Important Information

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

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

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

TIBCO, The Power of Now, TIBCO ActiveMatrix, TIBCO ActiveMatrix BusinessWorks, TIBCO Administrator, TIBCO ActiveSpaces, TIBCO Designer, TIBCO Enterprise Message Service, TIBCO Hawk, TIBCO Runtime Agent, TIBCO Rendezvous, are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

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

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README.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 © 2004-2013 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information
Contents

Tables .......................................................................................................................... v

Preface ....................................................................................................................... vi

Related Documentation ............................................................................................ viii
  TIBCO BusinessEvents and Add-On Product Documentation .................................. viii
  Accessing TIBCO BusinessEvents Functions Reference Documentation ................ xi
  Other TIBCO Product Documentation ................................................................... xi

Typographical Conventions ...................................................................................... xiii

Connecting with TIBCO Resources ........................................................................... xvi
  How to Join TIBCOmmunity ................................................................................ xvi
  How to Access TIBCO Documentation ................................................................. xvi
  How to Contact TIBCO Support ........................................................................... xvi

Chapter 1  TIBCO BusinessEvents Process Orchestration — Overview .................... 1

  Product Overview .................................................................................................. 2
    Support for BPMN 2.0 ......................................................................................... 2
    Project Requirements .......................................................................................... 3
    Types of Users ..................................................................................................... 3
    When to Use Process Orchestration .................................................................. 3

  Overview of the Process Diagram User Interface .................................................. 4

  Overview of Runtime .............................................................................................. 5
    Process Agent ....................................................................................................... 5
    Runtime Functionality ......................................................................................... 5

  Task Summary ....................................................................................................... 7

Chapter 2  Understanding Business Process Diagrams .............................................. 9

  Understanding Processes ....................................................................................... 10
    Types of Processes ............................................................................................. 10
    Process Variables .............................................................................................. 11
    Sub-processes ..................................................................................................... 11
    Elements in a Process ......................................................................................... 12

  Connectors (Edges) ............................................................................................... 13
    Sequence Flow .................................................................................................... 13

  Events .................................................................................................................... 14
    Start Events and End Events .............................................................................. 14
<table>
<thead>
<tr>
<th>Contents</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Activities</td>
<td>16</td>
</tr>
<tr>
<td>Activity Loop Characteristics</td>
<td>16</td>
</tr>
<tr>
<td>Gateways</td>
<td>18</td>
</tr>
<tr>
<td>Exclusive</td>
<td>18</td>
</tr>
<tr>
<td>Parallel</td>
<td>19</td>
</tr>
<tr>
<td>Chapter 3 Working with Processes</td>
<td>21</td>
</tr>
<tr>
<td>Creating a Project to Support Processes</td>
<td>22</td>
</tr>
<tr>
<td>Project Configuration Prerequisites</td>
<td>22</td>
</tr>
<tr>
<td>Project Resources Used in Process Elements</td>
<td>22</td>
</tr>
<tr>
<td>Adding a Process</td>
<td>23</td>
</tr>
<tr>
<td>Startup and Shutdown Processes</td>
<td>25</td>
</tr>
<tr>
<td>Chapter 4 Process Agent Configuration And Runtime</td>
<td>27</td>
</tr>
<tr>
<td>Process Execution (Runtime)</td>
<td>28</td>
</tr>
<tr>
<td>Task Execution</td>
<td>28</td>
</tr>
<tr>
<td>Checkpointing and Recovery</td>
<td>28</td>
</tr>
<tr>
<td>Locking and Unlocking</td>
<td>29</td>
</tr>
<tr>
<td>Adding and Configuring Process Agent Class</td>
<td>30</td>
</tr>
<tr>
<td>Running Process Agent</td>
<td>31</td>
</tr>
<tr>
<td>Build EAR file</td>
<td>31</td>
</tr>
<tr>
<td>Run Process Agent</td>
<td>31</td>
</tr>
<tr>
<td>Index</td>
<td>33</td>
</tr>
</tbody>
</table>

TIBCO BusinessEvents Process Orchestration Developer's Guide
Tables

Table 1 General Typographical Conventions .................................................. xiii
Table 2 Syntax Typographical Conventions .................................................... xiv
Table 3 Ontology Items that Support Process Elements. ................................. 22
Table 4 CDD Agent Classes Tab Process Agent Settings .................................. 30
vi | Tables
Preface

TIBCO BusinessEvents® Process Orchestration software is an add-on product to TIBCO BusinessEvents software. It provides CEP functionality within the context of a BPM process, enabling you to segregate different CEP rule sets within the flow of a BPM process.

Topics

- Related Documentation, page viii
- Typographical Conventions, page xiii
- Connecting with TIBCO Resources, page xvi
Related Documentation

This section lists documentation resources you may find useful.

TIBCO BusinessEvents and Add-On Product Documentation

The following diagram shows the main documents in the TIBCO BusinessEvents documentation set, and the documentation sets for the optional add-on products.

Each set also contains an installation guide, release notes, and a readme file.

TIBCO BusinessEvents Documentation

TIBCO BusinessEvents Studio, the design-time UI, is supported on Windows and Linux. The documentation set for TIBCO BusinessEvents is as follows.
• **TIBCO BusinessEvents Installation**: Read this manual for instructions on site preparation, installation, upgrading from an earlier release, and project migration.

