TIBCO BusinessEvents™

Getting Started

Software Release 4.0.1
November 2010

The Power to Predict™

》TIBCO®
The Power of Now®

Important Information

Contents

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

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

Two new tutorials are added in this release:

-
-

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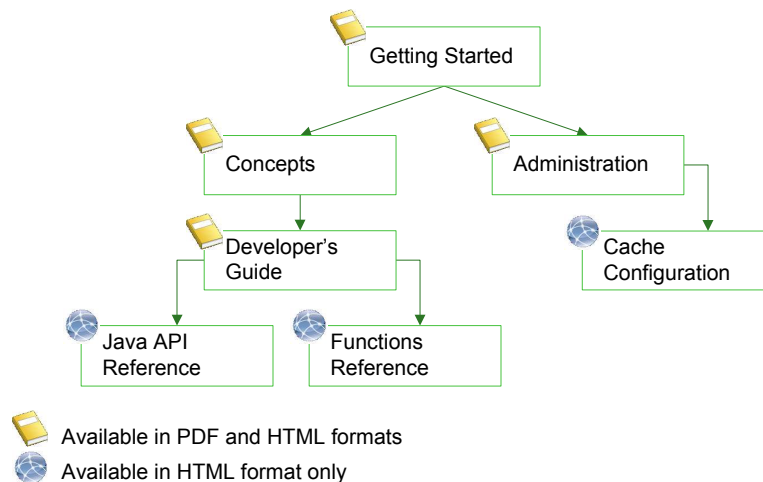
In addition to the above, the set contains an installation guide, release notes, and a readme file. The complete set is described next.

- *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
TIBCO_HOME *ENV_HOME* *BE_HOME*	Many TIBCO products must be installed within the same home directory. This directory is referenced in documentation as *TIBCO_HOME*. The value of *TIBCO_HOME* depends on the operating system. For example, on Windows systems, the default value is `C:\tibco`. 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 *ENV_HOME*. The value of *ENV_HOME* depends on the operating system. For example, on Windows systems the default value is `C:\tibco`. TIBCO BusinessEvents installs into a directory within *ENV_HOME*. This directory is referenced in documentation as *BE_HOME*. The value of *BE_HOME* depends on the operating system. For example on Windows systems, the default value is `C:\tibco\be\`4.0.
`code font`	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`	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)

Convention	Use
italic font	Italic font is used in the following ways:
	• To indicate a document title. For example: See *TIBCO BusinessWorks Concepts*.
	• To introduce new terms For example: A portal page may contain several *portlets*. 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: `MyCommand` *pathname*
Key combinations	Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C.
	Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.
	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
[ ]	An optional item in a command or code syntax.
	For example:
	`MyCommand [optional_parameter] required_parameter`
		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	param2	param3`

Table 2   Syntax Typographical Conventions

Convention	Use
{ }	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}`

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   **Introduction**

This brief chapter provides an overview of the tutorial provided in this guide.

Topics

Overview

This guide contains a tutorial based on one simplified business scenario. The tasks in the tutorial provide step-by-step instructions and also explain the main ideas so you gain understanding as well as practical knowledge. References to related information are provided so you can jump to the main documentation on any topic to learn more.

The tutorial shows you how to configure a BusinessEvents project, run it at the command line, and test its behavior. Using a simple scenario, this tutorial focuses on ontology and inferencing features.

A Configured Tutorial Project is Provided

The `FraudDetection` example is located in the *BE_HOME*/examples/standard directory.

Additional Example Projects Provide More Learning

After you have completed the tutorials, you can explore the examples in the *BE_HOME*/Examples directory. These examples demonstrate specific techniques that you can apply in your work.

Skills Required

The tutorial is written for users with little or no familiarity with TIBCO products. Readers should have some familiarity with Java programming and the Eclipse platform.

Eclipse Platform

If you are not familiar with Eclipse platform, it is recommended that you first read the topics available from the Overview link on the Welcome page, especially Workbench basics, which explains concepts such as perspectives and views. Then complete the "Create a Hello World Application" available from the Tutorials link on the Welcome page, and any other tutorials of interest.

The Fraud Detection Scenario

The tutorial is built on a simplified fraud detection scenario and decision making flow, illustrated by the following flow chart.

To establish whether fraud is suspected, the runtime engine correlates the frequency of and amount of debits in a rolling time window, and flags accounts that satisfy both of the following criteria:

- The account incurs more than three debit transactions in a two minute period.

- The sum of the debits that occurred in the two minute period totals more than 80% of the average monthly balance of the account.

For the purpose of the tutorial, messages arrive from an HTTP server, provided by BusinessEvents. The event correlation is performed using simple BusinessEvents rules. Suspicious accounts are simply set to "Suspended." Actions are printed to the monitor so you can see the project in action.

An additional event and rule not shown allow you to unsuspend an account.

```
┌─────────────────────┐
│ Customer makes a    │
│ debit transaction   │
└─────────────────────┘
           │
           ▼
┌─────────────────────┐
│ BusinessEvents      │
│ channel picks up a  │
│ debit transaction   │
│ message             │
└─────────────────────┘
           │
           ▼
       ◇ Account        yes   ┌──────────────┐
       suspended? ──────────→ │ Do not debit │
           │                  │ account      │
           no                 └──────────────┘
           ▼
┌─────────────────────┐
│ Debit the account   │
└─────────────────────┘
           │
           ▼
       ◇ Balance        yes   ┌──────────────────┐
       negative? ────────────→│ Set account status│
           │                  │ to "Suspended"    │
           no                 └──────────────────┘
           ▼
       ◇ Fraud          yes   ┌──────────────────┐
       suspected? ───────────→│ Set account status│
           │                  │ to "Suspended"    │
           no                 └──────────────────┘
           ▼
┌─────────────────────┐
│ Account status      │
│ remains active      │
└─────────────────────┘
```
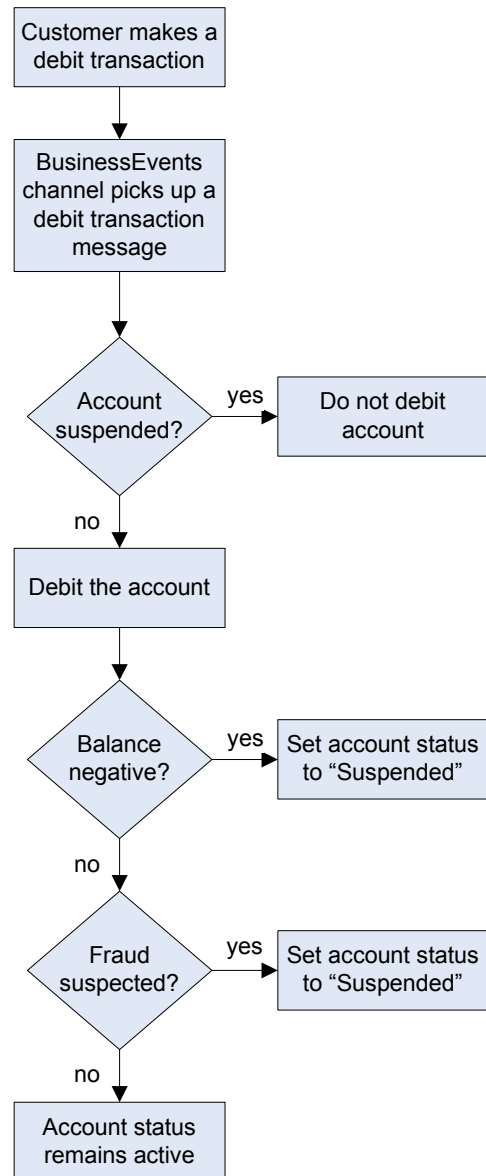
Chapter 2 **Project Design Tutorial**

This tutorial takes you through all the steps of configuring, building, deploying, and testing a BusinessEvents project. The emphasis is on basic project design, with deploytime activities limited to the basic actions required to test a design.

Topics

General Runtime Flow

The section The Fraud Detection Scenario on page 3 explains the general tutorial scenario, the fraud detection criteria, and how a customer account can become Suspended. This section explains in more technical terms what happens at runtime given an example debit that triggers the fraud detection rules. (You will learn more details about the terms shown in italics below, as you complete the tutorial steps):

1. A message arriving through a BusinessEvents *channel* is transformed into an *event*. (At design time you create an event type for this purpose, with the appropriate properties.) The event instance is then *asserted into the Rete network*, an in-memory network of objects based on the Rete algorithm which enables fast matching of facts with rule dependencies.

2. The presence of this new event in the Rete network causes the inference engine to check for rules that are designed to be triggered when this event is asserted.

3. A rule that has is triggered by this event executes. It may make changes to concept instances, create an event and send it to a channel (and out of the BusinessEvents application to some destination), and so on. The rule then consumes the event unless there is a reason not to do so.

Event lifespan It is important to consume events when they are no longer needed so that they don't trigger rules to fire erroneously. On the other hand, it is also important to use a long enough time-to-live (TTL) setting for an event, so that it exists long enough to perform all work needed, for example, to trigger rules that correlate multiple events and take appropriate actions.

Importing Existing Projects into Your Workspace

It is recommended that you complete the tutorials, because "learning by doing" is the most effective way to become proficient. If completing the tutorials is not possible, however, you can read the tutorials and refer to the fully configured example projects, provided here:

BE_HOME/examples/standard/FraudDetection

BE_HOME/examples/standard/FraudDetectionCache

Dependency of Readme File on a Higher Level Directory

The readme.html file uses resources in the *BE_HOME*/examples/_resources directory to enable you to send messages to the deployed example project using an HTTP channel. Therefore you must maintain the relative positions of these two directories.

Copy Example Directory Trees Before Importing

Because of this dependency between the readme.html file and the _resources directory (and also because it is good practice to work in copies of the shipped examples), it is recommended that you make a copy of the entire directory tree for any example you want to work with.

For example, copy *BE_HOME*/examples/standard/FraudDetection, and paste it as a peer of the provided example, then rename the top level directory, for example, to FraudDetection2. As an alternative you can copy the entire *BE_HOME*/examples directories and then work with the copied set of examples.

You can then make use of the readme.html file to test the deployed project.

At deploy time the only files that are used are the configured EAR and CDD files. When you construct a command to start an engine, these files must be available.

To Import an Example Project into Your Workspace

1. Start TIBCO BusinessEvents Studio. In Windows, click Start > All Programs > TIBCO > *YourEnvironment* > TIBCO BusinessEvents 4.0 > TIBCO BusinessEvents Studio.

2. As needed, create a workspace for your tutorial projects. Select **File > Switch Workspace > Other** and select the directory you want to use or create a new directory.

3. Select **File > Import > Existing Projects Into Workspace**, then click **Next**.

4. In the Select Root Directory, browse to a directory above the desired project directory and click **OK**.

    All TIBCO BusinessEvents Studio projects within the selected directory are listed.

5. Select the projects you want to import.

6. Check or uncheck the **Copy projects to workspace checkbox** to suit your needs. If you are working from a copy of the example project, as recommended, you do not have to copy projects to your workspace. It's up to you how you manage your files.

7. Click **Finish**. The selected projects appear in Studio Explorer.

Create the FraudDetection Project

In this task, you start TIBCO BusinessEvents Studio and create an empty project.
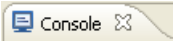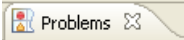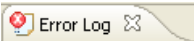
Learning Points   **What is TIBCO BusinessEvents Studio?** TIBCO BusinessEvents Studio is the Eclipse-based user interface for BusinessEvents. It enables you to build, test, and debug projects. Use of this industry-standard development framework shortens your learning curve and enables you to take advantage of common tools and facilities.

What are the BusinessEvents Perspectives? BusinessEvents has these Eclipse perspectives:

- TIBCO BusinessEvents Studio Debug
- TIBCO BusinessEvents Studio Development
- TIBCO BusinessEvents Studio Diagram

BusinessEvents switches perspectives transparently depending on the editor you are working with.

What are the bottom tabs? In the lower part of the user interface are various tabs used as needed:

Tab	Description
Console ⊠	**Console** The Console view displays and logs the errors resulting from improper execution of the BusinessEvents engine.
Problems ⊠	**Problems** The Problems view reports all validation errors in the project, including language validation errors, access control errors, and so on. It also displays messages relating to project analyzer. Double-click a problem entry to open the associated editor. See Analyze and Validate the Project on page 40 for more details.
Error Log ⊠	**Error Log** The Error Log view captures all the warnings and errors logged by Eclipse plug-ins. The log file itself has the extension .log file and it is stored in the .metadata subdirectory of the workspace.
Properties ⊠	**Properties** The Properties view shows metadata about a resource. To view the metadata, click the resource in Studio Explorer. Not all resources use this tab.

How should I organize project folders? When you create a new project a set of folders is created. You can use a different folder structure to organize your project components in any way you like. Example projects are kept simple and use provided folder names such as Concepts, Events, Rules. More complex projects might use a different folder hierarchy with names that relate to the purpose or contents of the folders.

Incorrect display? If the Eclipse user interface is not rendering correctly, try adjusting your computer display options. For example, on Windows XP, the following option causes rendering problems: Control Panel > System Properties > Advanced > Performance Options > Visual Effects > Adjust for best performance. In this example, choosing the Adjust for best performance option results in correct rendering.

More Information   Chapter 2, Project Tasks in *TIBCO BusinessEvents Developer's Guide* discusses actions you can take at the project level such as importing projects from an earlier release, validating projects, using project libraries, and so on.

Task A Create the Fraud Detection Project

1. Start TIBCO BusinessEvents Studio. In Windows, click Start > All Programs > TIBCO > *YourEnvironment* > TIBCO BusinessEvents 4.0 > TIBCO BusinessEvents Studio.

2. The first time you run TIBCO BusinessEvents Studio, a Welcome screen displays. You can access this guide from the Welcome page. Click the X next to Welcome to dismiss the screen.

3. As needed, create a workspace for your tutorial projects. Select **File > Switch Workspace > Other** and select the directory you want to use. You can create a new directory.

4. Right click in the TIBCO BusinessEvents Studio Explorer view (the panel on the left, where the project folders will display), and select **New > Project**. You see the New Project wizard.

5. From the list, select **TIBCO BusinessEvents > Studio Project**.

   Then click **Next**.

6. In the New Studio Project dialog, enter the project name **FraudDetection**, and click **Finish**. (If you want to use a non-default location, uncheck the Use default location checkbox and select the location.)

