines that can have timeouts at a state machine level or at a state level. This cache maintains an index to all objects that must be re-evaluated after a certain period of time.

Appendix E TIBCO Hawk Microagent Methods

TIBCO Administrator is the preferred monitoring and management application for TIBCO BusinessEvents. However, the TIBCO BusinessEvents engine is instrumented with a TIBCO Hawk microagent that can be used to perform many administrative functions. This appendix describes the microagent methods available for the TIBCO BusinessEvents engine.

Topics

- [TIBCO Hawk Methods Overview, page 289](#)
- [activateRuleSet\(\), page 290](#)
- [activateTraceRole\(\), page 291](#)
- [deactivateRuleSet\(\), page 292](#)
- [deactivateTraceRole\(\), page 293](#)
- [execute\(\), page 294](#)
- [forceOMCheckpoint\(\), page 295](#)
- [getChannels\(\), page 296](#)
- [getDestinations\(\), page 297](#)
- [getEvent\(\), page 298](#)
- [GetExecInfo\(\), page 299](#)
- [getHostInformation\(\), page 300](#)
- [getInstance\(\), page 301](#)
- [getMemoryUsage\(\), page 302](#)
- [getNumberOfEvents\(\), page 303](#)
- [getNumberOfInstances\(\), page 304](#)
- [getOMInfo\(\), page 305](#)
- [getRuleSet\(\), page 306](#)

- [*getRuleSets\(\)*, page 307](#)
- [*getScorecard\(\)*, page 308](#)
- [*getScorecards\(\)*, page 309](#)
- [*getSessionInputDestinations\(\)*, page 310](#)
- [*getSessions\(\)*, page 311](#)
- [*getStatus\(\)*, page 312](#)
- [*getTotalNumberRulesFired\(\)*, page 313](#)
- [*getTraceSinks\(\)*, page 314](#)
- [*reconnectChannels\(\)*, page 315](#)
- [*resetTotalNumberRulesFired\(\)*, page 316](#)
- [*resumeChannels\(\)*, page 317](#)
- [*resumeDestinations\(\)*, page 318](#)
- [*stopApplicationInstance\(\)*, page 319](#)
- [*suspendChannels\(\)*, page 320](#)
- [*suspendDestinations\(\)*, page 321](#)

TIBCO Hawk Methods Overview

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.

Types of Methods

The methods documented in this appendix are provided for the following purposes:

- 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, `getRuleSets()`, `getDestinations()`, `getTotalNumberRulesFired()`
- To make certain changes in the TIBCO BusinessEvents engine without stopping it, for example, `activateRuleSet()`, `forceOMCheckpoint()`, `reconnectChannels()`

Enabling the TIBCO Hawk Microagent

Before using the Hawk methods, you must enable the TIBCO Hawk microagent in the TIBCO BusinessEvents engine property file, `be-engine.tra`. To do this, set the `Hawk.Enabled` property to true and ensure that it is uncommented.

If you are using non-default transport parameters for TIBCO Hawk, you must also set the `repo.hawkDaemon`, `repo.hawkNetwork`, and `repo.hawkService` properties to the values for the transport you are using.

For More Information

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.

activateRuleSet()

Purpose 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	Is the RuleSet activated?

activateTraceRole()

Purpose Enable a Trace Role

Type ACTION

Parameters

Name	Description
Role Name	Name of a Role

Returns Returns nothing.

deactivateRuleSet()

Purpose 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?

deactivateTraceRole()

Purpose Disable a Trace Role

Type ACTION

Parameters

Name	Description
Role Name	Name of a Role

Returns Returns nothing.

execute()

Purpose 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.

forceOMCheckpoint()

Purpose Forces an Object Store checkpoint of a Session.

Type ACTION

Parameters

Name	Description
Session	Name of the Session (agent)

Returns Returns nothing.

getChannels()

Purpose 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

getDestinations()

Purpose 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.
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()

Purpose 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()

Purpose 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	Project version

getHostInformation()

Purpose 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()

Purpose 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.

getMemoryUsage()

Purpose 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.
PercentUsed	Estimate of the percentage of max memory used.

getNumberOfEvents()

Purpose Get 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()

Purpose Get 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()

Purpose 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.

getRuleSet()

Purpose 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
Rule	Name of the Rule
Priority	Priority of the rule.

getRuleSets()

Purpose Retrieves a 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()

Purpose 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()

Purpose 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()

Purpose 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()

Purpose Retrieves session names.