• **TIBCO BusinessEvents Getting Started**: After the product is installed, use this manual to learn the basics of TIBCO BusinessEvents: project design, cache OM, and backing store. This guide 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**: Use this guide when you implement a project design in TIBCO BusinessEvents Studio. It covers topics such as project-level tasks, resource-level tasks, debugging, and integration with TIBCO ActiveMatrix BusinessWorks. It also explains how to configure the CDD file for different object management options, and set up a backing store.

• **TIBCO BusinessEvents Administration**: This book explains how to configure, deploy, monitor, and manage a TIBCO BusinessEvents application and the data it generates using TIBCO BusinessEvents Monitoring and Management component, TIBCO Administrator, or at the command line. It includes authentication and authorization topics.

• **Online References**:
  — **TIBCO BusinessEvents Java API Reference**: This online reference is available from the HTML documentation interface. It provides the Javadoc-based documentation for the TIBCO BusinessEvents API.
  — **TIBCO BusinessEvents Functions Reference**: This reference is available from the HTML documentation interface. It provides a listing of all functions provided with TIBCO BusinessEvents, showing the same details as the tooltips available in TIBCO BusinessEvents Studio.

• **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.

---

**TIBCO BusinessEvents Event Stream Processing**

This TIBCO BusinessEvents add-on is available separately, and includes the TIBCO BusinessEvents Query Language features and the Pattern Matcher Service.

• **TIBCO BusinessEvents Event Stream Processing Installation**: Read this brief manual for installation instructions. A compatible version of TIBCO BusinessEvents must be installed before you install any add-on.
Related Documentation

- **TIBCO BusinessEvents Event Stream Processing Query Developer’s Guide**: This manual explains how to use the object query language to query various aspects of the running system. For details on configuring and deploying query agents, see **TIBCO BusinessEvents Developer’s Guide**.

- **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**

This TIBCO BusinessEvents add-on is available separately. It incorporates the Decision Manager decision modeling business user interface (supported on Windows and Linux), and the Rules Management Server (supported on all platforms supported by TIBCO BusinessEvents).

- **TIBCO BusinessEvents Decision Manager Installation**: Read this brief manual for installation instructions. A compatible version of TIBCO BusinessEvents must be installed before you install any add-on.

- **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 TIBCO 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 before you install any add-on.

- **TIBCO BusinessEvents Data Modeling Developer’s Guide**: This manual explains data modeling add-on features for TIBCO BusinessEvents. The database concepts feature enables you to model TIBCO 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 Process Orchestration**

This TIBCO BusinessEvents add-on is available separately. It provides CEP functionality within the context of a BPM process, enabling you segregate different CEP rule sets within the flow of a BPM process.

• **TIBCO BusinessEvents Process Orchestration Installation**: Read this manual for instructions on site preparation and installation. A compatible version of TIBCO BusinessEvents must be installed before you install any add-on.

• **TIBCO BusinessEvents Process Orchestration Developer’s Guide**: This guide explains how configure and deploy business processes whose actions are carried out using TIBCO BusinessEvents project resources.

• **TIBCO BusinessEvents Process Orchestration 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 TIBCO 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. A compatible version of TIBCO BusinessEvents must be installed before you install any add-on.

• **TIBCO BusinessEvents Views Getting Started**: After the product is installed, use this manual to learn how to use TIBCO BusinessEvents Views to create and run a dashboard using a step-by-step tutorial.

• **TIBCO BusinessEvents Views Developer’s Guide**: This guide explains how to use 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 TIBCO 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.
Accessing TIBCO BusinessEvents Functions Reference Documentation

Reference documentation for functions, including those used in add-ons, is available in the HTML documentation interface for the TIBCO BusinessEvents documentation set, and as tooltips in TIBCO BusinessEvents Studio. To use the HTML-based functions reference from the file system do the following:


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.

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

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENV_NAME</td>
<td>TIBCO products are installed into an installation environment. A product installed into an installation environment does not access components in other installation environments. Incompatible products and multiple instances of the same product must be installed into different installation environments.</td>
</tr>
<tr>
<td>TIBCO_HOME</td>
<td>An installation environment consists of the following properties:</td>
</tr>
<tr>
<td>BE_HOME</td>
<td>- <strong>Name</strong> Identifies the installation environment. This name is referenced in documentation as <code>ENV_NAME</code>. On Microsoft Windows, the name is appended to the name of Windows services created by the installer and is a component of the path to the product shortcut in the Windows Start &gt; All Programs menu.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Path</strong> The folder into which the product is installed. This folder is referenced in documentation as <code>TIBCO_HOME</code>.</td>
</tr>
<tr>
<td></td>
<td>TIBCO BusinessEvents installs into a directory within a <code>TIBCO_HOME</code>. This directory is referenced in documentation as <code>BE_HOME</code>. The default value of <code>BE_HOME</code> depends on the operating system. For example on Windows systems, the default value is <code>C:\tibco\be\5.2</code>.</td>
</tr>
</tbody>
</table>

**Code font** identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:

Use **MyCommand** to start the foo process.