   In the Studio Explorer viewer, the root folder of the project is called FraudDetection. It has a set of project subfolders.

   In addition you may see links for various diagrams: FraudDetection.conceptview, FraudDetection.eventview and

FraudDetection.projectview. These are respectively concept model, event model, and project (or element) diagrams. Diagrams are created dynamically and not saved. More diagrams are available for other purposes. They are UML compliant, with some exceptions. Some diagrams are created only when generated.

7. Save the project.

    —   To save all changes to all resources in a project (since last save), click **File > Save All** or click **Ctrl+Shift+S**.

    —   To save changes in just the currently viewed resource, click **File > Save** or click **Ctrl+S**, or click the **Save** button.

Summary and Next Steps

You have created a new empty project in the TIBCO BusinessEvents Studio Development perspective.

Next you will begin to define your BusinessEvents project by building a channel for information to enter the deployed application, and a destination for the application to listen to.

The order in which you build up the project is not fixed. For example, instead you might define the project ontology first.

Create an HTTP Channel and Destination

In this task you configure an HTML channel with one destination. The AllOps destination listens for messages that come from HTTP forms embedded in the project's readme.html file.

Example projects use the HTTP channel because it does not require use of any external software. If you have TIBCO ActiveMatrix BusinessWorks, TIBCO Enterprise Message Service, TIBCO Rendezvous, or other source for messages you can experiment with adding different types of channels.

Learning Points

What are channels and destinations? Messages enter and leave the system through channels. You create destinations within a channel to define the message sources and sinks. BusinessEvents events are created using data in incoming messages. Outgoing messages are created using data from events. Later in the tutorial, you will set up the relationship between these destinations and the event types that they listen to by default.

Note that in this tutorial outbound messages are simply sent to the console, so there are no outbound destinations.

How are channels and destinations created? You create channels and destinations at design-time, as explained below. When you are planning BusinessEvents projects, you would consider the incoming and outgoing messages for your project, and then define the channels, destinations, and the corresponding event types — outbound events are transformed into appropriate messages, and inbound messages are transformed into events of a specified type.

Why Use Shared Resources? Shared resources are generally used to configure communication with some external system such as a database server or JMS server. You can configure the connection once and use it in multiple places. If some configuration has to be changed, you just have to change it in one place.

More Information

Chapter 5, Channels and Destinations, and Chapter 6, JMS Channels in *TIBCO BusinessEvents Developer's Guide*.

Task B Create the HTTP Connection

1. Select the SharedResources folder and press **Ctrl+N**. You see the Select a Wizard dialog. (You could also get here using File > New > Other.)

2. Select **TIBCO Shared Resources > HTTP Connection** and click **Next**.

3. In the New HTTP Connection Wizard, name the connection `HTTPConnection` and click **Finish**. (In a real world situation you would probably give the connection a more meaningful name.) You see the HTTP Connection dialog.

Resource names and directory names in the path to a resource can't be any of the keywords or other words listed in Chapter 17, Rule Language Grammar in *TIBCO BusinessEvents Developer's Guide*, and they can't use spaces.

4. In the Host field, enter `localhost`.

5. In the Port field, enter `8104`. If that port is not free, use an available port in the 8000 range.

6. Save the resource (click the save button in the toolbar) and close it.

Task C   Create an HTTP Channel and Destination
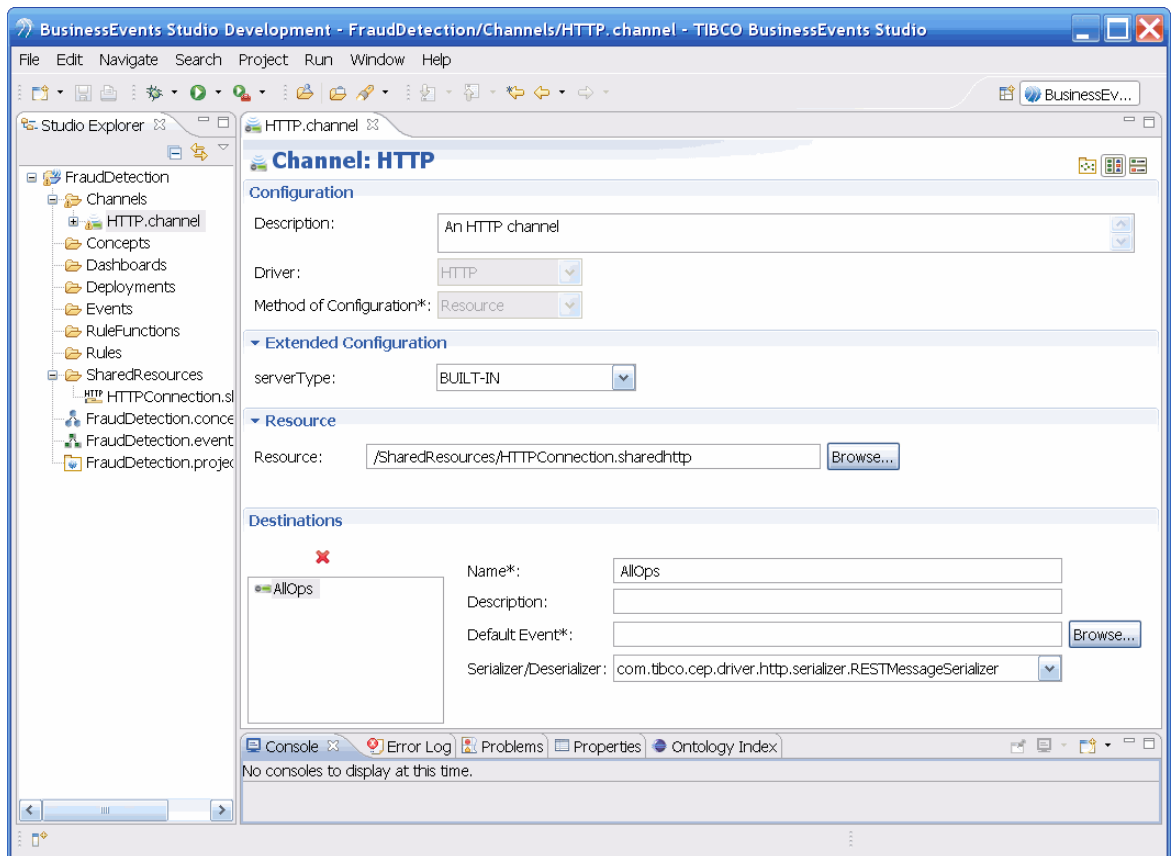
1. Right-click the **Channels** folder and select **New > Channel**.

2. You see the New Channel Wizard. In the Channel Name field, type **HTTP**.

    a. In the Description field, type **An HTTP channel**.

    b. In the Driver Type field, select **HTTP**.

    c. Click **Finish**. You see the Channel editor

In BusinessEvents, you can't change any new project resource name after you click Finish in the new resource wizard. (You can change the description, however.) The resource name displays in the title of the resource editor, for example, Channel: HTTP.

3. In the Extended Configuration section, Server Type field, select **BUILT-IN**.

4. In the Resource field, browse to and select the HTTP connection resource you created in Task B. Only valid shared resources for the current resource display.

5. In the Destinations section, click **Add** and name the destination **AllOps**. Leave all other fields set to their default values.

Provided Examples Requirements

- HTTP and AllOps are required names for examples. The readme.html uses the AllOps destination in its embedded forms.

- Adding a default event is not necessary for the provided examples because the readme.html, which starts an HTTP channel, specifies the event. However for other use cases, default events are convenient: all messages arriving at a destination are transformed to the destination's default event, unless the message specifies a different event. The default event in this channel is not required, but it does stop the warning sign from appearing! — the warning sign just lets you know when a destination does not have a default event.

6. Save and close the resource.

Summary and Next Steps

Now you have built a channel and a destination within that channel to listen for messages. The next step is to create some events.

Define the AccountOperation, CreateAccount, Debit, and Reply Events

In this task, you begin to build the project ontology by defining some events — or strictly speaking, event *types*. Before you define event types in a real-world project, you first examine the incoming and outgoing messages, as well as messages that you want to occur within the application, and configure each event type's characteristics accordingly. You can use inheritance (as demonstrated here) to simplify configuration.

BusinessEvents provides various kinds of events. Simple events are used in this tutorial to bring messages into the application. In addition you can use SOAP events, time events and advisory events, which you can learn about in the product documentation.

Learning Points

What is an event? The term *event* is overloaded: it means an activity that happens, and the definition of an object that represents the activity in BusinessEvents (an event type), and an instance of that event definition.

How are events (event instances) created? Simple event types are created at design time. Event instances are generally created using data in incoming messages. When a destination receives a message, it creates an event to hold the information from the message. Events from channels are automatically asserted into the Rete network, where their presence generally triggers rules (if all rule conditions are met).

Simple events can also be created by rules and rule functions. Events created this way are not asserted automatically because they could be outbound events, to be sent to a destination. You must explicitly assert such internally created events as needed.

Applications that Use Concurrency Features Require Use of Locking An important function of event preprocessors is locking, to ensure that multiple concurrent RTCs (whether in the same agent or different agents in a cache cluster) do not work with the same object at the same time, or read an out of date version of the object. See Using Locks to Ensure Data Integrity Within and Across Agents in *TIBCO BusinessEvents Architect's Guide* to understand how to use locking correctly.

What is an event payload? Just as messages have properties and a message body, events can have properties and payloads. The payload is optional. It is used to hold more complex data, for example, SOAP messages. (The events in this example do not use a payload.)

What is a default destination? Outbound events of the same event type are often sent to the same destination. To simplify the process of sending those events, you can specify a default destination in the event type.

Why do the events have this warning sign? Debit.event   The yellow triangle with exclamation point means something may not be right. A red triangle means there is an error. The yellow triangle is shown for events with no default destination. In this case, however, no events are sent out through channels so no default is required.

What is a default event? The default event configured for a destination is used to hold information transferred from an incoming message, when no event type is specified in the message.

See Default Destinations and Default Events in *TIBCO BusinessEvents Architect's Guide* for more details. (The FraudDetection project does not use these features.)

Rules that apply to a parent type also apply to its child types Concept and event types use inheritance in a similar way to Java classes. Because *rules that apply to a parent type also apply to its child types*, it is generally not advisable to create many levels of inheritance. However it can be a useful technique. In this tutorial the `AccountOperations` event is the parent of both the `CreateAccount` and `Debit` events. You'll see why in a later section.

More Information

- Chapter 2, Channels and Events in *TIBCO BusinessEvents Architect's Guide* for overview and conceptual information.

- Chapter 8, Simple Events, Chapter 9, Time Events and Scheduler Functions, and Chapter 10, Advisory Events in *TIBCO BusinessEvents Developer's Guide* for implementation details.

Task D   Define the AccountOperations Event

This event is a parent to events that are used in the project. It has one property: `AccountId`. All its child events inherit this property, and extend the parent by adding more.

1. Right click the **Events** folder, and select **New > Simple Event**.

2. You see the New Simple Event Wizard. In the Simple Event Name field, type **AccountOperations**. In the Description field, type **Parent event for all account-related events**. Click **Finish**.

3. You see the Simple Event Editor. In the **Properties** section, click the **Add** button. Click in the cell under Name and type the name **AccountId**. It's a `String` property, and `String` is the default type.

4. Save and close the resource.

Task E Define the CreateAccount Event

For example purposes, the information needed to create an account is an ID, a balance, and a monthly average balance (for the fraud detection calculation). The ID is inherited from the AccountOperations event.

1. Right click the **Events** folder, and select **New > Simple Event**.

2. In the New Simple Event Wizard Filename field, type **CreateAccount**. In the Description field, type **Triggers the CreateAccount rule to create an account**. Click **Finish**.

3. In the Simple Event Editor Inherits From field, click Browse. In the upper section of the event picker, select the **Simple Event** event type, and in the lower section browse to and select the **AccountOperations** event. Click **OK**. The AccountOperations event is now the parent event for the CreateAccount one.

4. In the Default Destination field, click the browse button and in the Select Destination dialog, select **/Channels/HTTP.channel/AllOps**. Click **OK**.

5. In the **Properties** section, add two properties:

   — **Balance**, of type **double**

   — **AvgMonthlyBalance**, of type **double**

6. Save and close the resource.

Task F   Define the Debit Event

Following the procedure above, add the Debit event:

Field	Value
Name (Defined in the wizard)	`Debit`
Description (Defined in the wizard and editor)	`Specifies an account (by inheritance) and a debit amount`
Inherits From	`AccountOperations`
Properties	`Amount` (double)

Task G   Define the Unsuspend Event

Following the procedure above, add the Unsuspend event:

Field	Value
Name (Defined in the wizard)	`Unsuspend`
Description (Defined in the wizard and editor)	`Triggers the UnsuspendAccount rule to change the customer status from Suspended to Normal`
Inherits From	`AccountOperations`
Properties	(No properties)

Summary and Next Steps   Next you will continue to configure the project ontology by defining a concept.

Define the Account Concept

In this task, you define the Account concept, which holds basic information about an account: an ID, a balance, an average monthly balance, and an account status. You also learn some useful information about concepts and how they are used.

Learning Points

What is a concept? A concept type is a definition of a set of properties that represent the data fields of an entity. Concept types are like Java classes, and concept instances are like Java objects.

How are concept instances created? Concept instances are created by rules and rule functions. Information from event properties or payloads is often used to create concept instances, but other information can be used, for example, the results of a query or a calculation.

What is a database concept? A BusinessEvents add-on product, TIBCO BusinessEvents Data Modeling, provides a feature that enables you to create concepts by importing them from a database. A set of functions enables you to update the database record to account for changes made in BusinessEvents. Unlike regular concepts, database concept instances are not asserted to the Rete network automatically, and they don't track history. (This add-on also provides a state modeler functionality.)

How can I persist concept instances? Instances of concepts (and events) are also known as "facts" and "entities." They can be persisted in various ways, generally using a cache and backing store, as determined by the business need. Later tutorials explain these features.

How is history tracked? When the History setting for a concept property is 0 (zero) the current value is stored without a date-time stamp. When the history setting is 1, the current value is stored, along with the date and time the value was added or changed. When the history value is greater than 1, BusinessEvents tracks changes to property values up to the specified number (using a ring buffer). The Policy setting additionally determines what values are recorded, all values or only changes to the prior value.

