TIBCO BusinessEvents® Views

Ticker Tracker Example

Step-by-Step Guide

*Software Release 5.1.0*

***TBD****, 2011*

**Important Information**

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

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

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

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

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

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

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README.TXT FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Copyright © 2004-2011 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

[Introduction 1](#_Toc289937013)

[Problem Statement 1](#_Toc289937014)

[Quick Solution Overview 1](#_Toc289937015)

[Development 1](#_Toc289937016)

[Deployment 1](#_Toc289937017)

[Running 2](#_Toc289937018)

[Detailed Solution Steps 2](#_Toc289937019)

[Development 2](#_Toc289937020)

[Create metric population artifacts 2](#_Toc289937021)

[Create a metric 2](#_Toc289937022)

[Create an event 5](#_Toc289937023)

[Create a rule triggered by the event 7](#_Toc289937024)

[Create an HTTP Connection Shared Resource 9](#_Toc289937025)

[Create a HTTP Channel 11](#_Toc289937026)

[Create a Rule Function to act as a pre-processor for the HTTP channel 13](#_Toc289937027)

[Create dashboard rendering artifacts 15](#_Toc289937028)

[Create dashboard system elements 15](#_Toc289937029)

[Create a data source 17](#_Toc289937030)

[Create a table chart 19](#_Toc289937031)

[Create a column chart 23](#_Toc289937032)

[Create a dashboard page 27](#_Toc289937033)

[Create a view 30](#_Toc289937034)

[Create a role preference 32](#_Toc289937035)

[Create JDBC Connection Shared Resource 34](#_Toc289937036)

[Building the EAR 36](#_Toc289937037)

[Deployment 37](#_Toc289937038)

[Create deployment artifacts 37](#_Toc289937039)

[Create Cluster Deployment Descriptor 37](#_Toc289937040)

[Configure Database for JDBC Backing Store 42](#_Toc289937041)

[Configure Database to work with TickerTracker Project 43](#_Toc289937042)

[Running 44](#_Toc289937043)

[Running inference and dashboard agents 44](#_Toc289937044)

[Log in to the dashboard 45](#_Toc289937045)

[Sending events 47](#_Toc289937046)

Figures

[Figure 1 New Studio Project Wizard 2](#_Toc289936964)

[Figure 2 New Metric Wizard 3](#_Toc289936965)

[Figure 3 Metric Fields 4](#_Toc289936966)

[Figure 4 New Event Wizard 5](#_Toc289936967)

[Figure 5 Event Fields 6](#_Toc289936968)

[Figure 6 New Rule Wizard 7](#_Toc289936969)

[Figure 7 Rule Contents 8](#_Toc289936970)

[Figure 8 New HTTP Connection Shared Resource Wizard 9](#_Toc289936971)

[Figure 9 HTTP Connection Shared Resource Fields 10](#_Toc289936972)

[Figure 10 New Channel Wizard 11](#_Toc289936973)

[Figure 11 Channel Properties 12](#_Toc289936974)

[Figure 12 New Rule Function Wizard 13](#_Toc289936975)

[Figure 13 Pre-Processor Rule Function 14](#_Toc289936976)

[Figure 14 New Dashboard System Elements Wizard 15](#_Toc289936977)

[Figure 15 Dashboard System Elements 16](#_Toc289936978)

[Figure 16 New Data Source Wizard 17](#_Toc289936979)

[Figure 17 Data Source Properties 18](#_Toc289936980)

[Figure 18 New Chart Wizard 19](#_Toc289936981)

[Figure 19 Chart Type Wizard Page 20](#_Toc289936982)

[Figure 20 Chart Options Wizard Page 21](#_Toc289936983)

[Figure 21 Chart Data Option Wizard Page 22](#_Toc289936984)

[Figure 22 New Chart Wizard 23](#_Toc289936985)

[Figure 23 Chart Type Wizard Page 24](#_Toc289936986)

[Figure 24 Chart Options Wizard Page 25](#_Toc289936987)

[Figure 25 Chart Data Option Wizard Page 26](#_Toc289936988)

[Figure 26 New Dashboard Page Wizard 27](#_Toc289936989)

[Figure 27 Dashboard Page Template Selection Wizard Page 28](#_Toc289936990)

[Figure 28 Dashboard Page Components Selection Wizard Page 29](#_Toc289936991)

[Figure 29 New Config Wizard 30](#_Toc289936992)

[Figure 30 Config Properties 31](#_Toc289936993)

[Figure 31 New Role Preference Wizard 32](#_Toc289936994)

[Figure 32 Role Preference Properties 33](#_Toc289936995)

[Figure 33 New JDBC Connection Shared Resource Wizard 34](#_Toc289936996)

[Figure 34 JDBC Connection Shared Resource Properties 35](#_Toc289936997)

[Figure 35 Build Enterprise Archive Dialog 36](#_Toc289936998)

[Figure 36 New Cluster Deployment Descriptor Wizard 37](#_Toc289936999)

[Figure 37 Backing Store Properties 38](#_Toc289937000)

[Figure 38 Backing Store JDBC Connection Properties 38](#_Toc289937001)

[Figure 39 Destination Properties 39](#_Toc289937002)

[Figure 40 Dashboard Agent Properties 40](#_Toc289937003)

[Figure 41 Processing Unit Properties 41](#_Toc289937004)

[Figure 42 Dashboard Login 45](#_Toc289937005)

[Figure 43 Empty Dashboard 46](#_Toc289937006)

[Figure 44 Send Event VIA Http Form 47](#_Toc289937007)

[Figure 45 Updated Dashboard 48](#_Toc289937008)

[Figure 46 Changed Values for TIBX Event 49](#_Toc289937009)

[Figure 47 Updated Values for TIBX 50](#_Toc289937010)

[Figure 48 New Ticker Event 51](#_Toc289937011)

[Figure 49 New Values for CSCO 52](#_Toc289937012)

# Introduction

This document describes a step-by-step process to create a simple dashboard using BusinessEvents Views. We start with a problem statement and then work towards the solution including developing, deploying and running.

# Problem Statement

Create a simple dashboard, which shows various tickers with their latest price and volume.

# Quick Solution Overview

We are going to define a simple metric, which captures ticker price and volume. We will populate the metric using an event sent via an HTTP channel. We will define all the objects needed to render the information in the metric on a dashboard.

To successfully render a dashboard page, we need to create two agents, namely, an inference agent to populate the metric and a dashboard agent to serve the dashboard. Both the agents will be part of the same coherence cluster.

Below is a gist of the steps to be done to complete the solution.

## Development

1. Create metric population artifacts.
   1. Create a metric.
   2. Create an event.
   3. Create a rule triggered by the event.
      1. Add code to compute the metric.
   4. Create an HTTP Connection Shared Resource.
   5. Create an HTTP channel to send in events.
   6. Create a Rule Function to act as a pre-processor for the HTTP channel.
2. Create dashboard rendering artifacts.
   1. Create dashboard system elements.
   2. Create a data source.
   3. Create a chart.
   4. Create a dashboard page.
   5. Create a view.
   6. Create a role preference.
   7. Create a JDBC Connection Shared Resource.
3. Build the EAR.

## Deployment

1. Create deployment artifacts.
   1. Create a Cluster Deployment Descriptor.
   2. Configure the Database for JDBC Backing Store.
   3. Configure the Database to work with the TickerTracker Project.

## Running

1. Run the inference and dashboard agents.
2. Send events.

# Detailed Solution Steps

## Development

1. Create a new Studio Project.
2. From the menu, click **File** > **Project**.
3. Expand TIBCO BusinessEvents, and select Studio Project.
4. Type the Project Name as “TickerTracker”.

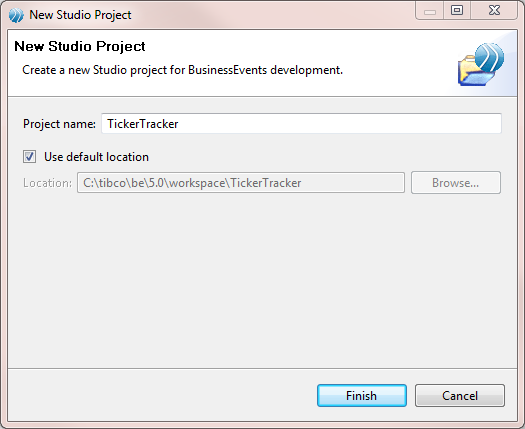


Figure New Studio Project Wizard

1. Click **Finish**.

### Create metric population artifacts

We will now create all the artifacts we need to successfully populate a metric with data.

#### Create a metric

Let us define a simple metric to capture the ticker information. The information captured includes the ticker symbol, price and volume. We will use the ticker as a group-by value to make sure that repeat events for the same ticker will update the price and volume instead of creating a new instance in the system. We will use set aggregation for price and volume since we want to maintain the latest values as they are received.

1. Click to select the Dashboard folder in Studio Explorer.
2. From the menu, click **File** > **Other**.
3. Expand TIBCO BusinessEvents, and then expand Views.
4. Select Metric, and click **Next**.
5. Type the Metric Name as “M\_TickerTracker”.