Type INFO

Parameters No parameters.

Returns

Type	Description
Line	Line number.
Session	Name of the Session.

getStatus()

Purpose Retrieves basic status information about the engine.

Type INFO

Parameters No parameters.

Returns

Type	Description
Instance ID	Instance ID of the application.
Application Name	Name of the application.
Uptime	Time elapsed since startup.
Process ID	Process ID of the application.
Host	Name of host machine on which this application is running.

getTotalNumberRulesFired()

Purpose 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()

Purpose 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()

Purpose 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()

Purpose Resets the total number of rules fired to zero.

Type ACTION

Parameters

Name	Description
Session	Name of the Session

Returns Returns nothing.

resumeChannels()

Purpose Resumes channels.

Type ACTION

Parameters

Name	Description
URI	URI of the Channel to resume (optional).

Returns Returns nothing.

resumeDestinations()

Purpose 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.

stopApplicationInstance()

Purpose	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.

suspendChannels()

Purpose Suspend channels.

Type ACTION

Parameters

Name	Description
URI	URI of the Channel to suspend (optional).

Returns Returns nothing.

suspendDestinations()

Purpose 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.

Index of Engine Properties

A

[Agent.AgentClassName.cacheTxn.updateCache](#) 67
[Agent.AgentClassName.checkDuplicates](#) 66
[Agent.AgentClassName.dbOpsBatchSize](#) 68
[Agent.AgentClassName.dbOpsQueueSize](#) 68
[Agent.AgentClassName.dbthreadcount](#) 68
[Agent.AgentClassName.maxActive](#) 66, 191
[Agent.AgentClassName.recoveryPageSize](#) 66
[Agent.AgentClassName.threadcount](#) 67

B

[be.agent.query.localcache.evictseconds](#) 65
[be.agent.query.localcache.maxelements](#) 65
[be.agent.query.localcache.prefetchaggressive](#) 68
[be.backingstore.commitSize](#) 43
[be.backingstore.dburi.pool.inactivityTimeout.0](#) 42
[be.backingstore.dburi.pool.waitTimeout.0](#) 42
[be.backingstore.dburi.schema.0 \(legacy\)](#) 44
[be.backingstore.readtimeout.0](#) 43
[be.engine.cacheServer.channel.disable](#) 69
[be.engine.cluster.cleanup](#) 43, 272
[be.engine.cluster.isCacheLimited](#) 44
[be.engine.limited.cache.back.size.limit](#) 38, 45, 45
[be.engine.name](#) 273
[be.engine.om.berkeleydb.cacheweight.agent](#) 81, 82
[be.engine.om.berkeleydb.dbenv](#) 83
[be.engine.om.berkeleydb.internalcachepersent](#) 81
[be.engine.om.eventcache.defaultmaxsize](#) 82
[be.engine.om.eventcache.maxsize.rule_session](#) 83
[be.jdbc.schemamigration.pswd](#) 250
[be.jdbc.schemamigration.url](#) 250
[be.jdbc.schemamigration.user](#) 250
[be.jms.reconnect.msgCodes](#) 76
[be.jms.reconnect.timeout](#) 75
[be.network.mode.standalone](#) 69

[be.oracle.schemamigration.pswd](#) 268
[be.oracle.schemamigration.url](#) 268
[be.oracle.schemamigration.user](#) 268

C

[com.tibco.cep.runtime.scheduler.default.numThreads](#) 57
[com.tibco.cep.runtime.scheduler.queueSize](#) 57, 58
[com.tibco.tibjms.connect.attempts](#) 75

J

[java.property.be.auth.file.location](#) 221
[java.property.be.auth.ldap.adminDN](#) 221
[java.property.be.auth.ldap.adminPassword](#) 222
[java.property.be.auth.ldap.baseDN](#) 222
[java.property.be.auth.ldap.roleAttr](#) 222
[java.property.jdbcdeploy.bootstrap.basetype.file](#) 243
[java.property.jdbcdeploy.bootstrap.keyword.file](#) 243
[java.property.jdbcdeploy.database.type](#) 243
[java.property.mode](#) 281
[java.property.tangosol.coherence.distributed.threads](#) 69, 215
[java.property.tangosol.coherence.override](#) 281
[java.security.auth.login.config](#) 219, 221