You'll set the Debits property history, to track "All Values," that is, BusinessEvents records the value of the property every time an action sets the value, even if the new value is the same as the old value — a person can debit the account twice by the same amount. For a property such as "address" you might want to track only changes to the value.

More Information

- Chapter 3, Concepts in *TIBCO BusinessEvents Architect's Guide*.

- Chapter 11, Concepts in *TIBCO BusinessEvents Developer's Guide*.

- *TIBCO BusinessEvents Data Modeling Developer's Guide*

Task H Define the Account Concept
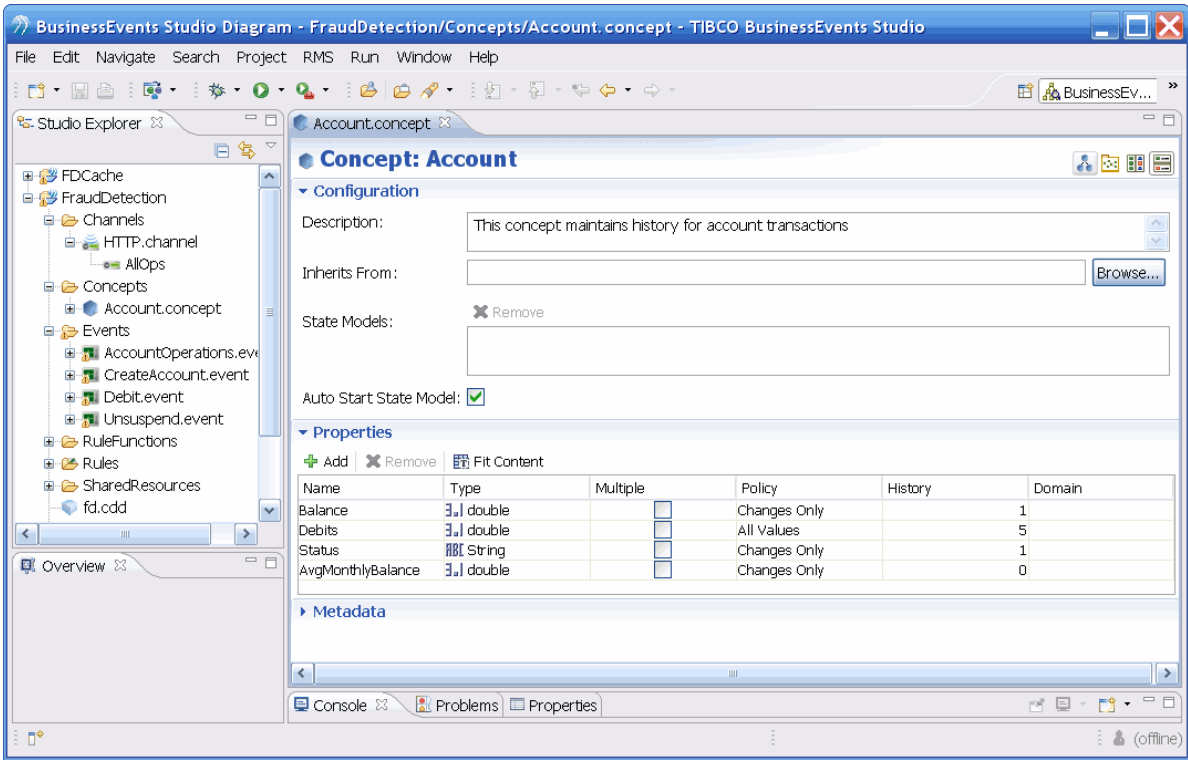
1. Right click the **Concepts** folder, and select **New > Concept**.

2. You see the New Concept Wizard. In the Filename field, type **Account**. In the Description field, type **This concept maintains history for account transactions**. Click **Finish**.

3. In the Account Concept Editor Properties section, add the following properties:

Name	Type	Policy	History
Balance	double	Changes Only	1
Debits	double	All Values	5
Status	String	Changes Only	1
AvgMonthlyBalance	double	Changes Only	0

The Multiple field is used to define an array property. In this release, domains are used only with TIBCO BusinessEvents Decision Manager, a BusinessEvents add-on product.

You may wonder where the account ID from the incoming event will be stored. It will go into the concept's extId attribute.

Concept Attributes and Concept Relationships

Attributes Concepts, events, and scorecards have some built-in attributes, in addition to the properties you define here. The attribute extId, referenced as @extId, will hold the account ID.

The extId must be unique across the cluster It's important to note that the optional extId attribute, if used, must be unique across all objects in the cluster. This attribute is used for events as well as for concepts. For example If you use database concepts (available in TIBCO BusinessEvents Data Modeling add on) you may expect to use the primary key from the database as the extId. However, that would not be possible if more than one table contains the same columns in its primary key.

Concept Relationships Concepts can have containment and reference relationships with other concepts. You set these up as concept properties, and define the kind of relationship by selecting an appropriate data type for the property. For example, a car concept can contain Wheel concepts, and can have a reference relationship to a Dealership concept.

Inheritance Concepts can also inherit from other concepts (and events can inherit from other events). Inheritance is a programming relationship and not an ontology relationship. It enables you to extend a base type for more efficient project management. See Define the AccountOperation, CreateAccount, Debit, and Reply Events on page 15 for important information about inheritance.

4. Save and close the resource.

Summary and Next Steps You have defined a concept type to hold information about bank accounts. The last step in building the ontology of your project is to set up a scorecard to hold fraud detection criteria that are used in rules.

Add the FraudCriteria Scorecard

In this task, you finish building the project ontology by creating a scorecard. The `FraudCriteria` scorecard will store the criteria used to determine fraud, not any specific data about customer accounts. In this example, you will use this scorecard in rules.

Learning Points

What is a scorecard? A scorecard is a special type of concept. A scorecard serves as a static variable. You can use a scorecard resource to track key performance indicators or any other information. Unlike concepts, there is only one instance of a scorecard. You create the scorecard at design time. Its values can be viewed and updated using rules.

It is more accurate to say there is one instance of a scorecard per inference agent. This tutorial uses one inference agent. However, in the next tutorial you will deploy multiple agents, and each has its own instance of the scorecard. This enables scorecards to be used for local purposes and minimizes contention between the agents. Do not use scorecards as a mechanism to share data between multiple agents.

More Information

- Chapter 12, Scorecardsin *TIBCO BusinessEvents Developer's Guide*.

Task I   Create the FraudCriteria Scorecard

There is only one scorecard in this project, so we don't need a folder for it. You will create it in the project root.

1. Select the FraudDetection root folder and press **Ctrl+N**. In the Select a Wizard dialog, expand **TIBCO BusinessEvents** and click select **Scorecard**.

   This is how you add resources that are less commonly used.

2. Name the scorecard `FraudCriteria`. Add the description `Stores the criteria used to determine fraud`, and click **Finish**.

3. In the FraudCriteria Scorecard editor, add the following properties:

Name	Type	Policy	History
interval	long	Changes Only	0
num_txns	int	Changes Only	0
debits_percent	double	Changes Only	0

4. Save and close the resource.

Summary and Next Steps

You have now set up the ontology for the project, that is, the definitions of all the events, concepts, and scorecards that are needed to store information (facts) about possible fraud detection.

Next, you will configure a rule function that sets values for the scorecard, and one that acts as an *event preprocessor* — a term you will learn about in that section.

Add the InitializeScorecard Rule Function

In this task, you configure a rule function that initializes values for the FraudCriteria scorecard. This rule function is used at system startup. (You'll configure that connection later.)

Learning Points

What is a rule function? A rule function is a function you write in the BusinessEvents rule language.

How are rule functions created and used? You write rule functions using the rule editor. You can use rule functions in rules and other rule functions, in event preprocessors, and as startup or shutdown functions for an agent.

What other types of functions are there? BusinessEvents provides a large library of functions for various purposes. In addition, an ontology function is automatically created for each concept and event in your project. These are constructor functions. Another type of ontology function enables you to create and schedule a time event.You can also add custom Java functions.

More Information

- Chapter 4, Rules and Functions in *TIBCO BusinessEvents Architect's Guide*.

- In *TIBCO BusinessEvents Developer's Guide*, the following chapters:

  — Chapter 15, Rules and Rule Functions

  — Chapter 16, Functions

  — Chapter 17, Rule Language Grammar

  — Chapter 18, Rule Language Datatypes

  — Chapter 19, Mapping and Transforming Data

  — Chapter 20, XPath Formula Builder

Task J   Add the InitializeScorecard Rule Function

1. Right click the **RuleFunctions** folder, and select **New > Rule Function**

2. You see the New Rule Function Wizard. In the Filename field, type **InitializeScorecard**. In the Description field, type **Sets values in the FraudCriteria scorecard**. Click **Finish**.

   You can work in the Source view or the Form view, according to your preference. The tutorial uses the Form view. In the lower area, click the Form tab to switch to the Form view.
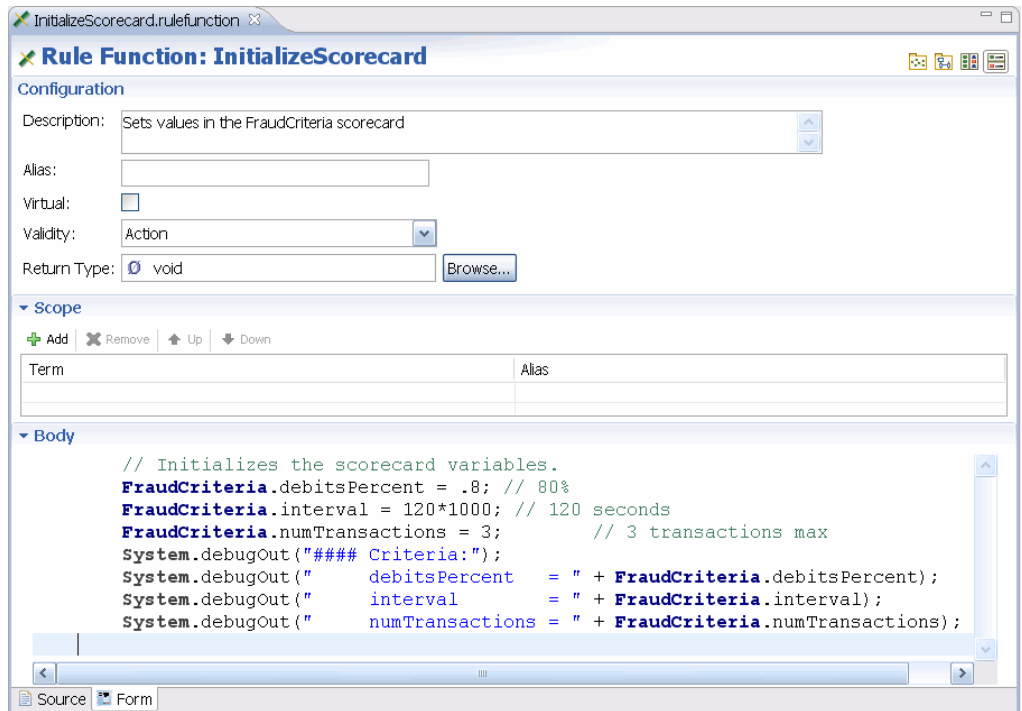
- **Rule Function Editor Preference** To set the default mode, go to Window > Preferences > BusinessEvents > Rules and check or uncheck the following checkbox as desired: **Initially show 'Form' tab in Rule Function Editor**.

- Ctrl-click the name of a rule function used in a rule or rule function editor to bring up the editor for that rule function.

3. Leave the Scope area empty, because this function is used at startup. In the Body area, add the following lines to provide values to the FraudCriteria scorecard (and to comment your code):

```
//Intialize scorecard variables
FraudCriteria.debits_percent =.8;
FraudCriteria.interval       = 120*1000; /* 120 seconds */
FraudCriteria.num_txns = 3;
```

Notice that when you type the period after FraudCriteria, a list of its properties appears so you can select a property.

4. Save and close the resource.

Summary and Next Steps   You've set up a startup rule function. Next you'll set up an event preprocessor rule function.

Add the PreProcessor Rule Function

In this task, you configure a rule function that replies to the request received from the HTTP channel. HTTP is a request-reply protocol, and this step is required so that the HTTP server is ready to process the next request from the readme.html form. This rule function executes when an event is received. (You'll configure that connection later.)

Learning Points   **What is an event preprocessor?** An event preprocessor is a rule function that processes incoming messages before BusinessEvents transforms them into events. In this case the preprocessor is used to send a response to the HTTP server. In real-world applications, however, a preprocessor might filter the messages so that only certain ones are used as events, and it might do other event enrichment actions. Preprocessors are multi-threaded and you can choose from various threading and queue options, as appropriate to handle the work load. By default the threading uses the system-wide shared queue and threads. See the topic Event Preprocessorsin *TIBCO BusinessEvents Architect's Guide*.

More Information   • Event Preprocessors in *TIBCO BusinessEvents Architect's Guide*.

Task K Add the PreProcessor Rule Function

1. Right click the `RuleFunctions` folder, and select **New > Rule Function**

2. You see the New Rule Function Wizard. In the Filename field, type `PreProcessor`. In the Description field, type `Closes requests from the HTTP server`. Click **Finish**.

3. Click the **Form** tab at the bottom of the editor.

4. In the Scope section, click **Add**.

5. You see the Select Rule Function Scope Arguments dialog. In the Types area (at the top), select **Event**. Click **OK**.

   In the Select Resource area, you could select a specific event type in the project. However, here we want any event to be in the scope of this rule function, not one specific event type.

6. In the Alias column (in the Scope section), replace the default alias (e) with **Request**.

7. In the Body area, type: `Event.` (Event followed by a period). Notice that when you type the period (.) you see a list of all catalog functions in the Event category. Use the down arrow to scroll down the list of functions and stop at

replyEvent. Its tooltip displays. Documentation for all catalog functions is provided in tooltips.

Function:	*Event.replyEvent*
Signature:	boolean replyEvent(SimpleEvent request, SimpleEvent reply)
Synopsis:	Replies with a reply SimpleEvent to a request SimpleEvent. If a reply destination on the request SimpleEvent is specified, it will be used to send the reply SimpleEvent, else no action is taken.
Parameters:	

Name	Type	Description
request	SimpleEvent	The original request SimpleEvent.
reply	SimpleEvent	The Reply SimpleEvent.
Returns:	boolean	true if the SimpleEvent is sent; false otherwise.
Cautions:		
Example:		

The tooltips are also reproduced in the HTML version of the product documentation, in the Online References area.