**Bold code font** is used in the following ways:

- In procedures, to indicate what a user types. For example: Type `admin`.
- In large code samples, to indicate the parts of the sample that are of particular interest.
- In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, `MyCommand` is enabled: `MyCommand [enable | disable]`
Table 1  General Typographical Conventions (Cont’d)

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>italic font</td>
<td>Italic font is used in the following ways:</td>
</tr>
<tr>
<td></td>
<td>• To indicate a document title. For example: See <em>TIBCO ActiveMatrixBusinessWorks Concepts</em>.</td>
</tr>
<tr>
<td></td>
<td>• To introduce new terms For example: A portal page may contain several <em>portlets</em>. Portlets are mini-applications that run in a portal.</td>
</tr>
<tr>
<td></td>
<td>• To indicate a variable in a command or code syntax that you must replace. For example: <code>MyCommand PathName</code></td>
</tr>
<tr>
<td>Key</td>
<td>Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C.</td>
</tr>
<tr>
<td>combinations</td>
<td>Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.</td>
</tr>
<tr>
<td>📝</td>
<td>The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.</td>
</tr>
<tr>
<td>💡</td>
<td>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.</td>
</tr>
<tr>
<td>⚠️</td>
<td>The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.</td>
</tr>
</tbody>
</table>

Table 2  Syntax Typographical Conventions

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ]</td>
<td>An optional item in a command or code syntax. For example: <code>MyCommand [optional_parameter] required_parameter</code></td>
</tr>
<tr>
<td></td>
<td>A logical OR that separates multiple items of which only one may be chosen. For example, you can select only one of the following parameters: `MyCommand param1</td>
</tr>
</tbody>
</table>
A logical group of items in a command. Other syntax notations may appear within each logical group.

For example, the following command requires two parameters, which can be either the pair `param1` and `param2`, or the pair `param3` and `param4`:

```
MyCommand {param1 param2} | {param3 param4}
```

In the next example, the command requires two parameters. The first parameter can be either `param1` or `param2` and the second can be either `param3` or `param4`:

```
MyCommand {param1 | param2} {param3 | param4}
```

In the next example, the command can accept either two or three parameters. The first parameter must be `param1`. You can optionally include `param2` as the second parameter. And the last parameter is either `param3` or `param4`:

```
MyCommand param1 [param2] {param3 | param4}
```
Connecting with TIBCO Resources

This section provides links to helpful TIBCO resources.

How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts, a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to http://www.tibcommunity.com.

How to Access TIBCO Documentation

You can access TIBCO documentation here:

http://docs.tibco.com

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, 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  TIBCO BusinessEvents Process Orchestration — Overview

This chapter provides a short overview of process orchestration features.

Topics

- Product Overview, page 2
- Overview of the Process Diagram User Interface, page 4
- Overview of Runtime, page 5
- Task Summary, page 7
TIBCO BusinessEvents Process Orchestration is an add-on product to TIBCO BusinessEvents software. It provides CEP functionality within the context of a BPM process, enabling you to simply and efficiently partition the rule space, that is, to segregate different CEP rule sets within the flow of a BPM process.

You could also say that TIBCO BusinessEvents Process Orchestration provides BPM functionality within the context of a CEP application, giving the business process access to TIBCO BusinessEvents project resources and features.

Thus TIBCO BusinessEvents Process Orchestration combines features of complex event processing (CEP) and business process management (BPM) technologies:

- Complex event processing (CEP) applications allow “events” to be correlated and processed on a continuous basis. The outcome is not predetermined. It depends on what events arrive, what rules are triggered, and what changes to the ontology may result in triggering more rules, and so on. All rules active in an agent can be triggered if conditions are right.

- Business process management (BPM) applications, by contrast, orchestrate various activities into a predefined process, so as to achieve a predetermined business goal, such as provisioning a cell phone. Only certain rules should execute at any given step of the process.

With TIBCO BusinessEvents Process Orchestration, you can execute different tasks, each of which may use a few or hundreds of rules to complete.

Support for BPMN 2.0

TIBCO BusinessEvents Process Orchestration is compliant with a subset of functionality specified in the Business Process Model and Notation (BPMN) specification version 2.0. It supports functionality appropriate for its role in a CEP application.

TIBCO BusinessEvents Process Orchestration provides event-driven process execution with BPMN-based modeling semantics. It is not intended to be a complete process modeling tool. For example it does not offer human workflow, workforce allocation, document management, asset management, or organizational model capabilities.

If you also use the TIBCO BusinessEvents Data Modeling add-on product, you can extend the functionality provided using state modeling features (provided in TIBCO BusinessEvents Data Modeling add-on), as desired.
Project Requirements

TIBCO BusinessEvents Process Orchestration projects require use of the following TIBCO BusinessEvents features:

- Cache OM with backing store, and process agents (a special type of agent).
- Load Balancer (Optional)


Types of Users

The add-on supports development of business process models by three kinds of users:

- **Business Users** Define business processes at a high level, with annotations. Business users own the processes.
- **Analysts** Add more detail so that the architect or developer can implement the model. Analysts interact with business users to improve and refine the model and update it in the TIBCO BusinessEvents Process Orchestration add-on.
- **Technical Users** Implement the design. They add necessary functions, simple and time events, Java code, and other items needed to fully implement the process so it can be executed by the TIBCO BusinessEvents engine.

Prerequisite Skills

Technical users should be familiar with the Business Process Model and Notation (BPMN) specification, which is available here:

http://www.omg.org/spec/BPMN/2.0/PDF

All users should have some familiarity with business process modeling.

When to Use Process Orchestration

**Straight Thru Processing** - TIBCO BusinessEvents Process Orchestration provides the state processing model to automate the life cycle of a concept.

State models are available in the TIBCO BusinessEvents Data Modeling add-on.

**Classical Business Process Model** - A classical business process model depends on the actions and illustrates the activities that occur in a predefined order (depending on conditions) to achieve the overall business goal of the process. This includes the workflow, Case Management (Intelligence).
Overview of the Process Diagram User Interface

This section explains the user interface sections.