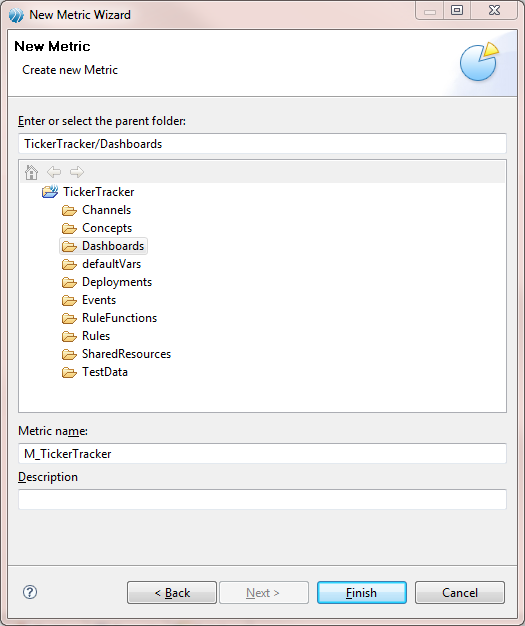


Figure New Metric Wizard

1. Click **Finish**.
2. In the metric, click to add a new field, and change the field name to “Ticker”.



1. Click to add another field, and change the field name to “Price” and the data type to “double”.



* 1. Click to add another field, and change the field name to “Volume” and the data type to “long”.

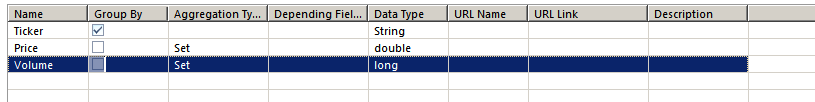


Figure Metric Fields

1. Save the metric.

#### Create an event

The metric we are defining needs to be populated. The metric can be populated via a rule, which can be triggered by any condition of your choice. For the sake of simplicity, we are going to populate our metric using an event sent on an HTTP channel.

The event also captures the ticker symbol and its price and volume.

1. Click to select the Events folder in Studio Explorer.
2. From the menu, select File > Other.
3. Expand TIBCO BusinessEvents, and select Simple Event.
4. Click **Next**.
5. Type the Event Name as “Ticker”.

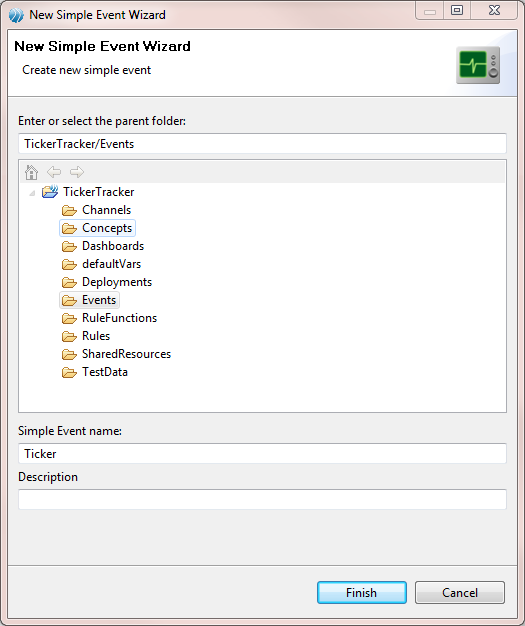


Figure New Event Wizard

1. Click **Finish**.
2. Click to add a new field, and change the field name to “Ticker”.



1. Click to add another field, and change the field name to “Price” and the data type to “double”.



1. Click to add another field, and change the field name to “Volume” and the data type to “long”.

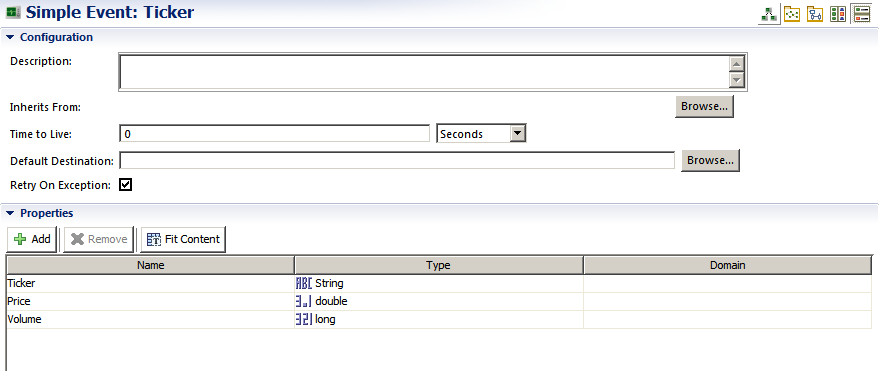


Figure Event Fields

1. Save the event.

#### Create a rule triggered by the event

We are going to write a rule, which will be triggered whenever a new “Ticker” event is asserted into the inference agent. Note that to populate the metric we use the Ontology function “compute(…)”. The compute function automatically checks if an instance of the metric already exists for the group-by values. If found, then the instance is updated, else a new instance is created.

1. Click to select the Rules folder in Studio Explorer.
2. From the menu, click **File** > **Other**.
3. Expand TIBCO BusinessEvents, and select Rule.
4. Click **Next**.
5. Type the Rule Name as “onTicker”.

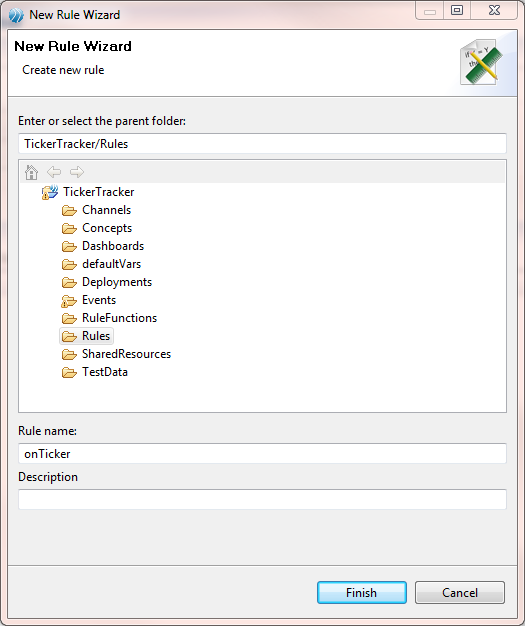


Figure New Rule Wizard

1. Click **Finish**.
2. In the declare section, type “Events.Ticker ticker;”.
3. in the **then** section, add the following text:

**System**.debugOut("Received Ticker[Ticker="+*ticker*.Ticker+",Price="+*ticker*.Price+",Volume="+*ticker*.Volume+")");

//compute the M\_Ticker metric

**Dashboards**.**M\_TickerTracker** instance = D**ashboards**.**M\_TickerTracker**.compute(*ticker*.Ticker,*ticker*.Price,*ticker*.Volume);

**System**.debugOut("Compute M\_Ticker[Ticker="+instance.Ticker+",Price="+instance.Price+",Volume="+instance.Volume+")");

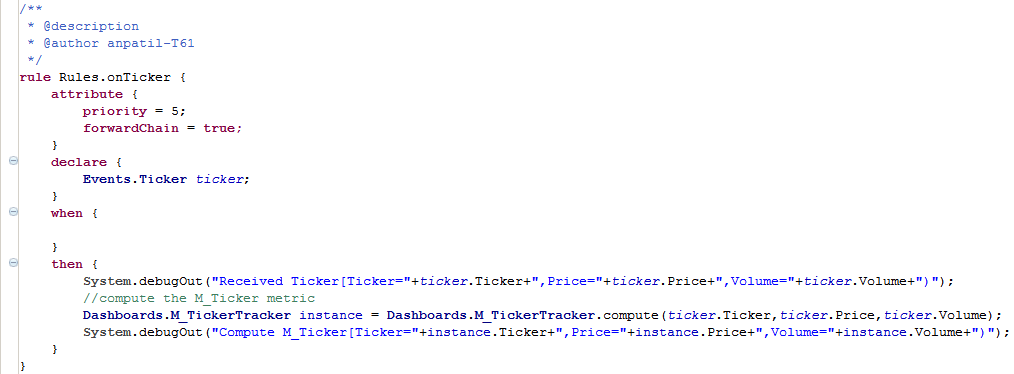


Figure Rule Contents

#### 

1. Save the rule.

#### Create an HTTP Connection Shared Resource

We are going to use an HTTP channel to receive the “Ticker” event. To configure the HTTP channel, we need to create an HTTP Connection Shared Resource.

1. Click to select the SharedResources folder in Studio Explorer.
2. From the menu, click **File** > **Other**.
3. Expand TIBCO Shared Resources, and select HTTP Connection.
4. Click **Next**.
5. Type the File Name as “HTTPConnection”.