8. Click the **replyEvent** function to select it. Now the body looks like this:

    Event.replyEvent(

    As you can see, the rule function arguments are two events, a request event and a reply event.

9. To specify the request event, type **Request**, the alias for the scope argument you added in step 5.

10. To specify the reply event, just type **request** again. The reply event can be any event in this case, so we can simply reply with the request event.

Summary and Next Steps

You have configured a rule function that will send a reply to requests sent by the HTTP server (through the HTTP channel). Next you will configure rules that take action on assertion of CreateAccount and Debit events, depending on various conditions.

Add BadCreateAccount and CreateAccount Rules

In this task you create two rules, one called `CreateAccount` and one called `BadCreateAccount`. Both rules can fire when a `CreateAccount` event is asserted into the Rete network. However, the `BadCreateAccount` rule has a higher priority, so it will fire before the `CreateAccount` rule.

The `BadCreateAccount` rule checks whether the account ID provided in the `CreateAccount` event matches the account ID of any Account instance already in the Rete network. One of the two following situations must occur:

* A matching ID exists in the Rete network: the `BadCreateAccount` rule prints a message to the console, and "consumes" — that is, deletes — the `CreateAccount` event. Because that event is consumed, the `CreateAccount` rule cannot fire.

* No matching ID exists in the Rete network: the `BadCreateAccount` rule does nothing, and then the `CreateAccount` rule fires, and creates the `Account` concept instance.

Learning Points   **What are rules and how are they created?** Rules define actions to take when certain conditions are met. Rules are written in the BusinessEvents rule language, which is similar to the Java language. Rules are declarative and are generally narrow in scope. A rule has three parts: the declaration (`declare`), the conditions (`when`), and the actions (`then`). As with rule functions, you can work in a source view, which displays the Java-like code, or in a form view.

How are rules used at runtime? The rule engine checks all changes and additions to the Rete network and evaluates or reevaluates rules, using their declaration and conditions, as needed. Eligible rules are added to the *rule agenda*.

What is the rule agenda   A rule fires when it is at the top of the agenda. The engine determines the order of firing using rule declarations and conditions, and each rule's priority and rank (if these features are used). As the contents of the Rete network change, the engine reevaluates rules and removes any that are no longer eligible to fire. See Understanding Conflict Resolution and Run to Completion Cycles in *TIBCO BusinessEvents Architect's Guide* for details.

How can you prioritize rule execution? The Priority setting is used by the runtime engine when determining the order in which rules are fired. Those with a number closer to one fire first. Within a set of rules that has the same priority, a ranking feature enables you to determine which fire before others. It uses a callback rule function that enables you to specify business logic to establish the rank. When there is no reason to force rules to execute in a particular order, leave the Priority and Rank fields set to the default and let the runtime engine determine rule order.

More Information
- See all references provided for Add the InitializeScorecard Rule Function on page 24
- Chapter 5, Run-time Inferencing Behavior in *TIBCO BusinessEvents Architect's Guide*.

Task L   Add the BadCreateAccount Rule

1. Right click the **Rules** folder, and select **New > Rule**.

2. You see the New Rule Wizard. In the Filename field, type `BadCreateAccount`. In the Description field, type `Checks for an existing account with the specified ID.` Click **Finish**.

3. In the Form tab, set the Priority field to **3**. It's higher priority than 5, the default. You'll set the `CreateAccount` rule to priority 5, so that the `BadCreateAccount` rule always fires before `CreateAccount` rule.

4. Expand the Declaration section as needed so you can see empty rows below the headings Term and Alias.

   Each of the sections can be expanded and contracted as needed.

5. Drag the Account concept from the Studio Explorer tree into the first empty row in the Declaration section. You see the project path of the Account concept in the Term column, and account in the Alias column.

6. Similarly, drag the `CreateAccount` event into the next available row.

   Declaration The Declaration provides the scope of the rule. It lists all the entity types to be used in the rule, and their aliases. By default the alias is set to the entity name in lower case letters. You can change it as desired.

7. In the Conditions panel, type the following:

```
//Checks whether the extId of an Account instance in working memory
//matches the incoming event's account ID

account@extId == createaccount.AccountId;
```

   Notice that when you type the At sign (@), a pick list of concept attributes appears. Attributes are built-in. You can't add or remove attributes. The `id` attribute value is set internally. However you can set the external ID, `extId`.

8. In the Actions panel, just below the Conditions panel, type these statements:

```
System.debugOut("#### Account already exists: " + createaccount.AccountID;
Event.consumeEvent(createaccount);
```

These actions are done only if the conditions are met. If not, then the next rule in the agenda fires — and that is likely to be the CreateAccount rule, which you'll define next.

You are also consuming the event, so it can't trigger any other rules.

9. Save and close the resource.

Task M Add the CreateAccount Rule

1. Right click the **Rules** folder again, and select **New > Rule**.

2. In the New Rule Wizard Filename field, type **CreateAccount**. It doesn't really need a description does it? Click **Finish**.

3. In the Form tab, set the Priority field to **5**. (This is a lower priority than the BadCreateAccount rule.)

4. Drag the CreateAccount event from the Studio Explorer tree into the first empty row in the Declaration section.

   This rule has no conditions — the BadCreateAccount means there is no need. You could have combined the two rules into one. There are many ways to write rules for a project. You have to use your judgment.

   In the Actions panel, you'll use an ontology function to create the Account concept instance. You can also get to ontology functions using the function catalog.

5. From the top menu select **Window > Show View > Other > TIBCO BusinessEvents > Catalog Functions**. The Catalog Functions view displays on the right (unless your Eclipse IDE is configured differently — it could display along the bottom, for example).

6. Below each concept, event, and rule function is its ontology function. Expand **Ontology Functions > FraudDetection > Concepts > Account > Account**:
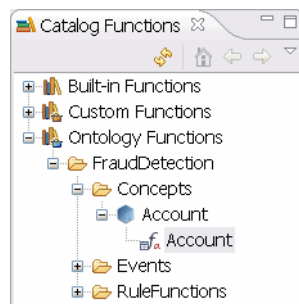
7. Drag the Account function into the Actions section. You see its signature:

```
Concepts.Account.Account(/*extId String */,/*Balance double */,/*Debits double
*/,/*Status String */,/*AvgMonthlyBalance double */)
```

8. Configure the function to create the Account concept, and type the rest of the actions as follows. You can double click the tab to expand the rule editor, to make rule writing easier.

```
Concepts.Account.Account(createaccount.AccountID/*extId String */,
        createaccount.Balance/*Balance double */,
        0/*Debits double */,
        "Normal"/*Status String */,
        createaccount.AvgMonthlyBalance/*AvgMonthlyBalance double */);

Event.consumeEvent(createaccount);

System.debugOut("#### Created account " + createaccount.AccountID);
```

9. Save and close the resource.

Note that in this case it is not really necessary to consume the event. No other rules have this event in their scope. Events with time to live (TTL) set to zero (0) are consumed at the end of an RTC. However it does no harm and makes the rules more consistent, reducing chances of error.

Add ApplyDebit, BadApplyDebit, and CheckNegativeBalance Rules

The rules you add in this section work together as follows:

When a `Debit` event is asserted into the Rete network, BusinessEvents checks rules with the Debit event in their scope.

- The `ApplyDebit` rule has the Debit event in its scope, and also an account ID. It's priority 1 so it will execute before any other lower priority rule. The engine checks the rule conditions.

  — If the Rete network also contains an Account instance whose ID matches the ID in the debit event, the rule executes. If the account is suspended, a message to that effect displays in the console. Otherwise, the account is debited. A message displays in the console to this effect.

  — If the Rete network does not contain a matching Account instance, then other rules in the agenda — those that have debit events in their scope — are eligible to execute. `BadApplyDebit` is the only rule with the Debit event in its scope, so it is eligible and it executes. It sends a message to the console that no matching account is found.

- The `CheckNegativeBalance` rule has an `Account` concept in its scope. When an account is debited the Account concept is changed, and so the `CheckNegativeBalance` rule becomes "newly true." The effect is similar to assertion of a new entity into the Rete Network. If the debit has made the account balance negative, this rule sets the status to Suspended.

Learning Points It's important to understand how conflict resolution and run to completion (RTC) cycles work. If you understand what triggers rules to execute, and why a rule may not execute, you can design rules more effectively. For a reminder, carefully reread Understanding Conflict Resolution and Run to Completion Cycles in *TIBCO BusinessEvents Architect's Guide*.

More Information
- See all references provided for Add the InitializeScorecard Rule Function on page 24
- Chapter 5, Run-time Inferencing Behavior in *TIBCO BusinessEvents Architect's Guide*.

Task N Add ApplyDebit, BadApplyDebit, and CheckNegativeBalance Rules

If you have completed Task L, Add the BadCreateAccount Rule and Task M, Add the CreateAccount Rule, you will be familiar with adding rules. This section shows the source code for the three rules you will add. As an extra hint, screenshot of the ApplyDebit rule is also shown below

ApplyDebit Rule Form View

The main purpose for showing the form view is so you can compare it with the source view when creating your own rule.
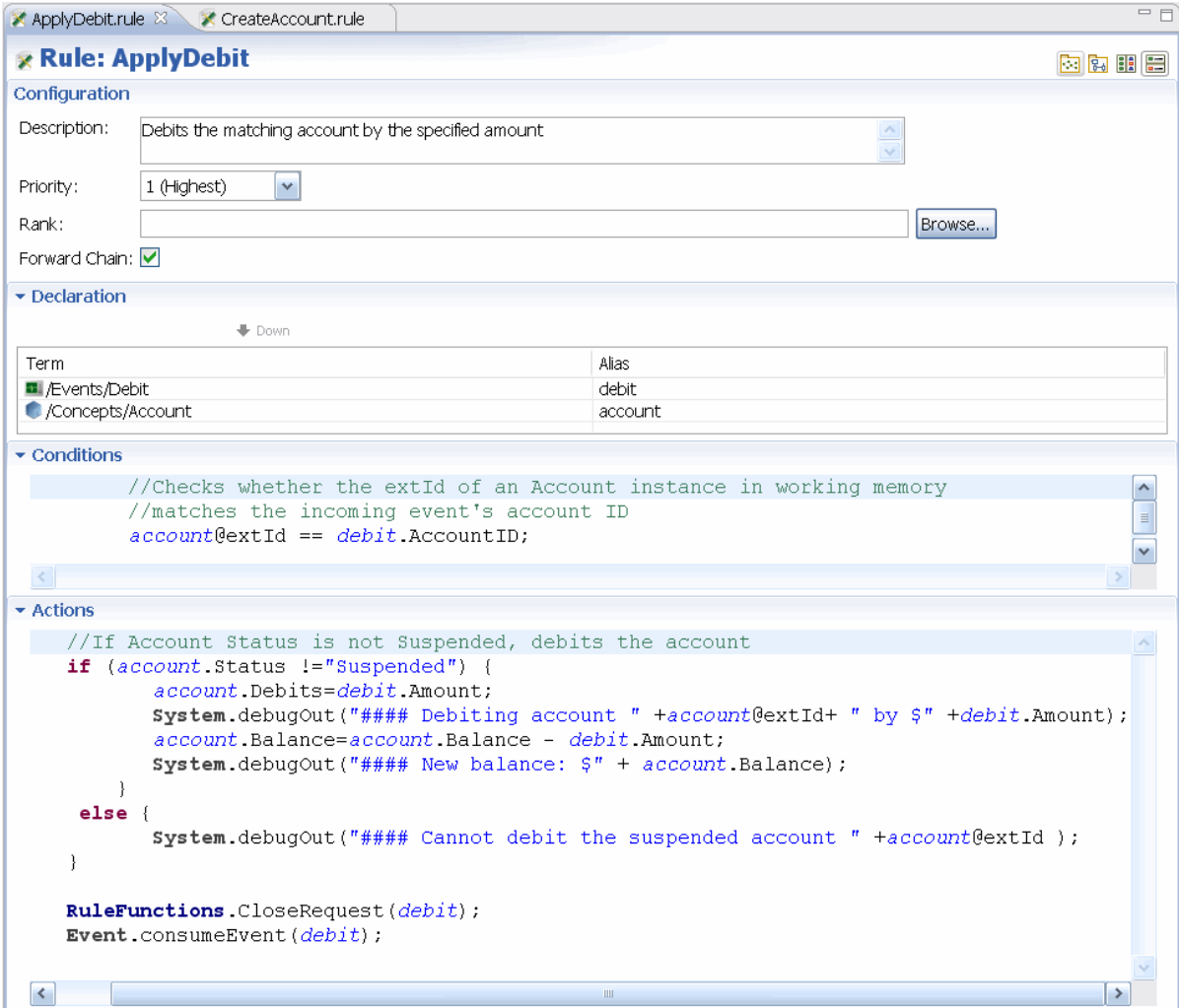
ApplyDebit Rule Source View Code

```
/**
 * @description Debits the matching account by the specified amount
 * @author
 */
rule Rules.ProcessDebits.ApplyDebit {
  attribute {
    priority = 1;
    forwardChain = true;
  }
  declare {
    Events.Debit   debit;
    Concepts.Account   account;

  }
  when   {
    //Checks whether the extId of an Account instance in working memory
    //matches the incoming event's account ID
    account@extId == debit.AccountId;
  }
  then {
    //If Account Status is not Suspended, debits the account
    if (account.Status !="Suspended") {
        account.Debits=debit.Amount;
        System.debugOut(
            "#### Debiting account " +account@extId+ " by $" +debit.Amount);
        account.Balance=account.Balance - debit.Amount;
        System.debugOut("#### New balance: $" + account.Balance);
        }
    else {
      System.debugOut(
          "#### Cannot debit the suspended account " +account@extId );
    }

    Event.consumeEvent(debit);
  }
}
```

BadApplyDebit Rule Source View Code

```
/**
 * @description
 * @author
 */
rule Rules.BadApplyDebit {
  attribute {
    priority = 5;
    forwardChain = true;
  }
  declare {
    Events.Debit  debit;

  }
  when {

  }
  then {
    System.debugOut(
       "#### Debit not applied, account not found: " + debit.AccountId);
  }
}
```

CheckNegativeBalance Rule Source View Code

```
/**
 * @description
 * @author
 */
rule Rules.CheckNegativeBalance {
  attribute {
    priority = 5;
    forwardChain = true;
  }
  declare {
    Concepts.Account  account;
  }
  when {
    //Checks that the balance is less than zero
    account.Balance < 0;
    //Checks that Account status is not set to Suspended
    account.Status!="Suspended";
  }
  then {
    account.Status="Suspended";
    System.debugOut(
       "#### Account ID "+account@extId+" STATUS set to Suspended. Balance"
+account.Balance+" is less than zero");
  }
}
```

Add the FraudDetection Rule and Unsuspend Account Rule

You are an experienced rule builder now! Add the FraudDetection rule, typing in the Source view or Form view, according to your preference. The source is shown below.