**Project Explorer** (Top left) Standard project explorer as normally seen in TIBCO BusinessEvents. Processes are added as resources in the usual way.

**Overview Window** (Lower left) Standard overview window as seen in other diagrams within TIBCO BusinessEvents.

**Editor** (Upper middle) Uses a similar canvas-based user interface to other diagramming tools in TIBCO BusinessEvents.

**Property Sheets** (Lower middle) Context sensitive property sheets enable you to view and edit items in the diagram that have properties. Select an item to display its properties. Multiple left tabs switch between different sheets.

**Palette** (Right) The palette has various collapsible sections for the different kinds of items you can drop onto the canvas when drawing a process. The palette is customizable and extensible.
Overview of Runtime

This section describes the functionality supported at runtime for process execution.

Process Agent

At runtime, a process agent allows users to select and deploy all processes marked public. All private processes are deployed internally.

A process agent is just an inference agent. The Process Orchestration engine works in cache-only mode and, therefore, you must choose the Cache Only OM option when configuring an inference agent. You can deploy any number of process agents, query agents and cache agents.

A cluster can have either an inference agent or a process agent, but not both, so do not mix a stand alone inference agent and process agent in a cluster.

A pre-processor attached to a Event Channel destination allows the agent to filter, modify, do locking and loading of transactional data before it is passed to the inference agent for processing using a preprocessing context. In case of a process agent, the pre-processor activities can be performed using a Rulefunction Task for an incoming event, and any contextual association is done using the process job variables.

Runtime Functionality

Following functionality is supported for process execution during runtime:

- Process Template: Process template is a model of a process (more details).
- Process Agent: A process agent runs all the deployed processes.
- Tasks Execution: Various tasks such as Script, Business Rule, Send Message, Receive Message, Web Service, Inference, Sub_Process, Call Activity are supported during process execution at runtime.
- Job - Each instance of a process template is a job.
- Job Context means data plus additional state that each job maintains for status, recovery, reporting, metrics etc.
- Checkpoint: Checkpointing allows persistence of the process job data to the cache and datastore.
• ReplyEvent: The ReplyEvent allows the acknowledgement and reply to Start events coming via EMS, HTTP transport.
• Gateway Execution: Both Exclusive and Parallel gateways are supported during process execution.
At a high level, here are the tasks required to configure a TIBCO BusinessEvents project that contains one or more processes.

- Create a TIBCO BusinessEvents Studio project and add a JMS channel, events, and other project artifacts as needed.

- Create process diagrams with tasks and a default flow. This step can be performed at a high level by a business user, using the Documentation tab to “storyboard” the process, that is, describe the tasks and transitions in the process without completing the implementation.

- Link the tasks to project resources such as events, rule functions, and rules. In this step the technical team member assesses the storyboard version of the project (if any) and implements it using process and project resources.

- Validate the processes.

- In the project CDD, configure the process agent, load balancer, and other aspects of the project

- Validate and deploy.
Chapter 2  Understanding Business Process Diagrams

This chapter provides a short introduction to the components of a business process diagram, and how in TIBCO BusinessEvents Process Orchestration, each component leverages features of TIBCO BusinessEvents.

Many resources on BPMN are available and this manual does not attempt to present a BPMN primer. However, it does outline some basics so that you can understand how TIBCO BusinessEvents project resources are used to implement the functionality of different parts of a BPM process.

Topics

- Understanding Processes, page 10
- Events, page 14
- Activities, page 16
- Gateways, page 18
Understanding Processes

A business process is a flow of business activities related to one business goal, such as providing a complex service to a customer. A process diagram is a graph that describes the flow of activities in a business process. The graph is built up of connected elements of different types such as activities and events.

A process can describe different paths of which one or more paths are taken at runtime, depending on conditions.

A notional token flows through the process, indicating which particular path or paths are taken as the process executes.

Types of Processes

Processes are of two main types, public and private.
Public Processes

A public process can be added to a process agent in the CDD and deployed. Public processes are often instantiated by arrival of a message at a TIBCO BusinessEvents destination, which causes an instance of a process & run the process.

Private Processes

Private processes are only used within a public process. They can be called using a call activity or they can be triggered by an error, or other situation.

Private processes can be shared by multiple public processes.

Private processes are automatically deployed when a public process that references them is deployed. All private processes are deployed.

Process Variables

Process variables are similar to concept properties in TIBCO BusinessEvents. Process variables are used to hold data that enters the process during its execution at runtime. An XPath mapper enables you to map data from simple events to process variables and from process variables to resources such as other simple events, rules, and so on.

You can define process variables of these types: String, int, long, double, Boolean, DateTime, and Concept. Variables can be flagged as Multiple, that is, arrays.

When you select Concept as the process variable type, you then select a concept from the project ontology. This concept behaves like a contained concept, with respect to the process, which behaves like the containing concept.

Sub-processes

Sub processes are processes that exist within another process.

A sub-process has access to all variables in all its parent processes (including parent sub-processes).

Sub-processes are of two types, standard and event-based.

Standard Sub-Process

A standard sub-process can only be reached by a sequence flow in the parent process.
A standard sub-process can begin with a unique empty Start Event, or an activity or gateway that has no incoming sequence flow. It cannot start with a message, signal, or timer start event.

### Elements in a Process

A BPMN process is a set of different types of flow elements. The following table introduces these BPMN standard elements supported in this release, and their icons.