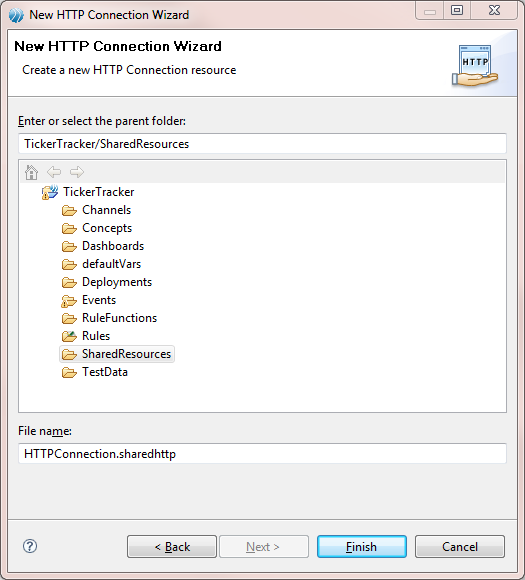


Figure New HTTP Connection Shared Resource Wizard

1. Click **Finish**.
2. Type “localhost” as the Host.
3. Type “8005” as the Port.

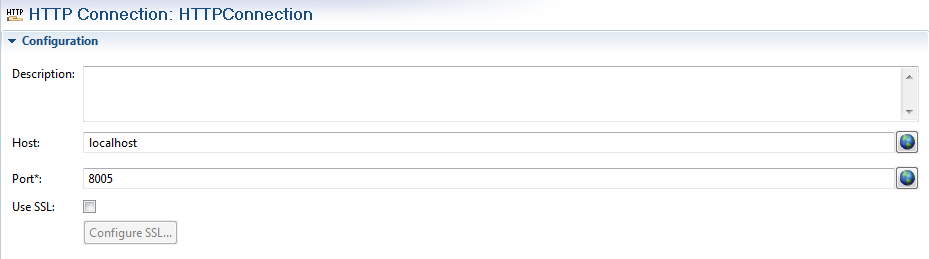


Figure HTTP Connection Shared Resource Fields

1. Save the Shared Resource.

#### Create a HTTP Channel

We will define the actual HTTP Channel and add a single destination to it to create the “Ticker” event.

1. Click to select the Channel folder in Studio Explorer.
2. From the menu, click **File** > **Other**.
3. Expand TIBCO BusinessEvents, and select Channel.
4. Click **Next**.
5. Type the Channel Name as “HTTP”.
6. Select the Driver Type as “HTTP”.

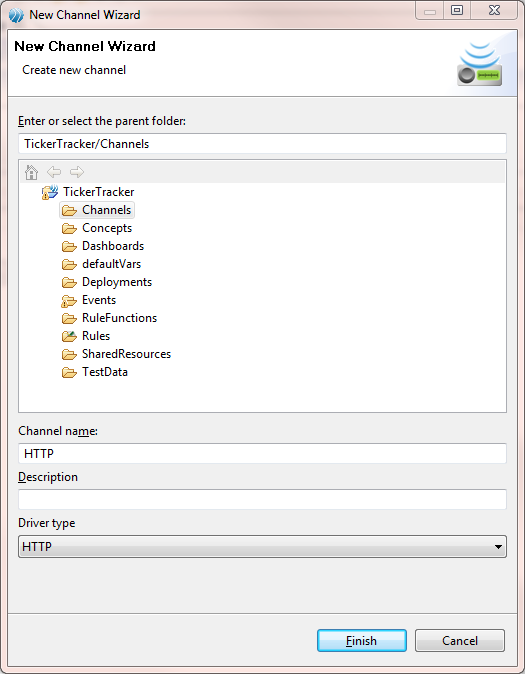


Figure New Channel Wizard

1. Click **Finish**.
2. In the Channel editor, select Resource “/SharedResources/HTTPConnection.sharedhttp”.
   1. Use Browse to launch the resource selection dialog.
   2. Select HTTPConnection.sharedhttp.
   3. Click **OK**.
3. Click to add a new destination.



1. Change the name to “AllOps”.
2. Select the Default Event as “/Events/Ticker”.
   * 1. Use Browse to launch the event selection dialog.
     2. Select the Ticker.
     3. Click **OK**.

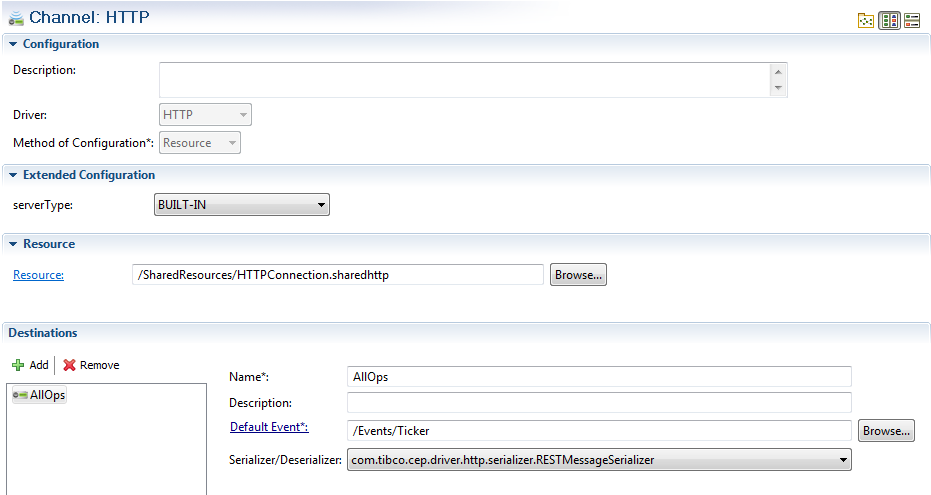


Figure Channel Properties

1. Save the channel.

#### Create a Rule Function to act as a pre-processor for the HTTP channel

An HTTP Channel is a synchronous channel. It is a good practice to respond to the incoming event with an outgoing event to complete the request-response cycle. Since our event-sending client is simple (an HTML form) and does not depend on acknowledgement that the event sent has been processed, we will write a pre-processor to respond to the incoming event with the same event.

1. Click to select the RuleFunctions folder in the Studio Explorer.
2. In the menu, click **File** > **Other**.
3. Expand TIBCO BusinessEvents, and select RuleFunction.
4. Click **Next**.
5. Type the Rule Function Name as “PreProcessor”.

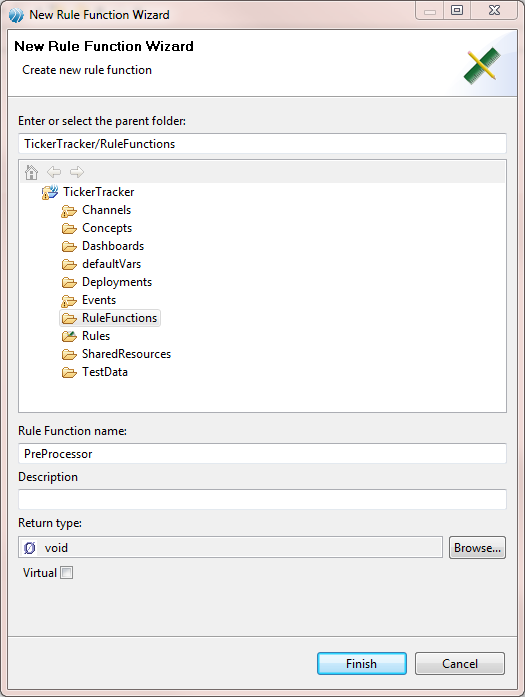


Figure New Rule Function Wizard

1. Click **Finish**.
2. Change the text of the Rule Function to the following:

/\*\*

\* @description Closes requests from the HTTP server

\*/

void rulefunction RuleFunctions.PreProcessor {

attribute {

validity = ACTION;

}

scope {

Event request;

}

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);

}

}

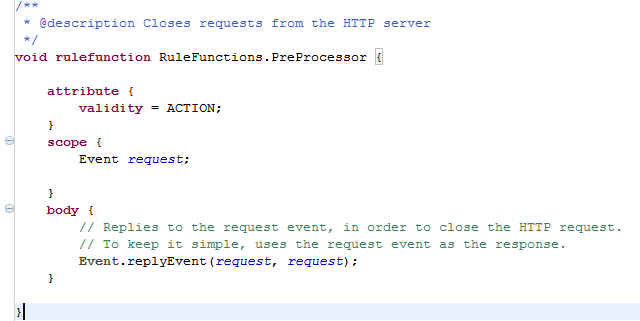


Figure Pre-Processor Rule Function

1. Save the rule function.

### Create dashboard rendering artifacts

We are now going to create all the artifacts we need to successfully render a dashboard using the dashboard agent.

#### Create dashboard system elements

TIBCO BusinessEvents Views needs certain bootstrap elements, such as a skin, to properly render a dashboard. For the sake of convenience, TIBCO BusinessEvents Views provides a way to create a standard skin using the “Dashboard System Elements” Wizard.

1. Click to select the Dashboard folder in Studio Explorer.
2. In the menu, click **File** > **Other**.
3. Expand TIBCO BusinessEvents, and then expand Views.
4. Select Dashboard System Elements, and click **Next**.
5. Type the SystemElement Name as “SE\_System”.