Most of the code is in the conditions. The first condition checks whether the number of debits in the specified interval prior to the current time is greater than the specified number of debits. The interval and the number of debits are set in the FraudCriteria scorecard.

The second condition checks whether the sum of all debits in the verification interval is greater than the specified percentage of the account average monthly balance. The specified percentage is set in the FraudCriteria scorecard. The average monthly balance, for the purposes of this tutorial, is set in the Account instance created by the CreateAccount rule.

You will also add a rule called UnsuspendAccount. This is a convenience rule that allows you to change a customer status from Suspended to Normal at run-time.

Learning Points **How are conditions processed?** All conditions in a rule must be met, before the action is done. That is, each condition is joined by an implied AND operator.

In what order are conditions evaluated? To learn more the effect of filters, equivalent join conditions, and non-equivalent join conditions on the efficiency of a rule, see Order of Evaluation of Rule Conditions in *TIBCO BusinessEvents Architect's Guide*. Understanding these points helps you design an efficient project.

How can I learn about all these catalog functions? BusinessEvents provides hundreds of catalog functions for use in rules and rule functions. You can use functions you already know about by typing the beginning of the name and then using the completion hints that appear. To learn about more functions, you can open the Catalog Functions view and browse. To see the tooltip for a function, hover the mouse over the function name. You can then drag a function into the editor, as you did in Task M, Add the CreateAccount Rule. As a reminder, here's how to open the Catalog Functions view: **Window > Show View > Other > TIBCO BusinessEvents > Catalog Functions**. The tooltips are also available in HTML form, in the Online References section of the HTML documentation for the product.

More Information

- See all references provided for Add the InitializeScorecard Rule Function on page 24

- Chapter 5, Run-time Inferencing Behavior in *TIBCO BusinessEvents Architect's Guide*.

Task O Add the FraudDetection Rule

```
/**
 * @description
 * @author
 */
rule Rules.FraudDetection {
  attribute {
    priority = 5;
    forwardChain = true;
  }
  declare {
    Concepts.Account   account;

  }
  when {
    //1. Checks the number of debits in the verification interval
    Temporal.History.howMany(account.Debits,
          DateTime.getTimeInMillis(DateTime.now())-FraudCriteria.interval,
          DateTime.getTimeInMillis(DateTime.now()),
          true)
        > FraudCriteria.num_txns;

    //2. Checks the percentage of the average balance that was
    //   debited in the verification interval
    Temporal.Numeric.addAllHistoryDouble(account.Debits,
          DateTime.getTimeInMillis(DateTime.now())-FraudCriteria.interval)
        > FraudCriteria.debits_percent*account.AvgMonthlyBalance;

    //Check whether Account status is not set to Suspended
    account.Status!="Suspended";
  }
  then {
    account.Status="Suspended";
    System.debugOut("#### Account ID "+account@extId+" STATUS set to Suspended.
Fraud suspected.");
  }
}
```

Task P Add the UnsuspendAccount Rule

```
/**
 * @description
 * @author TIBCO Software Inc
 */
rule Rules.UnsuspendAccount {

  attribute {
    priority = 1;
    forwardChain = true;
    }
```

```
declare {
  Concepts.Account account;
  Events.Unsuspend request;
}

when {
  account@extId == request.AccountId;
  account.Status == "Suspended";
}

then {
  account.Status = "Normal";
}

}
```

Summary and Next Steps
Congratulations! You have now configured the project's ontology and rules. Now you are ready to configure the Cluster Deployment Descriptor and build the archive for deployment. But before you do, it's wise to validate and analyze the project, and look at it in the Project diagram.

Analyze and Validate the Project

In this task, you check the project for errors. If you have configured it correctly, you'll see warnings but no errors. Warnings indicate potential issues, such as absence of default events. It's always worth checking them. In the FraudDetection project, the warnings do not indicate any actual issues.

Learning Points **How can I validate that there are no basic errors in my code?** To perform checks that don't take project logic into account, use Project > Validate.

What if there are problems in my code? You can debug it. BusinessEvents builds on the Eclipse debugger features to provide additional checks. Developers familiar with the Eclipse debugger will find it intuitive to use.

How can I analyze that the logic of my project is OK? For this kind of checking you use Project Analyzer or the Element Diagram or both. They work together to provide insights into your project. Project Analyzer is a document generated in the Problems view area. Element Diagram provides a visualization of the whole project or of selected project elements. It also shows all dependencies in a project. Project Analyzer and Element Diagram can be configured to run separately or together, using preferences.

How can I visualize different aspects of a project? BusinessEvents provides diagrams similar to UML sequence diagrams, class diagrams, and dependency diagrams. They are modified a little from the strict UML to be more useful in the context of TIBCO BusinessEvents Monitoring and Management.

How can I change the names of project resources, or move them around? Use the refactoring features. They ensure that changes you make are reflected throughout the project (with certain limitations).

More Information In *TIBCO BusinessEvents Developer's Guide*, see these sections:

- Chapter 4, Element Refactoring Operations

- Chapter 23, Testing and Debugging Projects in *TIBCO BusinessEvents Developer's Guide*

- Chapter 24, Diagrams, especially Project Analyzer and Selected Entity Project Diagrams, in

Task Q Analyze and Validate the Project

1. In Studio Explorer, highlight the project name, then from the top menus select **Project > View**. You see the project visualization in the main editor window.
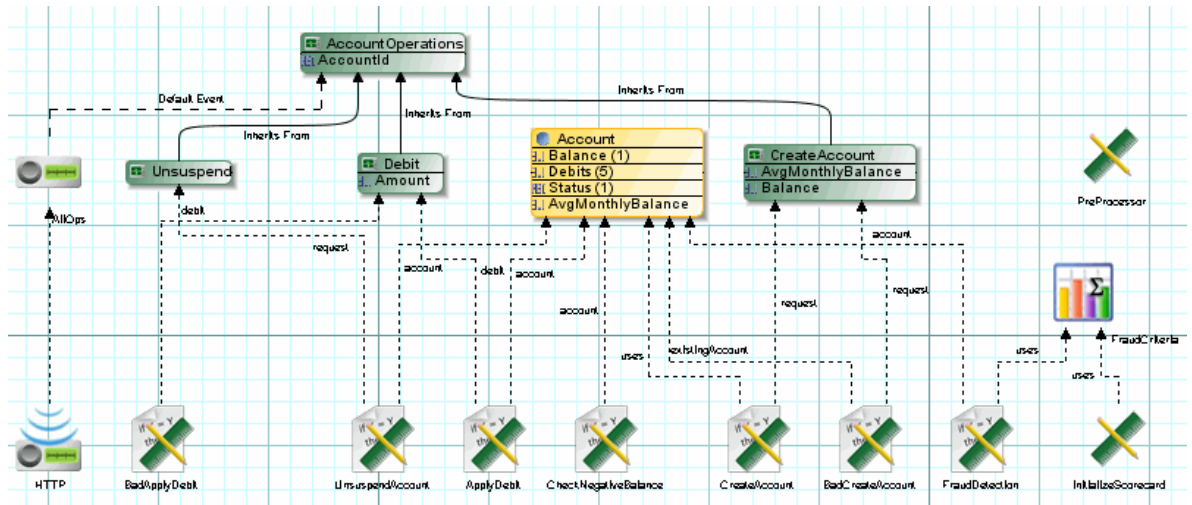
You can see the dependencies and relationships between the project resources. You can jump to a resource editor by clicking the resource's icon.

2. Click the Problems tab to see the project analysis.

The Type columns shows the type of check that resulted in the warning. You can sort by any column. Here are some warnings taken during project development. They are sorted by Resource and some columns are sized smaller to make others more visible.

Description	Resource ▲	Path	Location	Type
Warnings (14 items)				
⚠ Event /Events/AccountOperations does not have any destination configured fo	AccountOperations.event	FraudD...	Unkno...	Project Analyzer Problems
⚠ A default destination is not specified	AccountOperations.event	FraudD...	Unkno...	Resource Validation Problem
⚠ Rule /Rules/ApplyDebit may never fire (it has unused event in its scope)	ApplyDebit.rule	FraudD...	Unkno...	Project Analyzer Problems
⚠ Rule /Rules/BadApplyDebit may never fire (it has unused event in its scope)	BadApplyDebit.rule	FraudD...	Unkno...	Project Analyzer Problems
⚠ Rule /Rules/BadCreateAccount may never fire (it has unused event in its scope)	BadCreateAccount.rule	FraudD...	Unkno...	Project Analyzer Problems
⚠ Event /Events/CreateAccount does not have any destination configured for it	CreateAccount.event	FraudD...	Unkno...	Project Analyzer Problems
⚠ A default destination is not specified	CreateAccount.event	FraudD...	Unkno...	Resource Validation Problem
⚠ Rule /Rules/CreateAccount may never fire (it has unused event in its scope)	CreateAccount.rule	FraudD...	Unkno...	Project Analyzer Problems
⚠ Event /Events/Debit does not have any destination configured for it	Debit.event	FraudD...	Unkno...	Project Analyzer Problems
⚠ A default destination is not specified	Debit.event	FraudD...	Unkno...	Resource Validation Problem
⚠ The Event URI in the Destination "AllOps" of Channel "HTTP" is empty.	HTTP.channel	FraudD...	Unkno...	Resource Validation Problem
⚠ Destination AllOps in Channel /Channels/HTTP does not have a default event	HTTP.channel	FraudD...	Unkno...	Project Analyzer Problems
⚠ Rule Function /RuleFunctions/InitializeScorecard is never used	InitializeScorecard.rulefun...	FraudD...	Unkno...	Project Analyzer Problems
⚠ A default destination is not specified	Reply.event	FraudD...	Unkno...	Resource Validation Problem

3. Close the `FraudDetection.projectview` tab. Diagrams are not saved. They are generated when needed.

Summary and Next Steps Now you are ready to configure the Cluster Deployment Descriptor (CDD) and build the archive for deployment.

Add a Cluster Deployment Descriptor and Build the EAR File

To deploy a project you need a CDD file and an EAR file. The CDD is not included in the project EAR. This means you can reconfigure a project's deployment configuration without having to rebuild the EAR.

Learning Points

What is an EAR The Enterprise Archive or EAR file contains details for all the resources in a project, and project global variables.

What is a CDD The project's deployment configuration is defined in an XML file called the Cluster Deployment Descriptor, or CDD. You edit this file using the TIBCO BusinessEvents Studio Cluster Deployment Descriptor editor.

How do I set up preprocessors and startup and shutdown rule functions? The CDD is where you configure rule functions to act as event preprocessors, startup rule functions, or shutdown rule functions! Only rule functions whose Validity setting is Action are valid for these uses. Such rule functions cannot require anything to be in their scope, because they execute outside of the context of the Rete network and BusinessEvents project resources.

What is an object management type? Object management refers to how BusinessEvents manages the objects generated within the BusinessEvents application at runtime. For some applications, the objects are more important than for others. This tutorial focuses on the basics, so it uses In Memory, the simplest OM type. Objects are kept in memory only, and are lost when the engine stops.

What is an inference agent? An agent does certain work within the engine. With Cache OM, different types of agents do different work. But for this simple In Memory project, you use only one agent of one type, and that is the inference agent. Inference agents listen for messages arriving at destinations, and transform them into events. The events trigger rules, using the agent's Rete network and forward chaining, and the inference agent executes the rules.

What is a processing unit? A processing unit is deployed as a BusinessEvents engine. It has one or more agents and it runs in one JVM. For In Memory OM, you generally don't have to do much or any configuration of processing units. The default processing unit settings are sufficient.

More Information

In *TIBCO BusinessEvents Administration*, see these chapters:

- Chapter 3, CDD Configuration Procedures
- Chapter 4, Cluster Deployment Descriptor Reference

Task R Add and Configure a CDD

1. In Studio Explorer, right click the project name, FraudDetection, and select **New > Cluster Deployment Descriptor**. You see the New Cluster Configuration Wizard.

   You can create multiple CDD files for a project and at deploy time use the one that has the configuration you want to use.

2. In the File name field, type **fd** and click **Next**.

   Unlike other project resources, you can change the name later as desired. Short names are easier to type when starting the engine at the command line.

3. At the Template Selection page, select **In Memory** from the Object Management Type drop-down list. Then click **Finish**.

   You see the CDD editor, displaying the template for In Memory OM. For an In Memory project, very little deployment configuration is required.

4. Click the Collections tab, and do the following:

   a. Select **Destinations** and click **Add**.

   b. In the Destinations Collection field, type Http.

   c. A folder called Http appears in the Destinations node in the Collections panel. Select the **Http** folder and click **Add**. In the Select Destinations dialog, select **/Channels/HTTP/AllOps** and click **OK**.

   d. In the Select Destinations dialog, select **/Channels/HTTP/AllOps** and click **OK**.

   e. In the Configuration panel Destination ID field, type **AllOperations**, replacing the generated ID.

   f. In the Preprocessor field, select /RuleFunctions/PreProcessor.

   g. In the Threading Model field select **Caller**.

   For the HTTP channel, the Caller threading model is required.