T

[tangosol.coherence.cacheconfig](#) 55, 254
[tangosol.coherence.clusteraddress](#) 51, 52
[tangosol.coherence.clusterport](#) 51, 52
[tangosol.coherence.distributed.backupcount](#) 16, 50

tangosol.coherence.localhost [53](#)
tangosol.coherence.localport [53](#)
tangosol.coherence.log [72](#)
tangosol.coherence.log.level [73](#)
tangosol.coherence.log.limit [73](#)
tangosol.coherence.ttl [51](#), [52](#)
tangosol.coherence.wkan [19](#), [54](#)
tangosol.coherence.wkan.port [19](#), [54](#)
tibco.clientVar.HealthCheckFreq [113](#)
tibco.clientVar.JMXUpdateFreq [113](#)
tibco.clientVar.SweepFreq [113](#)
tibco.clientVar.SweepThreshold [114](#)
tibco.clientVar.TopologyUpdateFreq [112](#)

Index

A

Action Configuration Reference 143
 activateRuleSet() 290
 activateTraceRole() 291
 Adding a Cluster Deployment Descriptor 14
 Adding a JDBC Connection Resource to the Studio Project 252
 Additional Configuration Notes 83
 Advantages 207, 208, 208, 210, 211, 213, 214
 After You Finish Database Setup 261
 Agent Classes Tab 13
 agent key, and BEMM 71
 Agent Overview 170
 Agent.AgentClassName.cacheTxn.updateCache 67
 Agent.AgentClassName.checkDuplicates 66
 Agent.AgentClassName.dbOpsBatchSize 68
 Agent.AgentClassName.dbOpsQueueSize 68
 Agent.AgentClassName.dbthreadcount 68
 Agent.AgentClassName.maxActive 66, 191
 Agent.AgentClassName.recoveryPageSize 66
 Agent.AgentClassName.threadcount 67
 Agents Tab Settings and Properties 64
 Alert Configuration Reference 123
 Allow everything and specify exceptions 224
 AppManage 188
 Authentication Options 218
 Authentication Property Reference 221

B

Backing Store 17

backing store
 cache aside 39
 cache loader class 38
 enabled setting 38
 Initial Size (backing store connection pool) 41
 Max Size (backing store connection pool) 40
 Min Size (backing store connection pool) 40
 strategy (for Oracle DB in JDBC backing store) 39
 type 39
 URI 40
 backing store (legacy)
 deleted entities cleanup options 43, 272
 Backing Store Table Reference 255
 base_types.xml 238
 BE_HOME xxiv
 be.agent.query.localcache.prefetchaggressive 68
 be.backingstore.commitSize 43
 be.backingstore.dburi.pool.inactivityTimeout.0 42
 be.backingstore.dburi.pool.waitTimeout.0 42
 be.backingstore.dburi.schema.0 (legacy) 44
 be.backingstore.readtimeout.0 43
 be.engine.cacheServer.channel.disable 69
 be.engine.cluster.cleanup 43
 be.engine.cluster.cleanup (legacy) 272
 be.engine.cluster.isCacheLimited 38, 44
 be.engine.limited.cache.back.size.limit 38, 45, 45
 be.engine.om.berkeleydb.cacheweight.agent 81, 82
 be.engine.om.berkeleydb.dbenv 83
 be.engine.om.berkeleydb.internalcachepersent 81
 be.engine.om.eventcache.defaultmaxsize 82
 be.engine.om.eventcache.maxsize.rule_session 83
 be.jms.reconnect.msgCodes 76
 be.jms.reconnect.timeout 75
 be.network.mode.standalone 69
 Before Deploying a Project in a TIBCO Administrator Domain 190
 Before You Begin Database Setup 260
 BEMM, and agent key 71
 be-oradeploy (legacy) 262

C

Cache Agent Quorum 37
 Cache Aside 39
 cache clusters
 localport auto incrementing feature 53, 54, 55
 Cache Loader Class 38
 Cache-aside 212
 Caches for Internal Entities 284
 Caches for Ontology Objects 284
 Caches Used for Persistence-Based Object Management 84
 Caller's Thread 207
 changes from the previous release of TIBCO BusinessEvents Administration xviii
 checkpoint interval (persistence-based object management setting)
 recommendations 83
 Checkpoint Interval and Outstanding Database Operations 83
 Class-to-Table Mapping 257
 Cluster Discovery 17
 Cluster Explorer 158
 Cluster Member Paths 130
 Cluster Overview 165
 Cluster Overview Pane 165
 Cluster Settings 100
 Cluster Tab 12
 Cluster Tab — Berkeley DB Manager Settings and Properties 79
 Cluster Tab — Cache OM — Backing Store Properties 42
 Cluster Tab — Cache OM — Backing Store Settings 38
 Cluster Tab — Cache OM — Coherence Properties 50
 Cluster Tab — Cache OM — Configuration Settings 37
 Cluster Tab — Cache OM — Domain Objects Settings 46
 Cluster Tab — General Settings 36
 Collections Tab 12
 Collections Tab — Destinations Settings and Properties 56
 Collections Tab — Log Configurations Overview 59
 Collections Tab — Log Configurations Settings 61
 com.tibco.cep.runtime.scheduler.default.numThreads