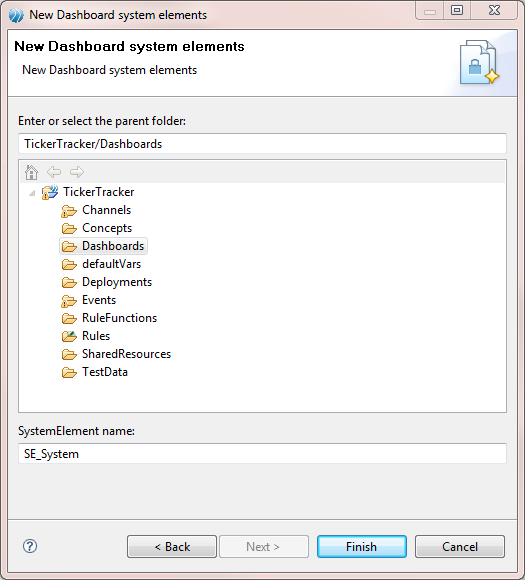


Figure New Dashboard System Elements Wizard

1. Click **Finish**.

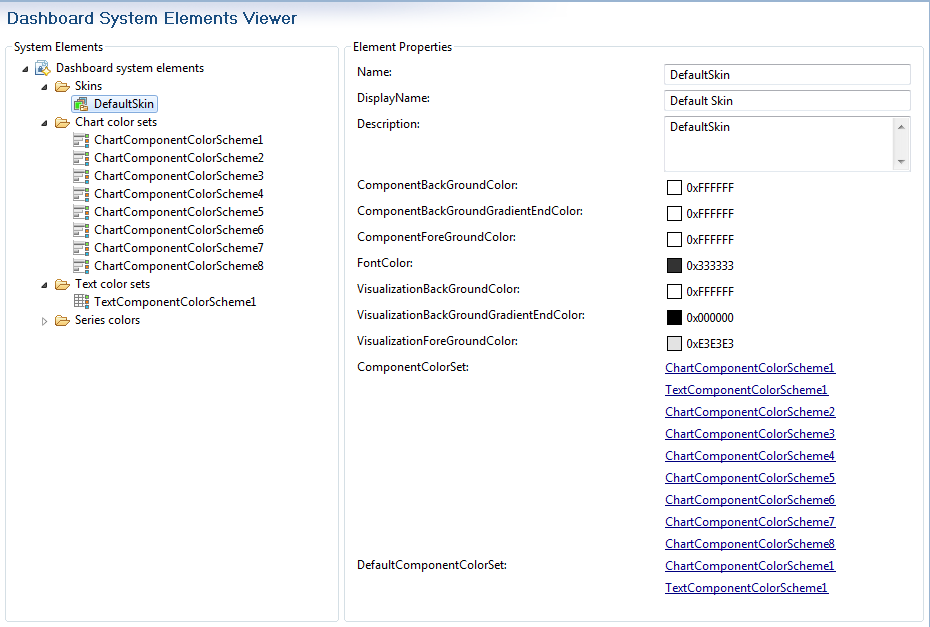


Figure Dashboard System Elements

#### Create a data source

Data source forms the query on top of a metric. We have the metric capturing the ticker, price and the volume. But we cannot filter that data or sort the data for rendering purposes in the metric. That is the job of the data source. Note that the data source is a shareable object, that is you can reuse the same data source across multiple charts.

We are now going to define a simple data source, which will pick all the data in the metric. We will sort the data by price to get a chart which shows the tickers ranked by their price with the highest at the top and the lowest at the bottom.

1. Click to select the Dashboard folder in Studio Explorer.
2. In the menu, click **File** > **Other**.
3. Expand TIBCO BusinessEvents, and then expand Views.
4. Select Data Source, and click **Next**.
5. Type the DataSource Name as “DS\_TickersSortedByPrice”.

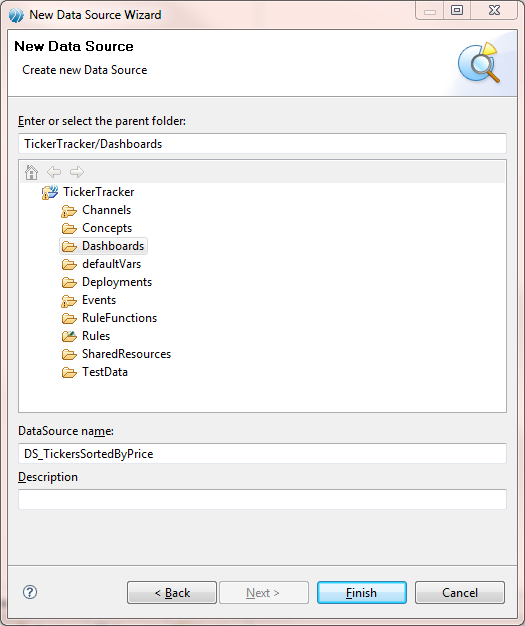


Figure New Data Source Wizard

1. Click **Finish**.
2. In the query section text area, add the following text:

select \* from /Dashboards/M\_TickerTracker order by Price desc;

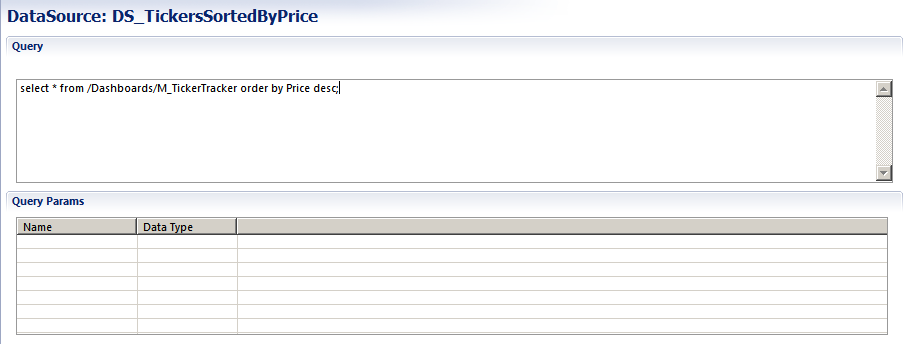


Figure Data Source Properties

1. Save the data source.

#### Create a table chart

We will now create a very simple table chart to show the data captured by the metric and filtered/sorted by the data source.

The chart wizard provides a quick way to configure your chart for easy use.

1. Click to select the Dashboard folder in Studio Explorer.
2. In the menu, click **File** > **Other**.
3. Expand TIBCO BusinessEvents, and then expand Views.
4. Select Chart, and click **Next**.
5. Type the ChartName as “C\_Tickers”.

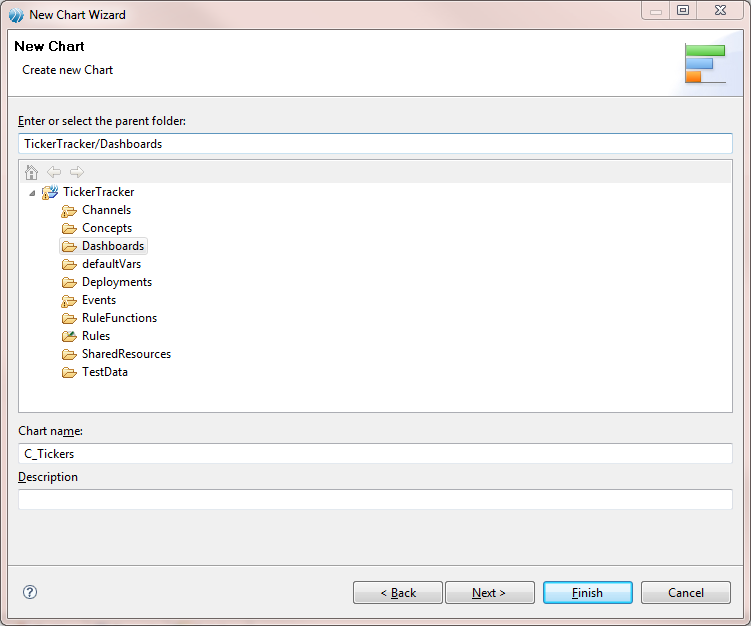


Figure New Chart Wizard

1. Click **Next**.
2. Select the Chart Type as “Table”.

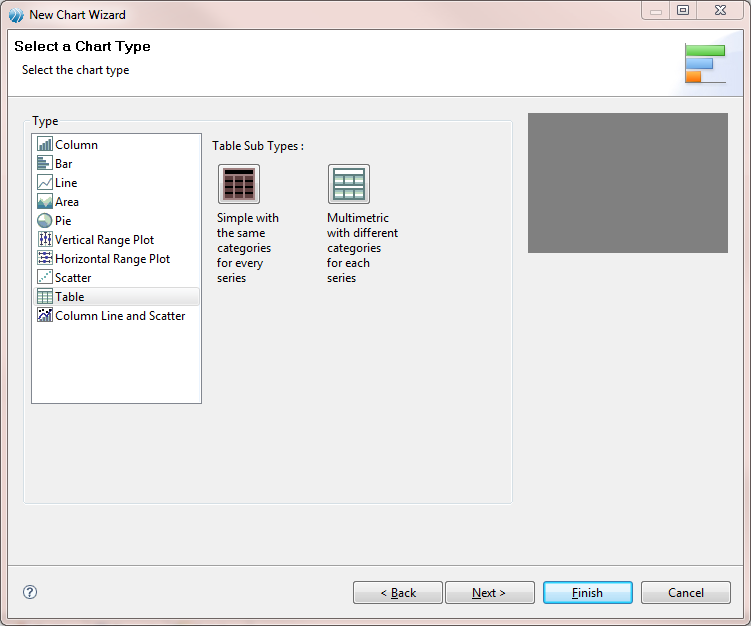


Figure Chart Type Wizard Page

1. Type the Chart Title as “Tickers” and type the Category Column Header as “Tickers”.
2. Change the Chart Size to 2 by 1 by clicking on the top row/middle column square.