The Configuration panel looks like this:
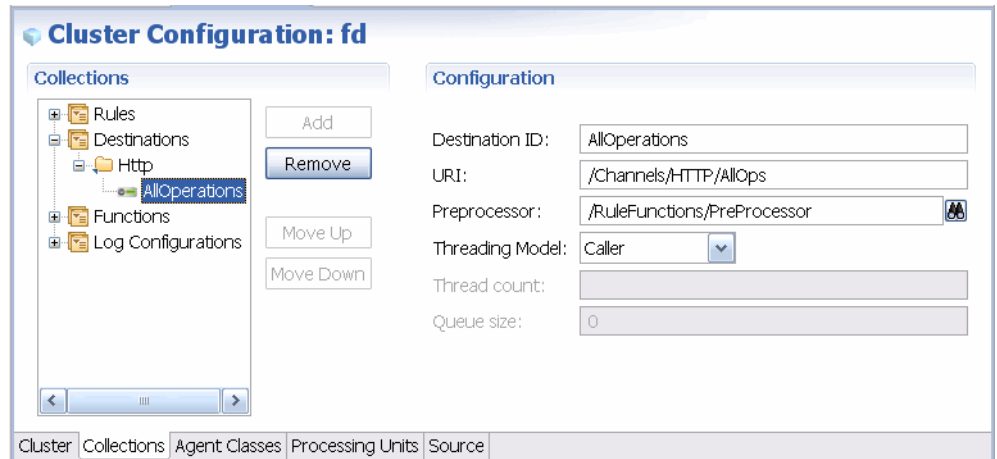
Collections enable you to create resources you can reuse when configuring multiple agent classes. Collections are used here to demonstrate the feature. In simple projects, you could simply configure the agent class without using collections. Configure the collection as follows:

5. Click the Agent Classes tab and expand the default agent class, which is called `inference-class`.

   When configuring an agent class, you can select a subset of the project rules, select and configure various destinations, and select rule functions that execute at engine startup and shut down, as needed. Thus different agent classes can behave quite differently at runtime.

6. Select **Destination Collections** and click **Add**. You see the Select Destinations dialog. Under Reference Collections, select **Http** and click **OK**.

   As mentioned above, you can configure destinations here, or refer to destinations configured in the Destination Collections tab.

7. Select **Startup Functions** and click **Add**. In the Select Rule Functions dialog, select **/RuleFunctions/InitializeScorecard** and click **OK**. When the engine starts, this rule function executes and initializes the scorecard values.

8. Save and close the CDD.

Task S   Build the EAR File

You must build the EAR file outside the project tree, or it will be recursively included in the next EAR file you build!

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

   If you see a message asking you to save all project resources, click Yes. It
   means an unsaved resource editor is open.

   At the Build Enterprise Archive dialog, you can change the name to fd. (Short
   names are easier to type when starting the engine at the command line.)

2. In the File Location field, browse to and select the directory above the project
   directory. (To build the EAR in the provided example location, you would
   choose *BE_HOME*/examples/FraudDetection/FD.ear. Replace *BE_HOME*
   with your actual value.)

   Where you build the EAR is not so important. You just have to specify the
   correct location when starting the engine at the command line.

3. Press Apply, then press OK. You see messages as the EAR file builds, then you
   see a message that the EAR file has built correctly:
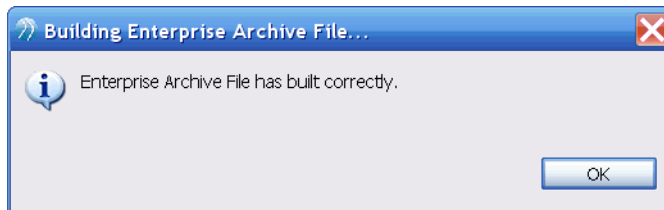
Summary and Next Step   You are ready to deploy the FraudDetection project!

Start the Engine and Run the Application

You can simply open the example readme.html file and follow instructions there. In order to send events into the engines, you must enter information into the forms provided in the readme.

To Start the Application at the Command Line

For your information, here is how to start the engine, using the default locations specified for this example. If you have built the example in a different location, as is recommended, please adapt the instructions. See Dependency of Readme File on a Higher Level Directory on page 7 to understand the requirements.

Open a command window and at the command prompt, start the BusinessEvents engine using the following command. Substitute your actual value for *BE_HOME*:

```
cd BE_HOME/examples/FraudDetection

BE_HOME/bin/be-engine --propFile BE_HOME\bin\be-engine.tra -u default -c
FraudDetection/fd.cdd fd.ear
```

The format for a command-line is as follows:

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

For a detailed explanation of the above format see Starting a TIBCO BusinessEvents Engine at the Command Line in *TIBCO BusinessEvents Administration*.

Congratulations!

You have completed the tutorial and now understand the basics of project design and deployment. Next you can undertake the tutorial provided in Chapter 3, Cache OM Tutorial, on page 49.

Chapter 3 **Cache OM Tutorial**

This tutorial shows you how to add caching functionality to your project, and how to deploy two cache agents and one inference agent.

This tutorial uses the fraud detection project configured in Chapter 2, Project Design Tutorial, on page 5.

Topics

Cache OM Tutorial Overview

Cache OM is used in most production systems because of the additional functionality, reliability, and high availability it provides, especially when a backing store is also used to store the data generated by the system on disk.

To explore basic caching features, you will make some minor modifications to the FraudDetection project used in the Project Design Tutorial. Modifications are required to retrieve objects from cache, or to check that objects do not already exist in cache before creating them.

Then you'll deploy two engines each with one agent: an inference agent and a cache agent. You will observe how stopping and starting the inference agent does not affect how events are handled, because the objects are stored in the cache. You can optionally start two cache agents and one inference agent, to see how the cache agents handle engine failure and restart. You can also start a second cache agent to observe how two cache agents work together.

Before you begin the tutorial, take a few minutes to learn a few important points about cache object management, in the next section, BusinessEvents Cache Fundamentals on page 51. Setting up cache OM is fairly simple. Understanding how to manage cache objects and how to design for concurrency take some thought.

Note that the FraudDetectionCache project contains a site topology file, `FraudDetectionCache.st`, for use in Chapter 4, TIBCO BusinessEvents Monitoring and Management Tutorial, on page 59.

BusinessEvents Cache Fundamentals

Read this section to grasp the basic ideas you need to understand and work with cache object management. For greater depth, see *TIBCO BusinessEvents Architect's Guide*.

What is object management? Object management (or OM for short) refers to various ways that BusinessEvents can manage the objects created by the actions of an inference agent's rules and rule functions. You define the OM options in the project's CDD file. The objects created are instances of the types defined in the ontology—concepts, scorecards and events. In the Project Design Tutorial, you used *In Memory OM*. With In Memory OM, when an engine shuts down, all its data is lost. This option is useful for a project where objects are not created or don't have to be persisted, for example, a project that routes incoming events to different BusinessEvents projects for handling.

What is a cache cluster? When you use Cache OM, objects are persisted redundantly in memory caches. Each BusinessEvents engine (JVM) participates as a node in the cache cluster. The cache manager manages the objects across the JVMs.

What is a distributed cache? BusinessEvents ships with a preconfigured distributed cache scheme. In a distributed cache, the cached object data is partitioned between cluster nodes (BusinessEvents engines). Each object is stored redundantly in two or more nodes, for reliability and high availability: if one engine fails, the objects are repartitioned across the remaining engines so that the configured number of backups is maintained.

What is a backing store? With a distributed cache, one or more engines can fail without loss of data (depending on the number of backups of each object). However if all engines fail, the data is lost. For this reason it is common to write the objects to a database, known as the backing store. BusinessEvents provides tools to create the backing store for a project. When you use a backing store you can also use a *limited cache* and populate it as needed from the backing store, balancing performance and memory requirements as needed.

What are cache agents? Cache agents store and serve the cache data for the cluster. Cache agents participate in distribution, partitioning and storage of the objects in the cluster. An engine (processing unit) that runs a cache agent can't run any other agents of any type. (For testing you can configure an inference agent to *also* act as a cache agent, but this is not recommended for production systems.)

What other types of agents are there? Besides cache agents and inference agents, two other types of agents can join the cluster, too. Query agents are used in the TIBCO BusinessEvents Event Stream Processing add-on. Dashboard agents are used in TIBCO BusinessEvents Views. (Add-ons are available separately from BusinessEvents.) A single engine can run multiple agents of these types.

How do agents use cached data? The inference agents can use objects in the cache as well as those they create from data coming in through channels (or create internally using rules and rule functions). Cached concept objects are shared by all agents in the cluster, scorecards are kept local to an agent, and events are clustered between the agents.

How does data get from the Rete network to the cache? At the end of each RTC— each "run to completion" cycle—data is flushed from the Rete network to the cache. In this way the Rete network does not become clogged with irrelevant data. (This behavior occurs when the entities are configured as cache-only, which is the default and is strongly recommended.

How does data get from the cache into the Rete network? Before each RTC, you must load the relevant data into the Rete network from the cache. If you don't then cached data that is needed by the rules triggered by incoming events is not present. For example, a request to create a new account arrives through a channel. It is important to check that no matching account already exists in the cache before you create the new account. To do so, you use an event preprocessor rule function to load any matching accounts into the Rete network. Then in the RTC rules can check for existing accounts and avoid creating duplicates.

How should I use Cache + Memory mode? With cache plus memory mode, data is not flushed from the Rete network automatically. This option is to be used only for constants or entities that change very seldom. Concurrency management is problematic with this mode, because of the difficulty in keeping all the agent Rete networks in all engines synchronized.) Instead, use cache-only mode for all or most of your needs.

How do I change the number of backups of each object are kept? By default one backup of each cache object is maintained. Backups are maintained in different cache agents. You can increase the number of backups by adding this property to the Cluster tab properties sheet:
`tangosol.coherence.distributed.backupcount`

How does BusinessEvents deal with concurrency and locking? Multiple inference agents can run concurrently, sharing the same ontology and same cache cluster. Another way to achieve concurrency is to use the multi-threaded Rete feature, which also requires cache OM. With agent or RTC concurrency, you must use locking: in both cases multiple RTCs are being processed at the same time, and data must be protected as in any concurrent system. The tutorial does not demonstrate concurrency features. See Chapter 9, Concurrency and Project Design in *TIBCO BusinessEvents Architect's Guide* for more details.

Rename or Copy the FraudDetection Example

This tutorial builds on the FraudDetection tutorial. As a starting point for the tutorial, you need to do one of the following:

• Use the FraudDetection project you configured in Chapter 2, Project Design Tutorial, on page 5.

• Use a copy of the provided FraudDetection example project.

If you don't want to build the Fraud Detection Cache (FDCache) project yourself, you can import the provided FDCache example to see how it is configured.

The tutorial assumes you will rename the basic FraudDetection project you configured earlier.

See Dependency of Readme File on a Higher Level Directory on page 7 to understand how to organize your files so that you can send events to the running engine.

Learning Points   **What is refactoring?** When you make changes to the structure of a project element, such as renaming or moving the element, all references to that element are updated accordingly. This operation is known as *project refactoring*. Some refactoring type checks are also done for copy-paste operations.

More Information   • Chapter 4, Element Refactoring Operations in *TIBCO BusinessEvents Developer's Guide*

Task A   Rename the Fraud Detection Project

1. As needed import the FraudDetection project (or a copy of it) into your workspace. See Importing Existing Projects into Your Workspace on page 7.

2. Highlight the project name and select File > Rename. Type the name **FraudDetectionCache**.

3. Save the project.

Summary and Next Steps   Next you will configure the project to load objects from the cache into the Rete network before the start of an RTC.

Update the Event Preprocessor to Load the Rete Network

After each RTC, the data created in the agent is saved to the cache and removed from the Rete network. Before the next RTC, therefore, you must reload any data needed by any rules that will be triggered in the RTC. This is generally done using the event preprocessor rule function, as it is the presence of event data that triggers most rules.

Learning Points   **What is a concept instance's mode?** Concepts can be set (globally or individually) as Memory Only, Cache Only, or Cache + Memory (cache plus memory). These modes refer to how the data is saved. Memory Only mode means the instances are not saved to the cache. See also BusinessEvents Cache Fundamentals on page 51 to understand more about cache mode.

Loading data from cache into the Rete does not trigger rules At least not in the way it would if the data was newly arriving in an event. In conjunction with other data, however, the presence of the loaded data can trigger rules, in the usual way.

More Information   In *TIBCO BusinessEvents Architect's Guide* see the following:

- Chapter 8, Cache Modes and Project Design
- Understanding Conflict Resolution and Run to Completion Cycles

Task B Update the Event Preprocessor

1. In TIBCO BusinessEvents Studio, open the **RuleFunctions > PreProcessor** rule function.

2. Click the **Source** tab at the bottom of the editor to work in the Source view. (In the Project Design Tutorial you worked in the Form view. You could stay in that view, but it's good become familiar with both views).

3. In the Scope section, replace Event request with **Events.AccountOperations request**. The section looks like this:

```
scope {
   Events.AccountOperations request;
}
```

The reason you have to narrow the scope is that the base Event class does not have an AccountId property, so it can't be used in the function you'll add in the next step.

4. In the Body section, add two new lines (shown in bold), just before the closing bracket:

```
body {
   // Replies to the request event, in order to close the HTTP request.
   // To keep it simple, uses the request event as the response.
   Event.replyEvent(request, request);

   // Attempts to load any existing matching account.
   Coherence.C_CacheLoadConceptByExtId(request.AccountId, false);
}
```

The Coherence.C_CacheLoadConceptByExtId() function loads any matching items from the cache into the Rete network. In this case, it loads any concept whose ExtId matches the AccountId in the incoming event. The loading is done before the event is asserted, so the Rete network will contain any matching Account concepts. The BadCreateAccount rule then fires and as in the Project Design Tutorial, to prevent creation of duplicate accounts.

Summary and Next Steps

You have modified the basic FraudDetection project to load instances from the cache into the Rete network using an event preprocessor. This enables the project code to check for existing instances first, when creating new ones.

Next you will work in the Cluster Deployment Descriptor to create two agents of different types.

Add a CDD File for Cache Object Management

To set up the project for cache object management, you create a new CDD file using the template provided for cache OM. The CDD template defines an inference agent class and a cache agent class, and default values for cluster level properties, so it doesn't take much to configure an example project to work. Other configuration options are available for different situations.

Learning Points **How do I configure cluster node discovery?** You can use the default multicast node discovery settings provided in the CDD template. You can also change the defaults as needed to avoid collisions with another cluster in the same network. Cluster discovery settings are available for different situations, such as hosts with multiple NICs. Another method of cluster discovery uses well-known addresses for situations where use of multicast is not an option.

How do I configure agents and processing units? You configure agent classes and processing units in the CDD. A deployed processing unit is a BusinessEvents engine. For this example, you don't have to do any processing unit configuration. However, for real-world scenarios, where concurrency features are used and perhaps add-on products, processing unit configuration requires some thought and planning. The processing unit definitions are also used in the TIBCO BusinessEvents Monitoring and Management component, for deploy-time configuration.