57

com.tibco.cep.runtime.scheduler.queueSize 58
 com.tibco.tibjms.connect.attempts 75
 Concurrent RTC 210
 Configuring a Cache OM Cluster — Cluster Tab 16
 Configuring Access Control—Overview 224
 Configuring Actions 142
 Configuring Alerts 121
 Configuring an In Memory Cluster 15
 Configuring Backing Store Properties in the CDD Editor 253
 Configuring Collections 21
 Configuring File-Based Authentication 219
 Configuring Health Metric Rules 135
 Configuring Log Configurations 23
 Configuring MM Console Properties 117
 Configuring the Agent Classes Tab (All Object Managers) 24
 Configuring the Cluster Tab for Berkeley DB Manager 78
 Configuring the Collections Tab (All Object Managers) 21
 Configuring the Processing Units Tab (All Object Managers) 27
 Configuring the Site Topology 97
 Configuring User Authentication 219
 Creating and Modifying an Access Control File 227
 customer support xxvii

D

Database Concepts 18
 Database Connection Settings 253
 Database Location 237
 dbkeywordmap.xml (for JDBC backing store) 239
 DBMS Software and Installation Requirements 237
 deactivateRuleSet() 292
 deactivateTraceRole() 293
 Dedicated Workers 208
 Default Level Settings 46
 Defining Additional Elements and System Properties in Override Files 278
 Defining Permissions 228

- Defining Permissions for Domain Models [229](#)
- Defining the Database Directories for Each Rule Session (Inference Agent) [84](#)
- deleted entities (legacy) [43](#), [272](#)
- Deny everything and specify exceptions [225](#)
- Deploying a Project in a TIBCO Administrator Domain [191](#)
- deployment
 - hot [196](#)
 - in TIBCO Administrator [191](#)
 - scripted [188](#)
- Deployment in a TIBCO Administrator Domain [188](#)
- Deployment Overview [184](#)
- Deployment Unit Settings [102](#)
- deploy-time configuration overview [184](#)
- disabling hot deployment [199](#)
- Disadvantages [207](#), [208](#), [209](#), [210](#), [211](#), [214](#), [214](#)
- Domain Objects [18](#)
- DOMAINMODEL Permission Type [229](#)
- domains [188](#)

E

- EAR files
 - deploying [191](#)
- Enable Hot Deployment (As Needed) [200](#)
- Enabled (backing store) [38](#)
- Enabling and Disabling Hot Deployment [199](#)
- Enabling Backing Store and General Settings [253](#)
- enabling hot deployment [199](#)
- Enabling TIBCO Hawk Microagent [289](#)
- encoding [188](#)
- engine name, how determined at runtime [273](#)
- engine property files
 - enabling hot deployment in [199](#)
- Entity Cache Names Format [283](#)
- Entity Level Object Overrides and Settings [48](#)
- Entity Metadata Settings [13](#)
- Entity-Level Configuration for Cache and Backing Store [30](#)
- ENV_HOME [xxiv](#)
- Establishing the User Roles [224](#)
- Event Preprocessor and Rete Worker Thread

- Options [206](#)
- Example
 - Adding Well Known Address Attributes [279](#)
- Example Alter Script [249](#)
- Examples Using Alerts [134](#)
- Examples Using Child Cluster Member Health Metrics [132](#)
- execute() [294](#)
- Extra Procedure to Handle Long Database Identifier Names [261](#)

F

- For More Information [289](#)
- forceOMCheckpoint() [295](#)
- functions documentation, accessing [xxi](#)