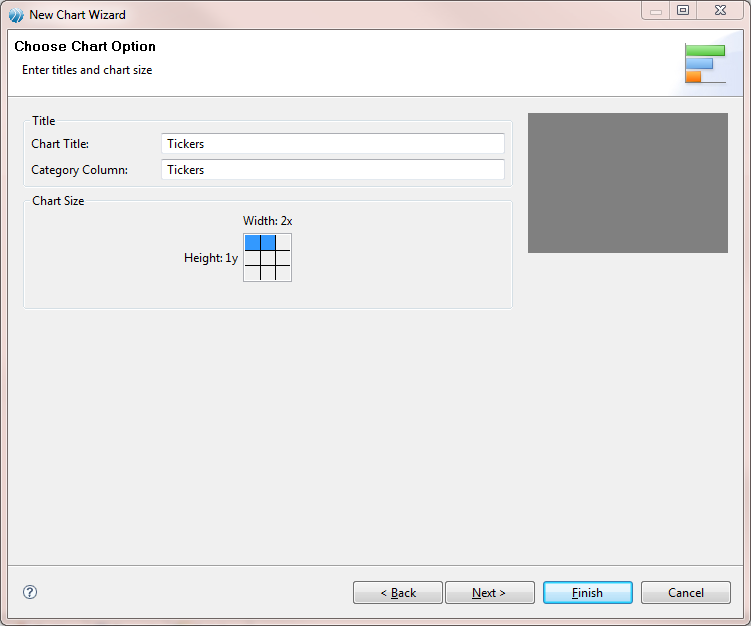


Figure Chart Options Wizard Page

1. Click Next.
2. Click Next.
3. Change the Series Name to “Price”.
4. Add a new series using the Add button.
5. Change the second series name to “Volume” and change the value field to Volume.

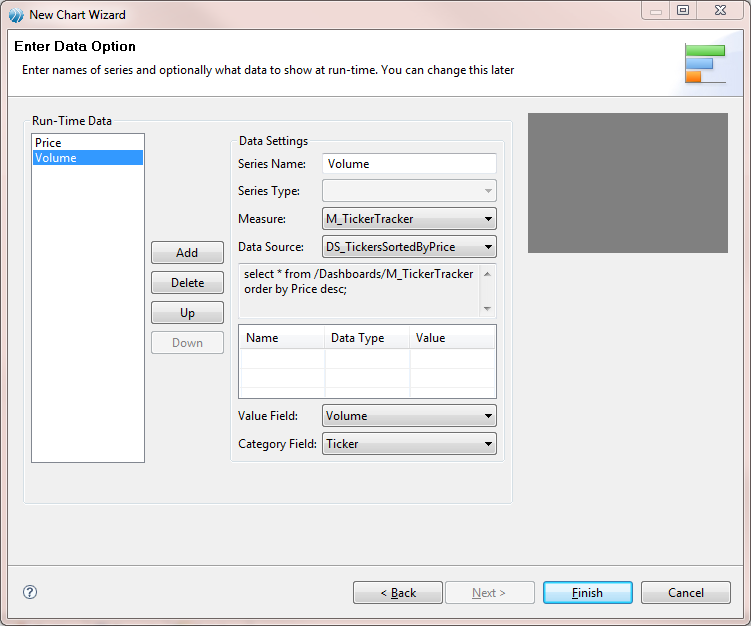


Figure Chart Data Option Wizard Page

1. Click **Finish**.

#### Create a column chart

We will now create a very simple column chart to show the data captured by the metric and filtered/sorted by the data source.

The chart wizard provides a quick way to configure your chart for easy use.

1. Click to select the Dashboard folder in Studio Explorer.
2. In the menu, click **File** > **Other**.
3. Expand TIBCO BusinessEvents, and then expand Views.
4. Select Chart, and click **Next**.
5. Type the ChartName as “C\_TickersByPrice”.

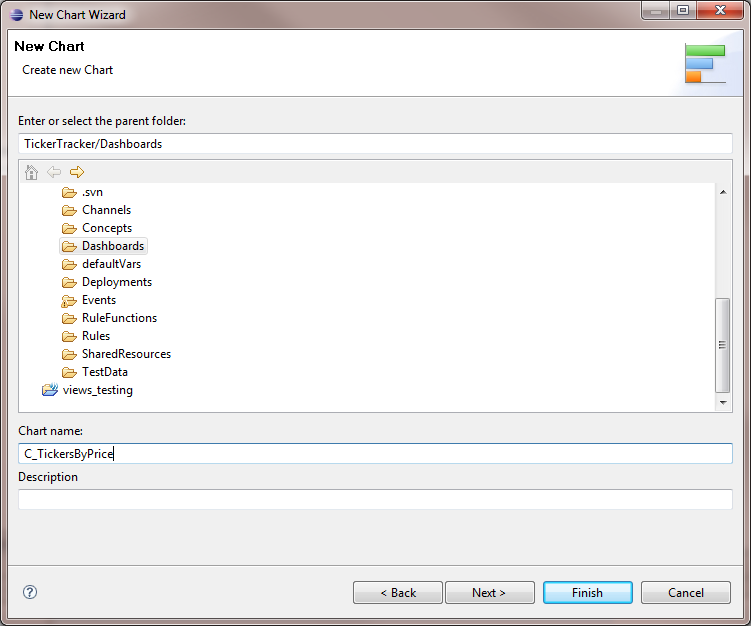


Figure New Chart Wizard

1. Click **Next**.
2. Select the Chart Type as “Column”.

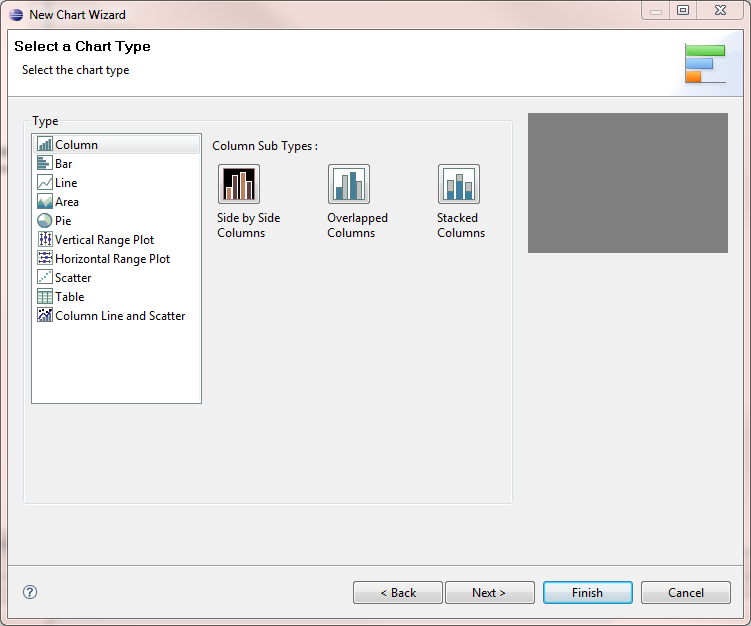


Figure Chart Type Wizard Page

1. Type the Chart Title as “Tickers By Price” and type the Category Axis as “Tickers” and the Value Axis as “Price”.

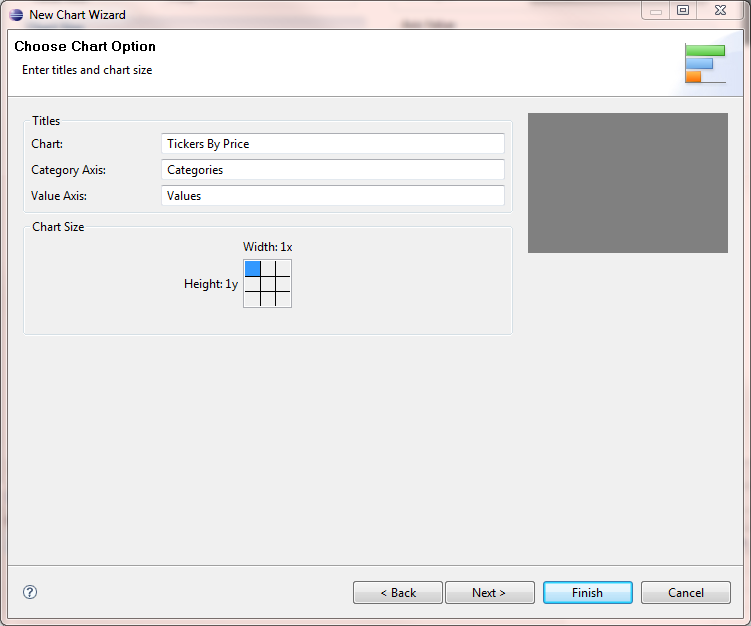


Figure Chart Options Wizard Page

1. Click Next.
2. Click Next.
3. Change the Series Name to “Price”.

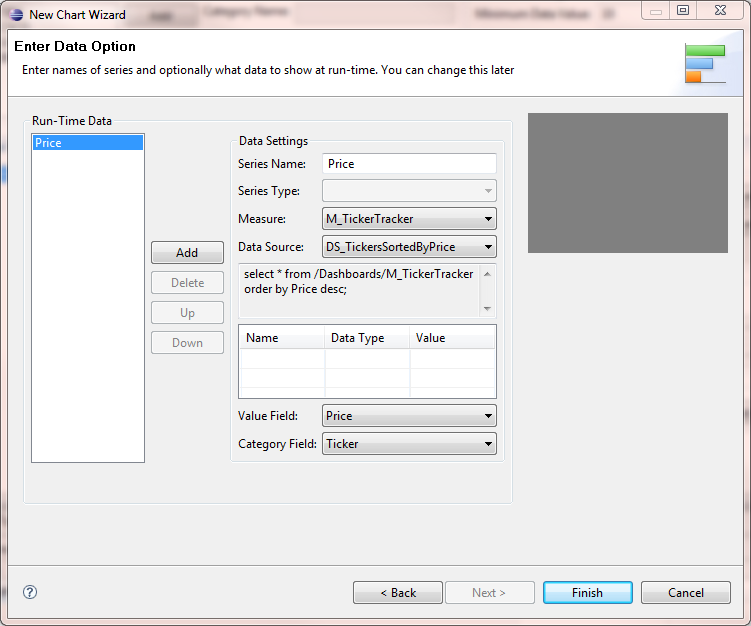


Figure Chart Data Option Wizard Page

1. Click **Finish**.

#### Create a dashboard page

We will now create a dashboard page to render the chart we created.