The icon shapes are as specified in the BPMN standard. Subtypes of each element type have the same basic shape with distinguishing characteristics, also as specified in the standard. Swimlanes are not supported in this release.

<table>
<thead>
<tr>
<th>Element Types</th>
<th>Elements and Icons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow elements</td>
<td><img src="image" alt="Start/End Events" /></td>
</tr>
<tr>
<td><strong>See</strong> Events, page 14</td>
<td><strong>See</strong> Activities, page 16</td>
</tr>
<tr>
<td>Connectors</td>
<td><img src="image" alt="Sequence flow" /></td>
</tr>
<tr>
<td>Artifacts</td>
<td>Note</td>
</tr>
</tbody>
</table>
Connectors (which might also be termed connectors, edges, or transitions) are of two types:

- **An association** associates data or artifacts with flow objects. They can show how data is provided to or generated by an activity. They can link text annotations with the objects they annotate. There is not much more to say about associations.

- **A sequence flow** is a connector that indicates the order in which activities are done. Sequence flow connectors connect events, activities, and gateways. The next section explains more about sequence flow.

### Sequence Flow

A sequence flow is an arrow that links flow objects. Sequence flows (along with gateways) show the order in which activities will be performed in a process.

A sequence flow can be one of the following kinds:

- **Default** When multiple paths flow from a gateway, check the Default Flow checkbox to specify one of them as the default. The default path is taken only when conditions are such that no other path is valid.

- **Conditional** If you add a Boolean expression to the sequence flow’s Expression tab, the flow proceeds to the next task only if the expression evaluates to true.

- **Uncontrolled** A flow that is unaffected by conditions, and does not pass through a gateway. Can be used for simply connecting two activities, or can diverge from or converge to an activity, in parallel. (Similar to the lambda transition in a state model)

Note that the BPMN specification discusses additional types of flow. In TIBCO BusinessEvents Process Orchestration, various types of flow are selected internally based on context, and there is no need to know about them.

### Sequence Flows Emerging from Gateways

In this release, only sequence flows emerging from gateways can have expressions.
Events

*Event* is a technical term both in BPMN and in TIBCO BusinessEvents Process Orchestration. A BPMN event is a signal that something happened. A BPMN event can be one of three types: start, intermediate, or end. Each type has subtypes used for different purposes.

**Message and Event**

In BPM, the terms *message* is used to denote anything that arrives into or leaves a process. In TIBCO BusinessEvents, the term message denotes information arriving at a destination, where it is transformed into a TIBCO BusinessEvents simple event. In TIBCO BusinessEvents, the term *event* is often used loosely to mean *message*, as in, “the event arrives at the JMS destination.”

**BPMN Events and TIBCO BusinessEvents Events**

BPMN event types are implemented using various TIBCO BusinessEvents resources, as shown in Table 5, Palette Reference Summary, on page 34. The Message Start BPMN event maps to a TIBCO BusinessEvents simple event, whose arrival causes an instance of a process (that is, a job) to be created.

**BPMN Event Triggers and Results**

In BPMN, events usually have a cause (*trigger*) or an effect (*result*):

**Events that catch a trigger** Start events and some intermediate events are catching events. The trigger is the cause. The trigger can be a simple event or it could be a system alert about shutdown, for example.

**Events that throw a result** End Events and some intermediate events are throwing events. The result they throw could be a TIBCO BusinessEvents simple event or timer event, or it could be an event sub-process.

**Start Events and End Events**

Start events indicate the beginning of a process. End events indicate where a path of a process ends.

**Implicit and Explicit Start and End Events**

Use of start events and end events is optional. Other types of activities can be used to start a process, as long as they have no input mapping. Gateways can also be used to start a process. Similarly, other types of activities can be used to end a process, as long as they have no output mapping. These are known as *implicit start events* and *implicit end events*. 

If there is an explicit start event, there must also be an explicit end event, and the reverse is also true. Also, if there are start and end events, then implicit start and end events cannot be used. A process does not end until all start events have been visited, or, in the case of implicit start events, all parallel paths have completed.

Since the events are consumed optionally at the process end events, the concept of pre-processing doesn’t apply to the Process Agents although users can add rule functions into the input destination for pre-processing.

Effect of Abnormal Termination

If a sub-process ends abnormally, the containing process does not terminate. If a multi-instance activity terminates abnormally, only the affected instance is terminated. (See Activity Loop Characteristics, page 16 for an explanation of multi-instance activity.)

Message Start and Signal Start Events

Message start and signal start events are configured using TIBCO BusinessEvents simple events. Validation checks at design-time ensure that you use a correctly configured simple event:

- Message start events require a simple event with a queue destination
- Signal start events require a simple event with a topic destination.

Example

As a simple example, suppose a process begins with a message start event and ends with a message end event.

**How the process starts** The process instance is instantiated when the simple event specified in a message start event arrives through a destination. The simple event properties appear as scope variables in the Output map, and you map them to process (job) variables, so they are available to the other activities later in the process.

**How the process ends** At the end of the process the end event transfers information available from the Input map (that is, values stored in the process variables) to the specified simple event’s properties and sends it to its default destination or specified destination.

Timer Start Events

The start of the process is triggered by a time-date or repeating time interval, for example, every Monday at 9am. In BPMN, the timer start event uses the TIBCO BusinessEvents time event trigger, therefore the event can be a repeating time event or a rule based scheduled time event. See TIBCO BusinessEvents timer event definition.
## Activities

A BPMN activity is a unit of work performed within a process. Activities are flow objects and they are shown using round-cornered rectangles. They are also known as tasks.