Number of cache agents to start For a production environment you can define how many cache agents must be started before the system startup can continue. The default is one. See the Cache Agent Quorum setting in the Object Management Configuration panel. When a backing store is used, cache agents can preload objects from a backing store at system startup. More cache agents make the preloading phase quicker.

More Information In *TIBCO BusinessEvents Developer's Guide* see the following:

- Chapter 3, CDD Configuration Procedures
- Chapter 4, Cluster Deployment Descriptor Reference

Task C Add and Configure a CDD

Start with a new CDD because the provided Cache OM template sets many default values you'll need.

1. In Studio Explorer, right click the project name and select **New > Cluster Deployment Descriptor**. You see the New Cluster Configuration Wizard.

2. In the File name field, type **fdCache** and click **Next**.

If you click Finish instead of Next, you don't have the opportunity to use the Cache OM template. If you accidentally click Finish, just delete the file and add a new one—and this time click Next.

By default, the CDD file name you specify is also used as the cluster name.

3. You see the Object Manager Selection dialog. Select **Cache** from the Object Management Type drop-down list, then click **Finish**. You see the CDD editor.

4. In the Cluster tab, select **Properties**. Expand the **coherence** property group. This group of properties with default values is provided in the Cache OM CDD template. If you need to change any defaults, configure the properties as follows.

   a. If it is possible to collide with any other project deployed on your network, use a non-default multicast IP address (the default is 224.3.5.2).

   b. If the default `tangosol.coherence.localport` value, 8088, is in use, specify a different value. (The shipped project uses 8080.)

   c. Because you will start all nodes on one machine for the tutorial, it's a good idea to set the `tangosol.coherence.ttl` to 0, to keep multicast packets from leaving the originating machine. (Works on most operating systems.)

5. Exactly as you did in the basic design tutorial, do the following:

   — Click the Agent Classes tab and configure the `inference-class` agent class with the `Channels/Http/AllOps` destination

   — Configure the `RuleFunctions/InitializeScorecard` rule function as the startup rule function.

   For details, see Task R, Add and Configure a CDD on page 44. and complete step 4 through step 7.

   No configuration is needed for the cache-agent class.

6. Save and close the CDD.

7. Build the EAR file. (Details are in the section Add a Cluster Deployment Descriptor and Build the EAR File, page 43.)

Summary and Next Steps   You have configured the CDD for cache cluster discovery, and you configured the processing units and agents as you did before. Now you can start the project and see how things work at runtime.

Deploy the Agents and Cache Servers and Send Events

It is simplest in this step to just run the provided example and follow instructions in its readme.html file.

The HTTP channel used to send events is provided by the example's `readme.html` file and by resources in the *BE_HOME*/examples/_resources directory.

You can recreate this setup in a different project location. If you do, the `_resources` directory must be in the same relative position to the `readme.html` as in its shipped location and you must adjust the commands according to your directory structure.

Learning Points   **Cache agents start first** It is important that cache agents start before inference agents. Set the Cache Agent Quorum setting in the Object Management Configuration pane to ensure that enough cache agents start before other types of agents can begin processing. In general, at startup, you start cache agents then other types of agents. At system shutdown, you stop other kinds of agents, then stop cache agents.

Task D Start the BusinessEvents Agents and Send Events

If you are working in a copy of the examples that preserves the relationship between the `readme.html` file and its resources (see Dependency of Readme File on a Higher Level Directory on page 7, you can go ahead and open the copied readme and follow its instructions (adapting the path to the CDD and EAR files as needed). Otherwise, just use the provided example files as follows:

Navigate to *BE_HOME*/examples/standard/FraudDetectionCache and open `readme.html` in your browser. Follow the instructions in the readme file to start two engines and send events to the inference agent.

Optional Activity

Start a second cache agent. Experiment with stopping one cache agent engine then restarting it again, to see how one cache agent can operate successfully. Because one backup copy of the data is kept, and there are two cache agents, each cache agent has the complete cache contents. In a production system with more cache agents no single cache agent has all the cache data.

Congratulations—You have completed the caching tutorial!

You can now continue to

Chapter 4 **TIBCO BusinessEvents Monitoring and Management Tutorial**

This tutorial provides all the steps required to configure TIBCO BusinessEvents Monitoring and Management (MM) to work with the Fraud Detection project, and then shows you how to start the server and MM Console, and deploy the cluster engines.

This tutorial uses the configured FDCache project:

BE_HOME/examples/standard/FraudDetectionCache

Topics

Configure Copssh, TIBCO Hawk, and TRA File

Before you begin to work with TIBCO BusinessEvents Monitoring and Management (MM) you first install a utility and configure JMX properties. For machine level metrics you must install and configure TIBCO Hawk. For the purposes of the tutorial, the TIBCO Hawk requirement is optional.

Task A Install CopSSH Software for Remote Start and Deployment

You will use the Copssh user you activate here to authenticate to the host, for example, when deploying an engine.

1. To use SSH on Windows machines, you download and install Copssh software. It is readily available for download.

2. After you install the software, *S*elect **Start > All Programs > Copssh > 01 Activate a User**.

3. In the User Activation Wizard, provide the name of a user who can log on to the machine you are using for this tutorial. You can use your own credentials. Provide the user's password as the "passphrase."

> The Copssh user must be an actual user who can log onto the machine. You need to provide these credentials in order to deploy the engines.

4. Click **Activate**. After a few seconds, you see a success message. Click **OK**.

5. Check that the server is running. Open the Windows Services dialog (Start > Control Panel > Administrative Tools > Services) You should see `Openssh SSHD` as the service name, and it should be started.

Task B Configure JMX Properties in Monitored Engine TRA Files

Running engines use JMX MBeans to expose monitoring and management information to the MM server, and for remote method invocation.

Open the *BE_HOME*`/bin/be-engine.tra` and uncomment the `jmxremote` properties shown next. Ensure that the value of the port property is set to `%jmx_port%`. You will specify the actual port value in the JMX Port field in the site topology file Processing Unit Configuration settings.

In a production system, you would do this for all TRA files for all monitored BusinessEvents engines. When more than one PU (engine) is deployed on the same host, use a different JMX port for each of the PUs.

```
#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
```

Task C (Optional) Install and Configure TIBCO Hawk

For displaying machine-level metrics in MM Console, TIBCO Hawk is required. If you skip this step, you just won't see machine level metrics for your deployed cluster.

Full instructions for this task are provided in the section Install and Configure TIBCO Hawk for Machine Level Metrics in *TIBCO BusinessEvents Administration*. Below are some basic points for a default setup, for the benefit of experienced Hawk users.

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

2. 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*

3. 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.

Summary and Next Steps   You have done the prerequisite setup for using MM. Next you will complete the main task in getting ready to deploy the project with MM: configuring the site topology for the deployed cluster.

Configure the Site Topology File

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, IP address, and login credentials.

Instructions in this section are for the tutorial scenario: all components running on one Windows machine.

Learning Points   **What is a processing unit configuration?** You define processing units in the CDD file. One PU deploys as one engine. In the site topology file you select processing units and add deploy-time configuration settings to them. They are then known as *processing unit configurations (PUCs)*.

What is a deployment unit A deployment unit deploys as one or more engines. It consists of one or more processing unit configurations. It also specifies the location of the deployed EAR file and CDD file, generated by MM at deploy time. A DU can be re-used for different machines, unless it has machine-specific settings.

What is a predefined engine? A predefined engine is one that is configured in the site topology file. You can deploy predefined engines and monitor and manage them using MM. Unpredefined engines can be monitored if they are running but can't be started or deployed if they are not running.

Can I edit the site topology file in a text editor? Editing in TIBCO BusinessEvents Studio is always preferable because it can validate the file, and the graphical editor maintains the correct structure. However, when you can't use the graphical editor, you can use the template file to help you work with the settings in a text editor:

BE_HOME/mm/config/site_topology_template.st

More Information   • See these sections in *TIBCO BusinessEvents Administration*: Site Topology Overview, Configure the Site Topology in TIBCO BusinessEvents Studio, and Site Topology Reference.

### Task D Configure the Site Topology in TIBCO BusinessEvents Studio

1. Begin with the project you configured in Chapter 3, Cache OM Tutorial, on page 49. Or, as needed, import the FraudDetection Cache project into your workspace. See Importing Existing Projects into Your Workspace on page 7 for guidelines on working with example projects.

2. Right-click the project name, `FraudDetectionCache`, and select **New > Other > TIBCO BusinessEvents > Site Topology**.

   The Fraud Detection Cache example project ships with a configured site topology file, so give yours a different name.

   Do not change the `FraudDetectionCache.st` file in the example location. It is referenced by the MM CDD file as shipped.

3. At the New Site Topology Wizard, do the following:

   a. Enter the name `fdcacheSt`. As desired, enter a description.

   b. In the Cluster Deployment Descriptor field, browse to and select `fdcache.cdd`. Only CDD files within the studio project you are configuring are available for selection.

   c. 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:
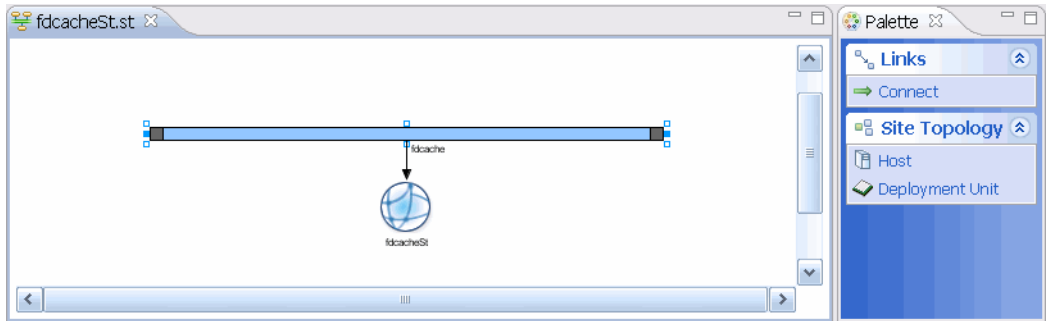
Cluster   4. Click the blue bar, which represents the cluster. In the Cluster Properties tab, do the following:

   — Copy the value of the Project CDD field to the Master CDD field.

   — In the Master EAR field, enter the location of the EAR file used for the Fraud Detection Cache project. If you are using the project in its shipped location, you would enter:

   `C:\tibco\be\4.0\examples\standard\FraudDetectionCache\fdcache.ear`

   The project CDD is used at design time to provide information about the processing units.

   The MM server reads the master files at deploy time and copies them to the remote deployment locations specified in the deployment units.

   If the BusinessEvents Version field appears in red, edit it to show the current version.

Deployment Units

5. In the Site Topology section of the palette to the right of the canvas, click the deployment unit (DU) icon and then click the canvas. A DU icon appears on the canvas. 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 the deployment unit to the cluster. For the tutorial you need only one DU. For a production system, you would need more, of course.

6. Click the DU icon on the canvas and configure the Deployment Unit Properties tab settings.

— The Deployment Unit Name field defaults to DU_1. Change it to **DU_WindowsOS**.

> It can be useful to put the operating system in the DU name because paths must be specified as appropriate for the deployed machine's OS. Other settings can be specific to a machine. It's good to use the name as a reminder about where you can deploy the unit to.

— 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. For the example deployment, this tutorial specifies `c:\temp\fdcache.cdd` and `c:\temp\fdcache.ear`. You can select different directories.

> A known issue in this release means you can't save the DU until you make a change in another field than Deployed CDD and Deployed EAR. You must make a change in the Deployment Unit Name field to enable the Save button. For example, you can add and then remove a letter.

Processing Unit
Configurations

7. In the DU property sheet, Click **Add** two times. Two processing unit configuration (PUC) entries appear:
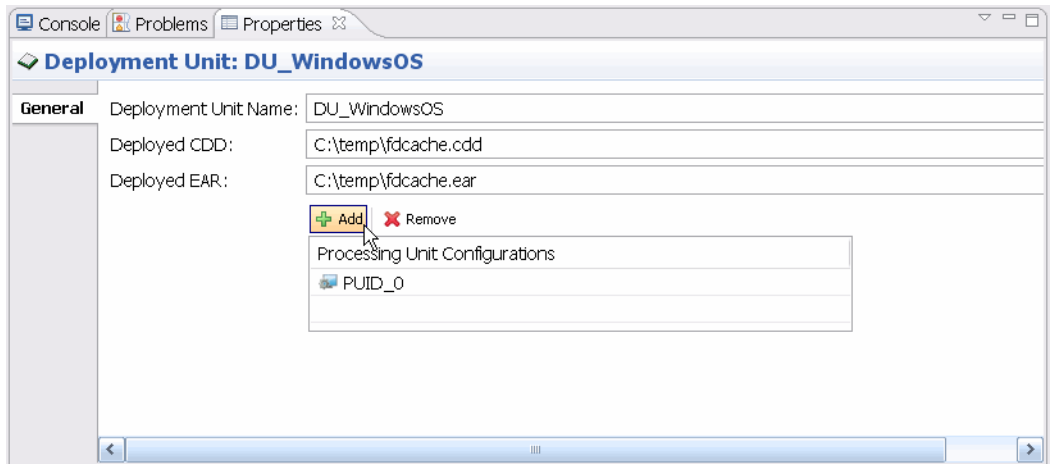
8. Double click the first PUC. (You can also click the PUC icon shown within the DU icon in the diagram). The Processing Unit Configuration properties appear. Configure the PUC as follows:

   — Replace the default name with the more descriptive name **InferenceAgent** because this engine will run one inference agent.

   — Select the option to use the PUC name as the engine name. This makes log files and the console easier to read.

   — Select the **default** processing unit. (The list displays the PUs defined in the CDD).

   — Set the JMX port to 5500.

9. Double click the second PUC and configure it as follows:

   — Replace the default name with **CacheServer**, because this engine will run one cache agent.

   — Select the option to use the PUC name as the engine name.

   — Select the **cache** processing unit.

   — Set the JMX port to 5501. When multiple PUs are running on one host, each PU must have a different JMX port.

Host

10. Add one host. In the Site Topology section of the palette, click the **Host** icon, and then click the canvas. A host icon appears on the canvas. Right-click in the canvas area to stop adding hosts.

11. Click the host icon and configure the host properties:

    — In the General tab, configure the host name as your actual machine name. Do not use `localhost`. In the Host IP field, enter the actual IP address of the machine.