1. In studio explorer, click to select the Dashboard folder.
2. From the menu, click **File** > **Other**.
3. Expand TIBCO BusinessEvents, and then expand Views.
4. Select Dashboard Page, and then click **Next**.
5. Type the DashboardPage Name as “DP\_TickerTracker”.

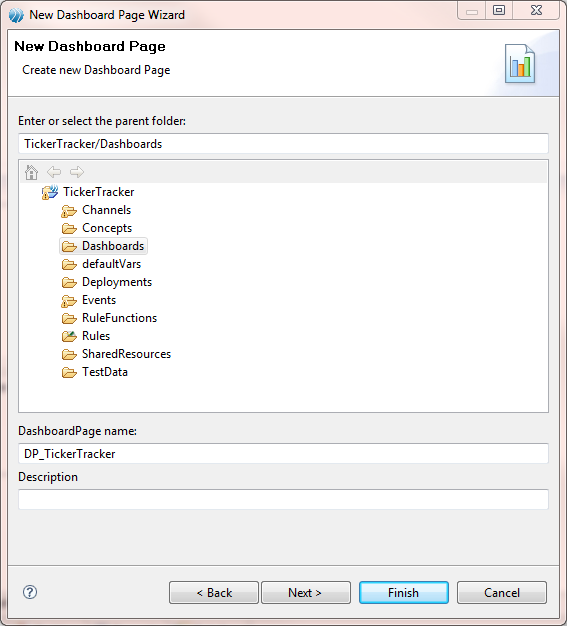


Figure New Dashboard Page Wizard

1. Click **Next**.

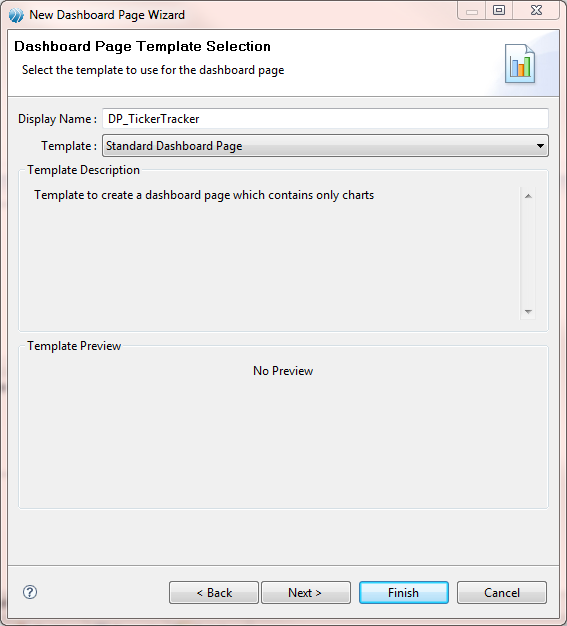


Figure Dashboard Page Template Selection Wizard Page

1. Click **Next**.
2. Select C\_Tickers and C\_TickerByPrice.

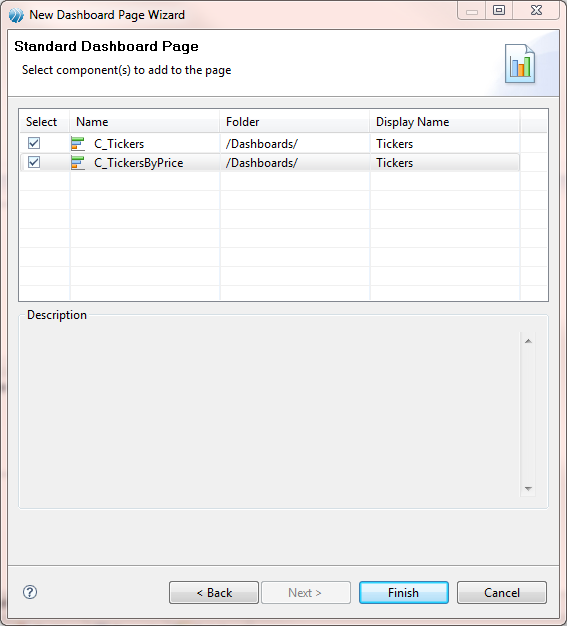


Figure Dashboard Page Components Selection Wizard Page

1. Click **Finish**.

#### Create a view

The view provides a way to collect various pages together to form a cohesive dashboard for the user. It also provides means to assign a skin to the dashboard.

1. In studio explorer, click to select the Dashboard folder.
2. From the menu, click **File** > **Other**.
3. Expand TIBCO BusinessEvents, and then expand Views.
4. Select View, and then click **Next**.
5. Type the View Name as “V\_TickerTracker”.

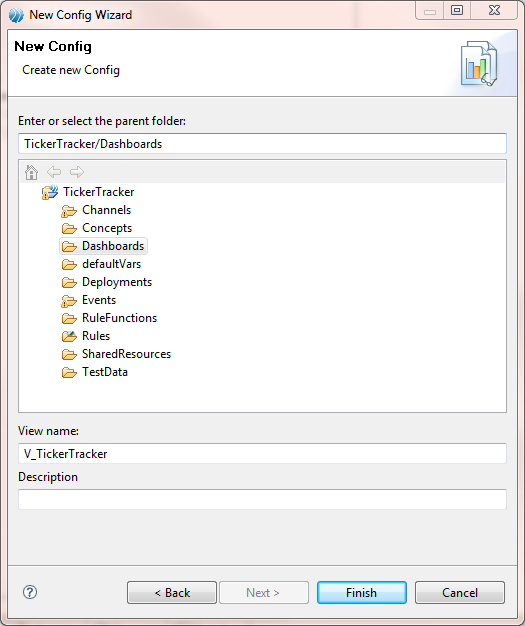


Figure New Config Wizard

1. Click **Finish**.
2. Select /Dashboards/DefaultSkin as the Skin.
3. Select DP\_TickerTracker Dashboard Page, and make it the default page by checking the check box in the default column.

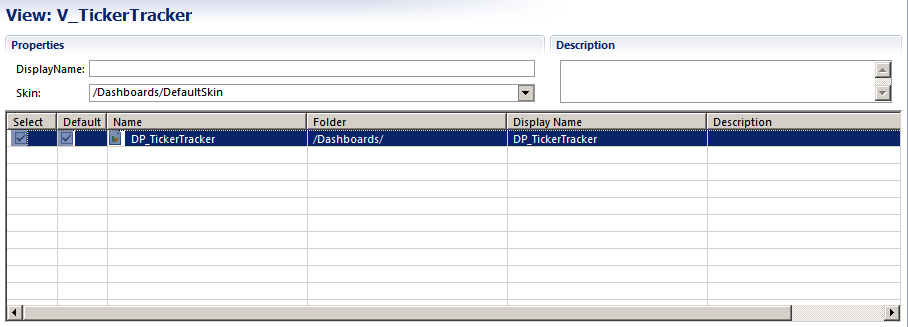


Figure Config Properties

1. Save the view.

#### Create a role preference

The role preference tells us which role will get to see which dashboards. It also lets us control the charts to which the user has access to via the component gallery.

1. In studio explorer, click to select the Dashboard folder.
2. From the menu, click **File** > **Other**.
3. Expand TIBCO BusinessEvents, and then expand Views.
4. Select Role Preference, and then click **Next**.
5. Type the RolePreference Name as “SUPER\_USER”.

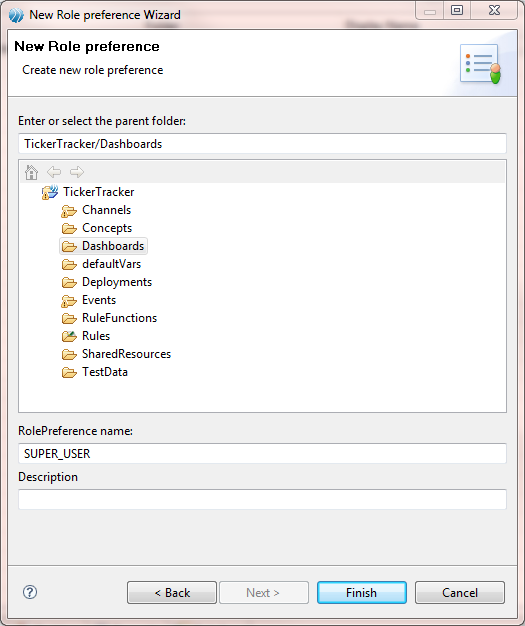


Figure New Role Preference Wizard

1. Click **Finish**.
2. Select /Dashboards/V\_TickerTracker as the view.
3. Create a single folder in the gallery:
   1. Right-click “No Galleries defined”, and select New.
   2. Enter the folder name as “root” and click **OK**.
   3. Select C\_Tickers under the “Select Components” section.