The following are the BPMN activity types:

- Tasks
- Call activities
- Sub-processes

Various task sub-types are available. Most of them provide TIBCO BusinessEvents-specific functionality, such as Send Event, Decision Table, Rule Function, and so on.

### Activity Loop Characteristics

Activities can have looping characteristics. Looping behavior is available for the following types of tasks:

- Script (TIBCO BusinessEvents rule function)
- Send Message
- Receive Message
- Web Service

### Types of Looping

When adding an activity, you can optionally use the Loop Characteristics tab to configure the activity to be a loop activity (Standard option) or multi-instance activity (Multi-Instance option). Activities configured in these ways are also known as loop activities and multi-instance activities.

#### Standard Looping

Standard looping behavior is based on a Boolean condition: the activity repeats (loops) as long as the Boolean condition is true. You can configure the evaluation to occur either at the start or end of each loop iteration and you can specify a maximum number of iterations.
Multi-Instance Looping

While standard looping continues until a condition, the multi-instance option lets you specify a certain number of activity instances to execute (loop cardinality). The number of instances is evaluated once.

The number of instances can be determined using an XPath expression.

In the BPMN specification, instances can execute either in parallel or sequentially. However in the current release of TIBCO BusinessEvents Process Orchestration, only sequential execution is supported.

The Multi-Instance Behavior setting determines whether a BPMN signal event is thrown from an activity instance that is about to complete. If one or more events is thrown (as in the NONE and ONE options), it is automatically passed all the attributes of the multi-instance task.
Gateways control how the process flows. They define interactions between sequence flows in a process.

Following types of gateways be supported:

- Exclusive
- Parallel

**Exclusive**

Exclusive gateways (also known as XOR gateways) specify multiple possible paths, only one of which is actually taken. All the outgoing flow paths are evaluated and the flow whose condition is evaluated to true is taken for process execution.

Exclusive gateways are used in two ways:

- Join: Join specifies the branching of the flow into multiple possible paths, or to merge these possible paths.
- Fork: Fork specifies the possible multiple paths. The selection logic is specified in the sequence flows. One sequence flow is defined as the default.

**Joining**

A converging exclusive gateway is used to merge alternative paths. Each incoming control flow token is routed to the outgoing Sequence Flow without synchronization. Each Token arriving at any incoming sequence flow activates the gateway and is routed to exactly one of the outgoing sequence flows.

The Exclusive Gateway has pass-through semantics for a set of incoming branches (merging behavior). This will behave like the parallel gateway merging except that each incoming sequence will require a individual rule and the rule will be satisfied with the message token coming via the sequence flow.

**Sequence Flows Emerging from Gateways**  
In this release, only sequence flows emerging from gateways can have expressions. Only one path is taken, the first one whose expression evaluates to true. If no path evaluates to true then the default path is taken.
When an exclusive gateway is used where two or more sequence flows merge, the gateway does not impose synchronization or evaluate any sequential flow expression, it simply allows the execution proceed to the next task after the gateway.

**Forking**

Gateway activation leads to the activation of exactly one out of the set of outgoing branches. Which branch is determined by if-then-else conditional logic using information available from the sequence flows. In order to determine the outgoing sequence flow, the conditions are evaluated in order. The first condition that evaluates to true determines the sequence flow the token is sent to. Evaluation ceases at that point. If none of the conditions evaluates to true, the default sequence flow is taken, and if a default flow has not been specified, an exception is thrown.

**Parallel**

Parallel gateways define multiple paths, or convergence and synchronization of multiple paths. Parallel gateways can be of two types: Join and Fork.

**Joining**

The joining type of parallel gateway synchronizes concurrent branches or threads of execution. The gateway is activated if it receives an execution token from every incoming sequence flow and the token is consumed.

To model this behavior a rule is required whose condition would require incoming messages from each incoming sequence flow to be asserted to the Process Rete and each message context providing identification of the sequence path travelled and the last flow node it originated from.

If there are excess Tokens at an incoming sequence flow, these Tokens remain at this sequence flow after execution of the gateway. This could mean that the out of context msg has arrived and it needs to be correlated there it should also get asserted.

**Forking**

Parallel gateway forking is used to spawn concurrent threads or branches. This will involve creating a Job hierarchy i.e. Parent job spawns child jobs and each job keeps progressing on the assigned sequence flow.
If the sequence flow activities have no outgoing Sequence Flow and there are no End Events in the containing Process or Sub-Process, the Activities terminate and termination semantics for the container are applied.

The parent job can have multiple child jobs but a child job can have only one parent job. Each job can be identified by <parent job id>.<child job id> and it can have its own job Context. All the concurrent job are scheduled to be executed immediately once the gateway synchronization happens in the rule action.
Chapter 3  Working with Processes

This chapter explains how to add and work with processes in TIBCO BusinessEvents Studio.

Topics

- Adding a Process, page 23
- Startup and Shutdown Processes, page 25
Creating a Project to Support Processes

Project Configuration Prerequisites

At a minimum, the project must provide the following configuration options:

- Cache object management
- Backing store
- Content-aware load balancer
- JMS channel and destination

See *TIBCO BusinessEvents Developer’s Guide* for details on all of the above. If you are new to TIBCO BusinessEvents, first gain familiarity with core features using the *TIBCO BusinessEvents Getting Started* guide.