G

- getChannels() [296](#)
- getDestinations() [297](#)
- getEvent() [298](#)
- GetExecInfo() [299](#)
- getHostInformation() [300](#)
- getInstance() [301](#)
- getMemoryUsage() [302](#)
- getNumberOfEvents() [303](#)
- getNumberOfInstances() [304](#)
- getOMInfo() [305](#)
- getRuleSet() [306](#)
- getRuleSets() [307](#)
- getScorecard() [308](#)
- getScorecards() [309](#)
- getSessionInputDestinations() [310](#)
- getSessions() [311](#)
- getStatus() [312](#)
- getTotalNumberRulesFired() [313](#)
- getTraceSinks() [314](#)
- Guidelines for Configuring Access Control [224](#)

H

Handling Deleted Entities [272](#)
 Health Indicators and Alerts [157](#)
 Health Metric Rule Configuration Reference [138](#)
 Health Metric Rule Examples [132](#)
 Host Settings [104](#)
 hot deployment [196](#)
 enabling and disabling [199](#)
 limited modifications allowed with [197](#)
 Hot Deployment Overview [196](#)
 How a Second Tier Override File Default Location is Specified and Overridden [282](#)
 How BusinessEvents Uses the Override Property [280](#)
 How the First Tier Override File Default Location is Specified [280](#)
 How to Specify a Different Location for the First Tier Override File [281](#)

I

If a Custom Cache Configuration File is Used [254](#)
 Inactive Members [159](#)
 Inference Agent Overview [173](#)
 Initial Size (backing store connection pool) [41](#)
 initialize_database.sql [244](#)
 initialize_database.sql (for JDBC backing store) [239](#)
 initialize_database.sql (legacy) [266](#)
 Introduction [2, 2](#)

J

java.property.be.auth.file.location [221](#)
 java.property.be.auth.ldap.adminDN [221](#)
 java.property.be.auth.ldap.adminPassword [222](#)
 java.property.be.auth.ldap.baseDN [222](#)
 java.property.be.auth.ldap.host [221](#)
 java.property.be.auth.ldap.port [221](#)
 java.property.be.auth.ldap.roleAttr [222](#)
 java.property.be.auth.type [221](#)
 java.property.tangosol.coherence.distributed.threads

[215](#)

java.security.auth.login.config [219, 221](#)
 JDBC Backing Store Database Configuration Tasks [241](#)
 JDBC Backing Store Database Setup Overview [234](#)
 JMS reconnect attempts [75](#)
 JMS reconnect codes [76](#)
 JMS reconnect timeout [75](#)

K

Key word mapping file [246](#)

L

L1Cache [65](#)
 Limitations in Use of Certain Processing Unit Configurations [95](#)
 localhost [53, 110](#)
 localport [53, 110](#)
 auto incrementing feature [53, 54, 55](#)

M

Machine Overview [167](#)
 Maintaining a Backing Store — If Ontology Object Definitions Change [261](#)
 Max Connections, JDBC resource [40, 252](#)
 Max Size (backing store connection pool) [40](#)
 Metadata Properties for Entities (Events and Concepts) [30](#)
 microagent, enabling [289](#)
 Min Size (backing store connection pool) [40](#)
 Minimum User Permissions [237](#)
 Mixing these approaches [225](#)
 MM Property Group Reference [109](#)
 Modifications Allowed in Hot Deployment [197](#)
 Modify the Project as Needed and Build the EAR File [202](#)

Monitoring and Management Configuration
 Overview 86

N

Names that Exceed the DBMS Maximum Column
 Length 235
 New property 249
 New table 249

O

Ontology (Cache Objects) Overview 175
 Ontology Identifiers That Use Database Key
 Words 236
 Optional User-Defined Property for Flexibility in the
 Filename 281
 Oracle 238
 Oracle-Only Backing Store Database Configuration
 Tasks 264
 Oracle-Only Backing Store Database Setup
 Overview 260
 Oracle-Only Backing Store Database Setup Tasks 260
 Oracle-Only Backing Store Requirements 260
 Oracle-Only Backing Store Runtime Behavior 271
 Overriding and Extending the Operational Deploy-
 ment Descriptor 278
 Overriding Element Values in Engine Property
 Files 278
 Overview of BusinessEvents Monitoring and
 Management 156
 Overview of Cluster Deployment Configuration 10