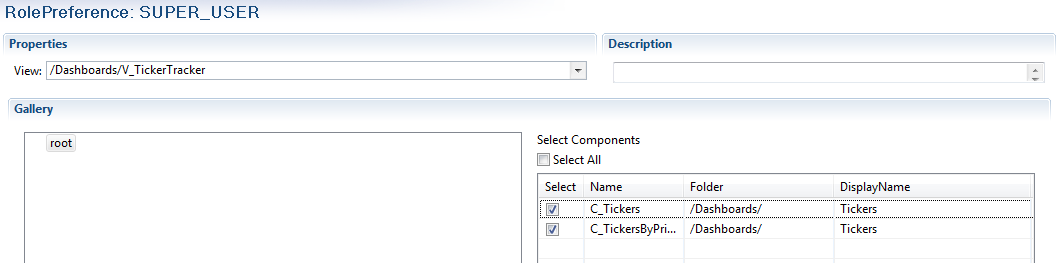


Figure Role Preference Properties

1. Save the role preference.

#### Create JDBC Connection Shared Resource

A JDBC backing store is a prerequisite for TIBCO BusinessEvents Views. We will now configure a JDBC Connection Shared Resource to provide all the database connectivity information to both, the inference and the dashboard agent, at runtime.

1. In studio explorer, click to select the SharedResources folder.
2. From the menu, click **File** > **Other**.
3. Expand TIBCO Shared Resources, and select JDBC Connection.
4. Click **Next**.
5. Type the File Name as “JDBCConfig”.

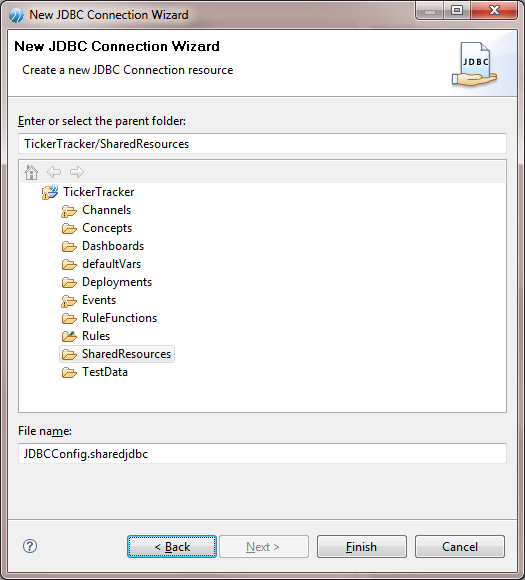


Figure New JDBC Connection Shared Resource Wizard

1. Click **Finish**.
2. Type the Database URL as jdbc:oracle:thin:@localhost:1521:XE, and change the URL to match your database instance connectivity URL.
3. Type the User Name as “be\_user” and change the User Name to match your database login name.
4. Type the Password as “be\_user” and change the Password to match your database password.

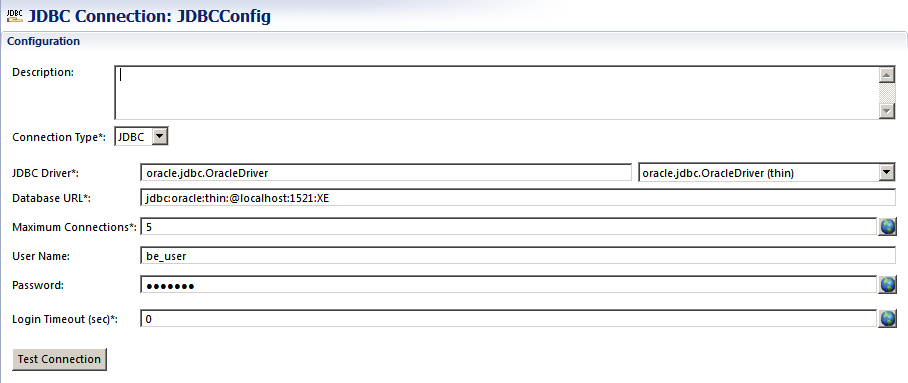


Figure JDBC Connection Shared Resource Properties

1. Save the shared resource.

### Building the EAR

Now we are ready to build the Enterprise Archive.

1. In studio explorer, click to select the TickerTracker Project.
2. From the menu, click **Project** > **Build Enterprise Archive…**
3. Select the File Location and click Apply.
4. Click **OK**.

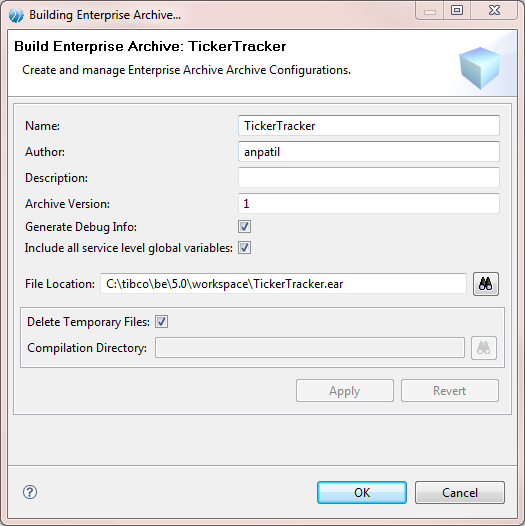


Figure Build Enterprise Archive Dialog

## Deployment

Now that we have successfully built the Enterprise Archive, we need to set up our runtime environment.

### Create deployment artifacts

For successful deployment, we need the runtime environment to know how the cluster is configured and we also need a properly initialized JDBC backing store.

#### Create Cluster Deployment Descriptor

We will now define a CDD. We will configure using a JDBC backing store and define two agents, namely, inference and dashboard. We will also configure the Coherence Cluster.

1. In studio explorer, click to select the TickerTracker Project.
2. From the menu, click **File** > **Other**.
3. Expand TIBCO BusinessEvents, and select Cluster Deployment Descriptor.
4. Click **Next**.
5. Type the File Name as “tickertracker”.

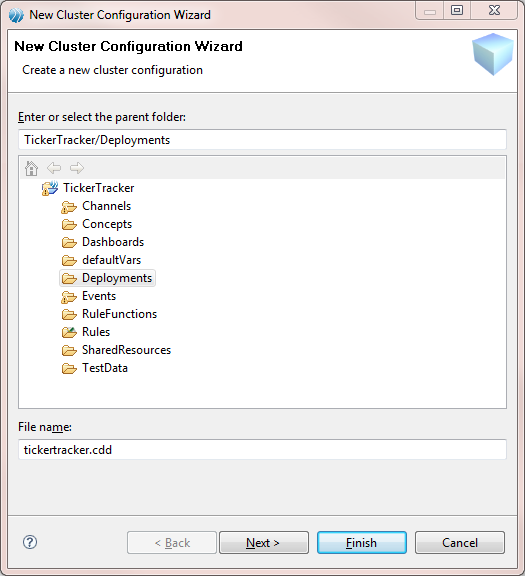


Figure New Cluster Deployment Descriptor Wizard

1. Click **Finish**.
2. Change the Object Management to “Cache” by right-clicking Object Management in the cluster tab and selecting “Change to Cache”.
3. Enable the Backing Store by clicking the Backing Store, and clicking the checkbox next to Enabled in the Configuration section.

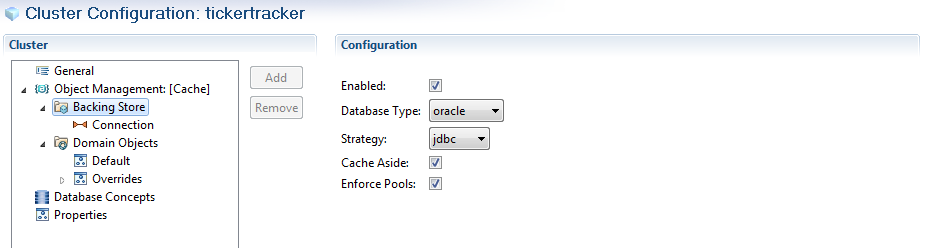


Figure Backing Store Properties

* 1. Click Connection, and change the URI to “/SharedResources/JDBCConfig.sharedjdbc”.
     1. Click **Browse** to open the shared resource selection dialog.
     2. Navigate to and select JDBCConfig.sharedjdbc.
     3. Click **OK**.