    — As desired enter the Copssh username and password you activated in Task A. If you leave these fields blank you are prompted for the credentials at deploy time, for every action, such as deploy, start, and so on.

    — In the Installation tab, enter the BusinessEvents Home and TRA file locations. By default BusinessEvents Home is `C:\tibco\be\4.0` and the TRA file is here: `C:\tibco\be\4.0\bin\be-engine.tra`.

      If the BusinessEvents Version field appears in red, edit it to show the current version. (This happens only when an older ST file is copied to a later version's project.)

    — In the Start-PU-Method tab, select **Use SSH** (the default).

12. Connect the host to the deployment unit:

    — In the Links section of the palette, click the **Connect** icon.

    — Click the host and then the title bar of the deployment unit

    Right-click to stop connecting.

13. **Save**.

Summary and Next Steps   Now the site topology file is configured, you just have to specify its location in the MM.cdd file, to make the connection between the MM server and the cluster it will manage.

Specify the Site Topology File Location in the MM CDD File

You can configure the monitoring behavior of MM extensively using properties in its CDD file. In this task you will do just the minimum to connect MM with the cluster you will deploy and monitor.

More Information

See Chapter 7, MM Metrics and Features Configuration in *TIBCO BusinessEvents Administration*.

Task E   Specify the Site Topology File Location in MM.cdd

1. Import the *BE_HOME*/MM/project/emonitor project into your workspace and edit the MM.cdd.

2. In the mm-class agent properties list, expand the Topology group and enter the location of the site topology file you configured. The default location of the Fraud Detection Cache example's site topology file is:
   C:/tibco/be/4.0/examples/standard/FraudDetectionCache/FraudDetectionCache/FraudDetectionCache.st

   Enter a different value for the location as needed.

3. If you copy files into the workspace, remember to copy the updated MM.cdd file to the required location, the *BE_HOME*/mm/project/emonitor directory.

Summary and Next Steps

Now you're ready to start the server and deploy and start the engines.

Start MM Server and Log on to the MM Console

Task F   Start MM Server and Log on to the Console

1. To start do one of the following:

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

   — Open a console window, navigate to *BE_HOME*/mm/bin and execute this
     command:

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

2. 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. To change them, import
   the emonitor project and edit the MM.cdd in TIBCO BusinessEvents Studio. In
   the mm-class agent class properties, expand the
   global_variable_overrides > http property groups and edit the
   tibco.clientVar.HTTPHost and tibco.clientVar.HTTPPort variables.

3. Login using the credentials admin/admin.
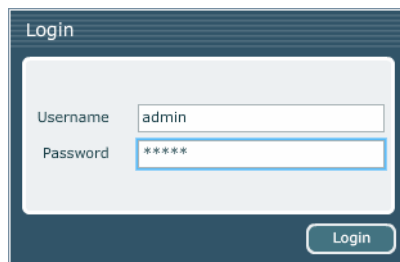
You see Cluster Explorer in the left panel. Expand the nodes under site. Hold the
cursor over the host icon. You see the text "Machine not deployed."
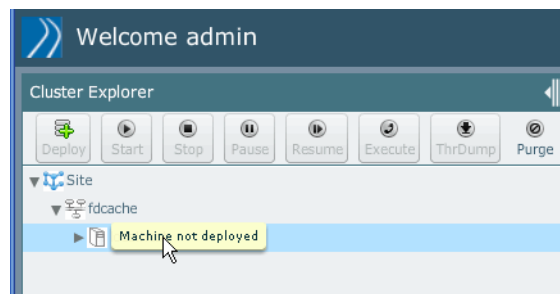
If you click on the host node, you see some charts in the overview section, but they show no information.

Task G   Deploy the Engines

1. In the Cluster Explorer, select the host node.

2. Click the **Deploy** button.

3. Provide the credentials you configured for Copssh (your login information).

4. Click **OK**. The engines deploy. Now when you hold the cursor over the host node, you see that it has deployed. You can also see the date and time.
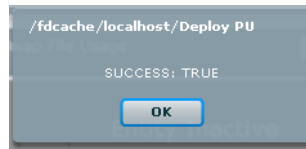
5. Now select the CacheServer processing unit and click **Start**.

6. Then select the InferenceAgent processing unit and click **Start**.

7. Now you can open the readme.html located by default in the *BE_HOME*/examples/standard/FraudDetectionCache folder. If you copied your project and saved your configured resources to the parallel folder set, open the readme there. Skip the instructions on starting the engines. Follow instructions to send data into the running engines, and observe the changes in the console window and in the MM Console page.

Congratulations! You have completed the TIBCO BusinessEvents Monitoring and Management tutorial!

Glossary

A

advisory event

A notice from BusinessEvents about activity in the engine, for example, an exception. See also event, simple event, simple event definition, time event.

agenda

A prioritized list of rule actions that may execute. Also known as the rule agenda. BusinessEvents recreates the agenda each time a change in the Rete network requires rules to be re-evaluated, for example, when a simple event is asserted. A rule action in an agenda may disappear without firing when the agenda is recreated, because conditions are no longer met.

agent

BusinessEvents operates at runtime using one or usually several agents of different types. See cache agent, inference agent, query agent.

agent class

An agent type, defined in the CDD file, that deploys as an agent instance.

agent group

Multiple deployed instances of one agent class. Used for load balancing and fault tolerance.

assert

Put facts into the Rete network. When object instances or events are asserted into the Rete network, rules may fire as a result. See also fact, retract.

B

backing store

A disk-based database to provide persistence and a better level of reliability for cache objects.

C

cache

In BusinessEvents, a type of object management. Generally refers to distributed storage of facts (objects) in memory. See also object management (OM), cache mode.

cache cluster

A group of processing units each running one or more agents configured for use by the cache-based object management system. Some agents are generally configured as cache agents. They store cache data for use by other cluster members. See object management (OM), cache.

cache mode

Various cache mode options are available at the object level, for use with cache-based object management. They enable you to tune the performance of your application, and reduce its footprint in memory. See also object management (OM), cache only, cache plus memory, memory only (local only).

cache agent

A processing unit configured with a non-reasoning agent used as a storage node only.

Used with Cache object management. See also agent, cache.

cache only

One of the cache modes, used at the object level. Instances of entity types that use cache only are serialized and kept in the cache until needed. They must be explicitly loaded into the Rete network when needed for rule processing. See also object management (OM), cache, cache mode

cache plus memory

One of the cache modes, used at the object level. Instances of entity types that use cache plus memory are serialized and kept in cache until needed. Used for objects that change infrequently. See also object management (OM), cache, cache mode.

cache server

See cache agent.

caching scheme

Caching schemes define how various cache-based object management options are used. BusinessEvents provides preconfigured caching schemes that are used automatically based on configuration choices. The caching schemes are stored in a cache configuration descriptor file. See also object management (OM), cache.

channel

A named configuration that allows BusinessEvents to listen to a stream of events from a given type of source, for example, JMS messages. A channel contains one or more destinations. See also destination.

checkpoint

As used in persistence object management, a checkpoint is the point in time at which working memory data is written to disk. The term checkpoint also encompasses all the activities involved in writing the data to disk.

cluster deployment descriptor

XML file containing properties to define the cluster, processing units, and agent classes. Configuration done using TRA file properties in earlier releases is now done using the CDD editor in TIBCO BusinessEvents Studio.

complex event

An abstraction that results from patterns detected among simple events.
Example: A complex event might result from the following simple events that all occurred within one week's time: A stock broker buys shares of xyz stock. The same broker submits a very large order for xyz stock on behalf of a customer. The same broker sells shares of xyz stock at a profit. See also simple event.

complex event processing (CEP)

Correlation of multiple events from an event cloud, with the aim of identifying meaningful events and taking appropriate action.

concept

An abstract entity similar to the object-oriented concept of a class. A concept is a description of a set of properties that, when grouped together, create a meaningful unit. Concepts can be organized in a hierarchical structure.
Example: *Department, employee, purchase order,* and *inventory item* are all concepts. The term *concept type* refers to the definition and the term *concept instance* refers to the actual object. Concepts are generally created using event data. See also event.

concept reference

A property within one concept that references the ID of another concept, known as the referenced concept. A type of relationship between concepts.

conflict resolution cycle

A cycle of activities during which the engine executes one set of rule actions on the currently asserted facts. One RTC may contain multiple conflict resolution cycles. See also Run to completion (RTC) cycle, agenda.

contained concept

A concept that exists entirely within another concept. A type of relationship between concepts.

custom function

You can add Java-based custom functions as needed to supplement the library of standard functions provided with BusinessEvents. See also ontology function, rule function, standard function.

D

decision table

A tabular form presenting a set of conditions and their corresponding actions. A graphical tool for building rules. Used in TIBCO BusinessEvents Decision Manager.

deserializer

A class that performs conversion tasks. In BusinessEvents, a deserializer converts messages to events. See also serializer

destination

A channel property that defines a contact point on a given channel. For example, for a TIBCO

Rendezvous channel, the destination properties would specify the subjects on which to listen.

distributed cache

In BusinessEvents, a form of cache-based object management. In a distributed cache, cached object data is partitioned between the processing units (JVMs) in the cache cluster for efficient use of memory. See also object management (OM), cache.

E

entity

A concept, simple event, or scorecard. Entity types are the definition of the entity. Similar in meaning to object. The term "instance" generally refers to a concept instance.

event

An object representing some occurrence or point in time. See also advisory event, simple event, simple event definition, time event.

evict

To remove an object or entry from a cache. An eviction policy defines when an object is removed from the cache. See also object management (OM).

expires (event)

At the end of the event's time to live period, the event is said to expire. It is removed from the Rete network and (as needed) acknowledged. Other actions depend on the type of object management used. See also time to live.

F

fact

An instance of an event or concept or scorecard in the Rete network.

I

in memory

In BusinessEvents, a form of object management. Refers to storage of facts (objects) used by the runtime engine in JVM memory. See also object management (OM).

memory only (local only)

One of the cache modes, available for cache-based object management configuration. Instances of entity types that use this mode are not stored in cache or backing store and are available only in the processing unit's local JVM memory. See also object management (OM), cache, cache mode.

inference agent

In a deployed system, inference agents process incoming events using a Rete network and a set of rules that are triggered by conditions in incoming events. Inference agents in Cache OM systems allow fault tolerance and load balancing. See also Rete algorithm.

instance

Similar to the Java term "object instance." By custom, applied only to concepts, though event definitions have object instances also.

L

lambda transition

A transition without a condition. This term is used in state model configuration.

M

O

object management (OM)

The aspect of BusinessEvents that deals with management of all the facts used in the runtime engine. Often shortened to OM in documentation. See also in memory, cache, persistence.

ontology function

BusinessEvents generates ontology functions for each entity type in a project. There are three types of ontology functions: *constructors*, to create a simple event or concept instance; *time events*, to create and schedule time events, and *rule functions*, to invoke rule functions. See also custom function, rule function, standard function.

P

payload

Similar to a JMS message, a simple event can contain properties and a payload. The payload holds the content of the message. You can define the XML schema for the payload when you configure the simple event definition. Payloads can also contain strings and Byte arrays.

persistence

In BusinessEvents, a form of object management. Refers to storage of facts (objects) used by the runtime engine, in a database. See also object management (OM).

processing unit

Definition of a BusinessEvents engine which runs in one JVM. Contains agents and other properties.

Q

query agent

A query agent is a non-reasoning agent that and has read-only access to the underlying objects in the cache cluster. A query agent has no Rete network. Available only with TIBCO BusinessEvents Event Stream Processing add-on software. See also agent.

R

Rete algorithm

Dr Charles L. Forgy developed the Rete algorithm for expert systems. See also Rete network.

Rete network

An in-memory network of objects based on the Rete algorithm which enables fast matching of facts with rule dependencies. "A Rete-based expert system builds a network of nodes, where each node (except the root) corresponds to a pattern occurring in the left-hand-side (the condition part) of a rule. The path from the root

node to a leaf node defines a complete rule left-hand-side. Each node has a memory of facts which satisfy that pattern." (http://en.wikipedia.org/wiki/Rete_algorithm)

retract

Remove facts from the Rete network. See also assert, fact, Rete network, working memory.

RMS

Rules Management Server. The server component of TIBCO BusinessEvents Decision Manager add-on software. RMS manages the rules management repository.

Run to completion (RTC) cycle

A run to completion (RTC), cycle generally begins when an external action causes changes to the Rete network. It ends when there are no more rule actions to execute as a result of that initial change (and any subsequent changes caused by rule actions). One RTC is composed of one or more *conflict resolution cycles*. Other terms for RTC are *forward chaining* and *inferencing*. See also conflict resolution cycle.

rule

A declaration, with a set of conditions and actions. If all the conditions in the rule are satisfied by facts in the Rete network (and the rule is at the top of the agenda), BusinessEvents executes the action. See also working memory, Rete network, agenda.

rule based time event

See time event.

rule function

Custom functions written using the BusinessEvents rule language and using a provided user interface. See also custom function, ontology function, standard function.

Also used to refer to a type of ontology function.

rule session

An older term that has been replaced by the term inference agent. See also inference agent.

S

serializer

A class that performs conversion tasks. In BusinessEvents, a serializer converts events to messages. See also deserializer

simple event

An object representing a business activity that happened at a single point in time. A simple event includes information for evaluation by rules, metadata that provides context, and a separate payload — a set of data relevant to the activity. See also advisory event, event, simple event definition, time event, time to live, expires (event).

simple event definition

A description of the channel, destination, properties, and payload for a simple event. See also simple event.

standard function

A library of standard functions is provided with BusinessEvents for use in rules and rule functions. See also custom function, ontology function, rule function.

T

time event

A type of event definition, used as a timer. Two types are available: rule based, and interval based. See also advisory event, event, simple event.

time to live

A simple event property that defines the delay after which a simple event expires. See also expires (event).

U

UML (Unified Modeling Language)

A language that assists in building a diagram of any complex entity. BusinessEvents diagrams use the UML. The BusinessEvents term, *concept*, is similar to a UML class.

V

virtual rule function

A rule function whose signature is defined in a BusinessEvents project and whose implementation is defined using decision tables. For use by TIBCO BusinessEvents Decision Manager add-on software. See also rule function.

W

working memory

The runtime processing area for rules, objects, and actions. Rules apply only to data in the working memory. Often used to mean Rete network. See also rule session, Rete network

Index

W