P

Pane Types Reference for Alert Configuration 128
 Panels and Panes 163
 Perform Hot Deployment 200, 202
 Performing Hot Deployment in a TIBCO Administra-

tor Domain 200
 Performing Hot Deployment Outside a TIBCO
 Administrator Domain 202
 Permissions—ALLOW and DENY 226
 Pluggable JAAS Login Module 218
 port conflicts, avoiding 53, 110
 Post RTC Options — Cache-aside and
 Write-behind 212
 Primary Tables 255
 Process Overview 168
 Processing Unit Settings 103
 Processing Units Tab 13
 Processing Units Tab Settings and Properties 70
 Project Configuration for Oracle-Only Backing
 Store 270
 Project, Master, and Deployed CDD and EAR Files 95
 Property type change 249
 Provided Configuration Resources 238

Q

Query Agent Overview 174

R

Reading From the Backing Store At Runtime 271
 Reading From the Backing Store at Startup 271
 reconnect to JMS, attempts 75
 reconnect to JMS, which messages 76
 reconnectChannels() 315
 Reference to the Metrics 156
 Required Location of Access Control Files 227
 resetTotalNumberRulesFired() 316
 Resource Types and Corresponding Action Types 231
 Resources Required for Setting Up the Database 237
 Resources Required for Setting Up the Oracle
 Database 262
 resumeChannels() 317
 resumeDestinations() 318
 Reverse Reference Tables 256
 RTC Options — Single-Threaded or Concurrent 210

S

- scripted deployment 188
- Secondary Tables 256
- Shared Queue and Threads 206
- Shutdown Sequence 276
- Single-Threaded RTC 210
- Site Settings 100
- Site Topology Overview 93
- Site Topology Reference 100
- Special Cases 234, 235
- Specifying and Grouping Decision Project Resources 227
- Specifying Limited Cache Size 254
- Specifying Operational Override File Locations 280
- Specifying Permissions
 - Action Types 231
- Specifying Resources 227
- Specifying the Alert Message 126
- Specifying the Path to an Alert Metric Value (and a Reference Value) 125
- SQL Server 237
- Starting a BusinessEvents Engine at the Command Line 185
- Starting MM Management Server and Using MM Console 146
- startup actions, deleted entities cleanup (legacy) 43, 272
- Startup Sequence 275
- startup sequence 275
- stopApplicationInstance() 319
- Strategy (backing store) 39
- String Properties That Exceed the Maximum Database Column Length 236
- Structure of the Access Control File 225
- support, contacting xxvii
- suspendChannels() 320
- suspendDestinations() 321
- System Alerts Pane 166

T

- tangosol.coherence.cacheconfig 55, 254

TIBCO BusinessEvents Administration

- tangosol.coherence.clusteraddress 51, 52
- tangosol.coherence.clusterport 51, 52
- tangosol.coherence.distributed.backupcount 16, 50
- tangosol.coherence.localhost 53
- tangosol.coherence.localport 53
- tangosol.coherence.localport.adjust 53, 54, 55
- tangosol.coherence.log 72
- tangosol.coherence.log.level 73
- tangosol.coherence.log.limit 73
- tangosol.coherence.ttl 51, 52
- tangosol.coherence.wkan 19, 54
- tangosol.coherence.wkan.port 19, 54
- Task Summary 12, 87
- technical support xxvii
- The Procedure 250
- Thread Analyzer 161
- Threading Models Overview 204
- TIBCO administration domains 188
- TIBCO Administrator
 - and server management 188
 - deployment procedure 191
- TIBCO Hawk
 - enabling microagent 289
 - list of methods 287
 - methods overview 289
- TIBCO Hawk Methods Overview 289
- TIBCO_HOME xxiv
- tibco.clientVar.Domain 114, 114, 114, 114
- tibco.repouurl property for EAR name and path 186
- timeouts
 - JMS reconnect 75
- Tuning Properties for Cache-aside Strategy 213
- Two Types of Thresholds 131
- Type (of backing store) 39
- Types of Methods 289

U

- Understanding Alerts 121
- Understanding and Configuring Alerts 121
- Understanding Entity Caches 283
- Understanding Health Metric Rules 130
- Updating an Existing Backing Store Database

Schema [248](#)

Updating an Existing Oracle Database Schema [268](#)

URI (for backing store) [40](#)

User Authentication Overview [218](#)

Using Agent Class Properties at Different Levels [10](#)

Using Collections [24](#)

W

What the Schema Update Utility Can and Can't Handle Automatically [248](#)

Working with Access Control Files [227](#)

Working with Cluster Explorer [161](#)

Working with MM Metric Panes [163](#)

Write-behind [214](#)

Writing to the Backing Store [271](#)