Project Resources Used in Process Elements

Your TIBCO BusinessEvents process must provide the resources needed to support the functionality defined in the tasks of a process:

<table>
<thead>
<tr>
<th>Process Element</th>
<th>Project Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Script</td>
<td>Rule Function</td>
</tr>
<tr>
<td>Business Rule</td>
<td>Decision Table (Requires TIBCO BusinessEvents Decision Manager)</td>
</tr>
<tr>
<td>Inference</td>
<td>Rule</td>
</tr>
<tr>
<td>Send Message</td>
<td>Simple event</td>
</tr>
<tr>
<td>Receive Message</td>
<td></td>
</tr>
<tr>
<td>Message Start</td>
<td></td>
</tr>
<tr>
<td>Message End</td>
<td></td>
</tr>
<tr>
<td>Error End</td>
<td></td>
</tr>
</tbody>
</table>

Table 3  Ontology Items that Support Process Elements
Adding a Process

This procedure assumes that the TIBCO BusinessEvents project resources required to support the process are in place. See TIBCO BusinessEvents Developer’s Guide for details about creating project resources.

**To Add a Process**

1. In BusinessEvents Studio Explorer, right click the folder where you want to store the process and select **New > Process**. You see the New Process Wizard.

2. In the Process Name field, type a name for the process. In the Description field, type a description as desired.

3. Click **Finish**. You see the Process Editor showing a simple process outline.

4. Click in any blank part of the process canvas so that the process-level property sheets are visible, then complete them as follows:

   — In the General tab, enter a label and author as desired, and specify if the process type is Public or Private.

   — In the documentation tab, as desired, use the rich text editor to add documentation about the process. See Adding, Exporting and Viewing Process Documentation on page 68.

   — In the Variables tab, add all the process variables needed for data storage needs. For example, data from incoming events is copied to the process variables for use in a later activity, for example by a rule function.

5. Place a Message Start event on the canvas and configure it as follows:

   — In the General tab Resource field, specify the simple event or time event that triggers the process to execute.

   — In the documentation tab, as desired, use the rich text editor to add documentation about the event. See Adding, Exporting and Viewing Process Documentation on page 68.

   — In the Output map, map the properties of the event you selected in the General tab to process variables. Map event properties on the left to process (job) variables on the right.

Click **OK**.

You cannot change the name in the editor. To change the name of any project element, right-click the element in BusinessEvents Studio Explorer and select **Refactor > Rename**.
6. Do any of the following according to the needs of the process (Use Chapter 5, Palette Reference, on page 33 for flow element usage guidelines):

   — Place an activity on the canvas, such as a Script task. Configure the activity according to the reference sections in Chapter 5, Palette Reference. For example, specify loop characteristics, use the Input map to map data from process variables to the activity, use the Output map to map data from the activity (that is, values after processing has occurred) to process variables, and so on, according to the available options for each type of activity. You can use the Documentation tab to add documentation about the activity.

   — Connect the items with sequence flows and Add gateways to control the flow. Configure expressions for sequence flows emerging from gateways, and select one as the default flow. You specify the type of sequence flow on the General tab: Uncontrolled, Conditional, or Default. If you select Conditional, then define the expression as explained next. (See Gateways on page 18 for details on gateways.)

   — If you are configuring a Conditional sequence flow out of a gateway, then in the Expression tab, use the XPath Formula builder to define an expression that has to evaluate to true in order for the next activity to execute. In the Data tab of the Expression editor, you can use process variables in the formula. In the Functions tab you can use a library of provided functions and use the Constants tab to select items such as Timezone Formats, quotes and other items as needed.

7. Add one or more End events, as needed to end the paths of the process.

   — In the General tab, choose an event type as the output of the process (if you chose message end or signal end)

   — In the Documentation tab, use the rich text editor to document the task, as desired.

   — In the Input tab, map process variables as needed to the event that is sent out at the end of the process, if any event is sent.

8. Save the process.
Startup and Shutdown Processes

Startup and Shutdown processes are executed by the Process Agent at the start and shutdown of TIBCO BusinessEvents Engine respectively.

**Startup Process**
When the TIBCO BusinessEvents engine starts up, it executes the startup process if defined.

**Shutdown Process**
When the TIBCO BusinessEvents engine shuts down, it executes the shutdown process if defined.

Consider the following conditions for a process when you configure the startup and shutdown process:

- Startup and Shutdown processes must have a start event with none.
- Startup and Shutdown processes should not have message and signal start events. They are activated on call to the process and do not depend on a event to start.
- Startup and Shutdown processes can have none, message and signal end events as they can notify the status to the external message end points.
- Startup and Shutdown processes can not have any receive message task, rule task and parallel join gateways.
- If a Startup or Shutdown process contains any Call Activity task, then the referred process in Call Activity task should meet the conditions of startup and shutdown processes.

Configuring Startup and Shutdown Process

Startup and Shutdown processes are configured for a process agent in the Cluster Deployment Descriptor (CDD) file. To configure the startup and shutdown processes for any process agent, perform the following steps:

1. Open the CDD file.
2. In the Agent Classes tab, expand the process agent node for which startup and shutdown processes need to be added.

3. Select the **Startup Processes** node to add a startup process. This process behaves as the startup process for a process agent.

4. Click Add button and select the process in the **Select Processes** dialog box. This process behaves as the startup process for a process agent.

5. Select the **Shutdown Processes** node to add a shutdown process.

6. Click Add button and select the process in the **Select Processes** dialog box. This process behaves as the shutdown process for a process agent.

7. Save the CDD file.
Chapter 4  

Process Agent Configuration And Runtime

This chapter describes the lifecycle of a business process execution at runtime. It explains the procedure of deploying and running the process agents.

Topics

- Process Execution (Runtime), page 28
- Adding and Configuring Process Agent Class, page 30
- Running Process Agent, page 31
Process Execution (Runtime)

LifeCycle of the process execution has the following characteristics:

- Every process definition is first compiled and converted into a process template class.
- A process template is instantiated for this process definition.
- The process template contains the process graph, task and sequence.
- Each task model from design time is created with a task ID.
- Process template registry is maintained.
- Process is executed.
- Upon receipt of an event, which is a start trigger point for the process, an implicit rule is sent to find the start event and match it with the process template. After that, a job is created from the process and the output mapper executed.

Task Execution

The result of Input mapping and Task Execution is mapped to the output mapper.

Checkpointing and Recovery

Check pointing persists process data to the backing store or cache for use in recovery. At a checkpoint, the values of all process variables are saved to the backing store or cache.

You can checkpoint individual tasks to provide more points of recovery. Checkpointing impacts performance due to the time it takes to persist the data. Checkpoint only important tasks whose state needs to be persisted in the event of a shutdown.

Values of system variables (such as id, _extid, and messageid) are persisted, as well as the values of user variables (those you add to a project). Stale checkpoint values are automatically replaced by fresh ones as a process instance executes, so that the latest checkpointed states are used during recovery.

Checkpointing can be implicit or explicit:

**Explicit Checkpoints**

Explicit checkpoints are done when the user configures it at individual tasks.
Implicit Checkpoints

The engine checkpoints job definitions, by version. Additionally, job instance states are checkpointed at the following points of execution:

- Parallel Gateway - At gateways, checkpoints are made:
  - After all forks.
  - Before all merges.
- Call Activity
- Assert Event - These implicit checkpoints are performed for the ‘Send’ events that are done via the ‘Send Message’ task in the process palette.

Recovery

When an engine restarts, only jobs that were active at the time of the shutdown are recovered. They are loaded from the backing store into memory.

Jobs that were passive during a shutdown remain in the backing store until activated, for example by a message arriving at an intermediate event, or on activation by a timer event.

Configure Explicit Checkpoint for An Activity

To enable checkpointing for any task, perform the following steps:

1. Select the task on the canvas.
2. Click the Properties > General tab.
3. Check the Checkpoint checkbox field. This enables the checkpoint to persist the process data for this activity to a backing store.
4. Save your project.

Locking and Unlocking

There is an implicit lock on the job when the process is ..... The developers are expected to take a lock in objects as needed.

- Parallel Gateway
- Job clustering

Checkpointing removes locking.
Adding and Configuring Process Agent Class

Defining a process agent class is similar to defining an inference agent class. The process agent configuration is defined in the CDD file. In the CDD file, you select the project resources that the agent will use at runtime, and configure other settings.

This section assumes familiarity with adding and configuring inference agent classes, as documented in *TIBCO BusinessEvents Developer’s Guide*. If you require a more detailed procedure please refer to that guide.

To Add a Process Agent Class in CDD File

1. In the Agent Classes tab, click **Add Agent**.
2. In the New Agent Class dialog enter an Agent Class Name and select the Process Agent Class Type from the list. Click **OK**.
3. The new agent class name appears on the left. Select the agent class name. Settings for the process agent type appear in the Configuration section.
   — Complete the settings as explained in **Table 4, CDD Agent Classes Tab Process Agent Settings**. Add any properties as needed.
4. Add the resources you want to use in each of the collections. The collection types available are the same as for an inference agent class, with the addition of process definitions. Ensure that all project resources used in the process definitions are included. For a detailed procedure, see *TIBCO BusinessEvents Developer’s Guide*.
5. If you added any individual destinations to the Input Destinations Collections category, highlight their name on the left and configure their settings on the right. See *TIBCO BusinessEvents Developer’s Guide* for details. (Destinations within input destination collections are configured at the Collections tab.)
6. Save the CDD file.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Active</td>
<td>Specifies the maximum number of active agents of this class. 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. A value of 0 indicates an unlimited number of active instances. Default is 0.</td>
</tr>
</tbody>
</table>

TIBCO BusinessEvents Process Orchestration Developer’s Guide
Running Process Agent

After the deployment configuration is defined for the Process Agent in the CDD file, run the process agent to execute the processes defined at the design time.

Build EAR file

Once you have defined the deployment configuration for the Process Agent in the CDD file, you can build the project EAR file within TIBCO BusinessEvents Studio as below:

1. Select the project.
2. Go to the menu bar, select **Project > Build Enterprise Archive**.
3. Type the value for the fields, as required on the **Build Enterprise Archive** Wizard. For example, **File Location** field specifies the location where the EAR file is built.
4. Click OK.
5. Verify that the EAR file is built at the location specified by the **File Location** field.

For a more detailed procedure at command line, Refer to **Building and Deploying EAR Files at the Command Line** chapter in the **TIBCO BusinessEvents Administration Guide**.

Run Process Agent

You can run the process agent in the following ways:

- Run using TIBCO BusinessEvents Studio
- Run at the Command Line
- Deploy And Run using TIBCO Administrator
Index

B
BE_HOME  xiii

C
complex event processing  
defined  2
customer support  xvi

E
ENV_HOME  xiii

F
functions documentation, accessing  xii

M
Max Active  30

S
support, contacting  xvi