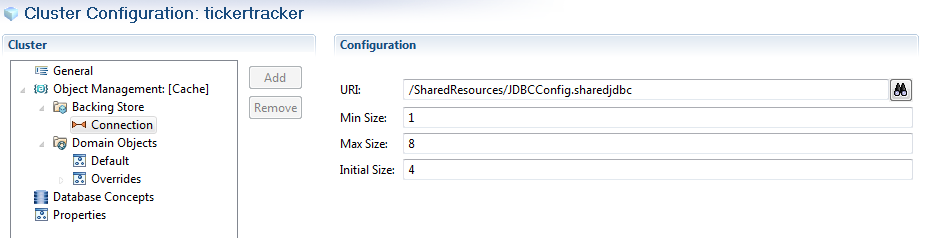


Figure Backing Store JDBC Connection Properties

1. Add a destination Group.
   1. Open the Collections Tab.
   2. Click Destinations and click Add.
   3. Name the Destination Collection as “AllOperations” and click Add.
   4. Select **Channel** > **HTTP** > **AllOps**, and click **OK**.
   5. Change the Destination ID to “ticker” and select the Threading Model as “Caller”.
   6. Change the Preprocessor to “/RuleFunctions/PreProcessor”:
      1. Click Browse to open the preprocessor selection dialog.
      2. Navigate to and select the “/RuleFunctions/PreProcessor”.
      3. Click **OK**.

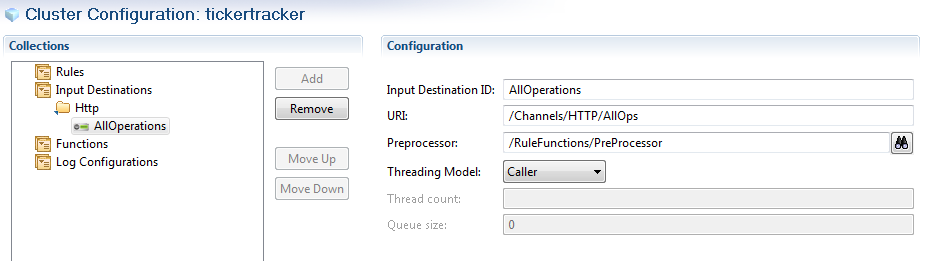


Figure Destination Properties

1. Add cluster level properties
   1. Add the following properties to the Properties

|  |  |
| --- | --- |
| **Property Name** | **Property Value** |
| be.mm.auth.type | file |
| be.mm.auth.file.location | C:/tibco/be/5.1/mm/config/users.pwd |

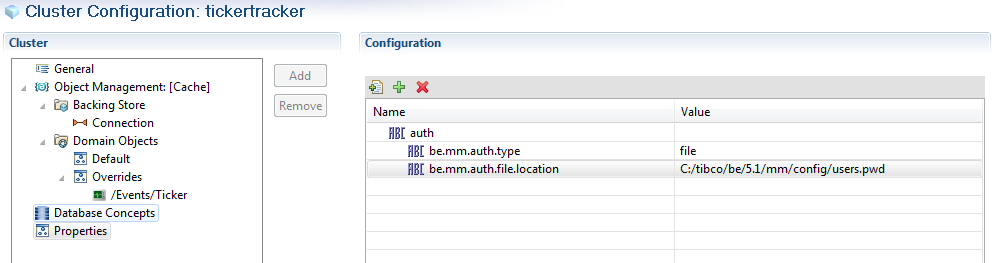


Figure Cluster Level Properties

1. Update the inference-class:
   1. Add the HTTP destination group.
   2. Click the Destination Collections:
      1. Click Add.
      2. Select Reference **Collections** > **HTTP**.
      3. Click **OK**.
2. Add a new dashboard-class:
   1. Click Add Agent.
   2. Type the Agent Class Name as “dashboard-class”.
   3. Select the Agent Class Type as “Dashboard”.
   4. Click **OK**.
   5. Click dashboard-class.
   6. Add the following properties to the dashboard-class:

|  |  |
| --- | --- |
| **Property Name** | **Property Value** |
| be.auth.type | file |
| be.auth.file.location | C:/tibco/be/5.1/examples/views/TickerTracker/config/users.pwd |
| be.agent.dashboard.mal.storage.file.root | C:/tibco/be/5.1/examples/views/TickerTracker/personalizeddata |
| be.agent.dashboard.hostname | localhost |
| be.agent.dashboard.pullrequestport[[1]](#footnote-1) | 8181 |

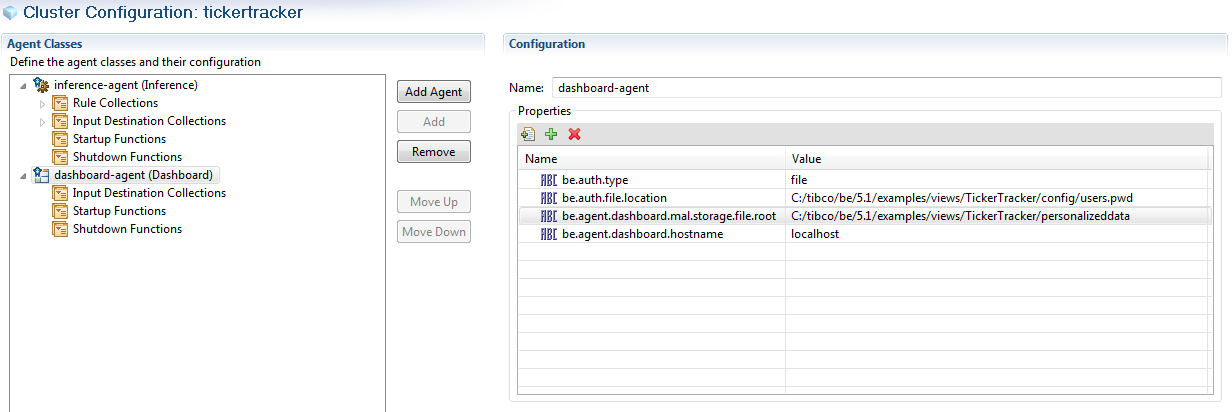


Figure Dashboard Agent Properties

1. Update the processing-unit:
   1. Check ‘Enabled Cache Storage’
   2. Add the dashboard-class agent:
      1. Click the button and select dashboard-class.



* + 1. Click **OK**.

#### Configure Database for JDBC Backing Store

We will now configure our database to work with the JDBC backing store.

1. Install Oracle XE.
2. Create the user name/password.

Open a command prompt and then run the following commands:

cd BE\_HOME/bin

sqlplus system/<your system login password> @initialize\_database\_oracle.sql

1. Initialize the database:

Open a command prompt and then run the following commands:

cd BE\_HOME/bin

sqlplus be\_user/be\_user @create\_tables\_oracle.sql

#### Configure Database to work with TickerTracker Project

We will now configure the JDBC backing store to work with our TickerTracker Project.

1. Open a command prompt and run the following commands:

cd <BEStudioWorkspace>

BE\_HOME/bin/be-jdbcdeploy --propFile BE\_HOME\bin\be-jdbcdeploy.tra -o TickerTracker TickerTracker.ear

sqlplus be\_user/be\_user @TickerTracker.sql

## Running

Now that we have a properly configured environment, we can start the system and see how our dashboard looks.

### Running inference and dashboard agents

1. Copy the Oracle JDBC Thin driver libraries (classes12.zip) to BE\_HOME/lib/ext/tpcl.
2. Open a command prompt and run the following commands to change the TickerTracker.ear location based on what you entered for File Location in the Build Enterprise Archive dialog:

cd <BE\_HOME>/views/bin

be-views –c <BEStudioWorkspace>/tickertrackerdb.cdd -u default <BEStudioWorkspace>/TickerTracker.ear

### Log in to the dashboard

1. Launch Internet Explorer.
2. Type [http://localhost:8181](http://localhost:8080) in the address bar.
   1. If you have changed “be.agent.dashboard.pullrequestport” in the <BEStudioWorkspace>/tickertracker.cdd, then change the URL to use your custom port number.

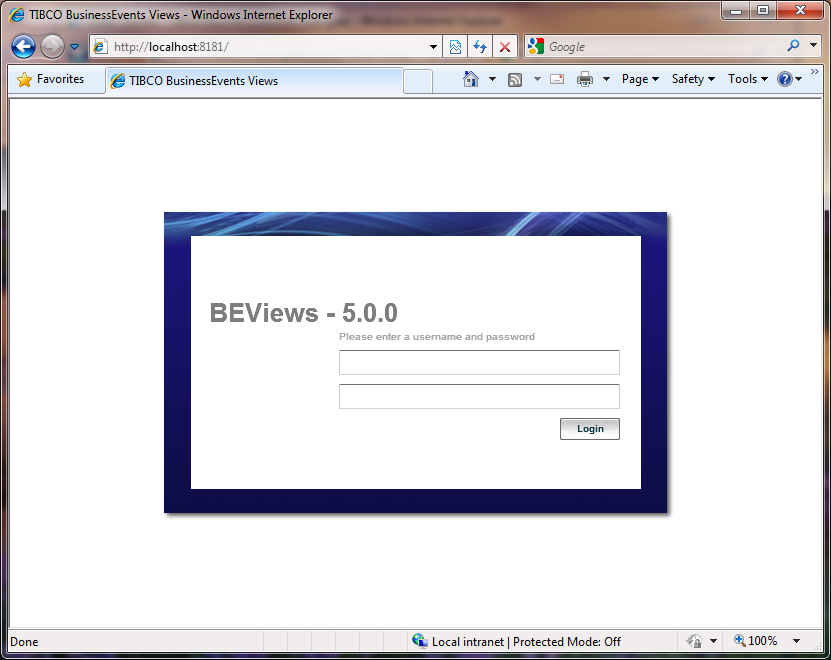


Figure Dashboard Login

1. Type the User Name as “admin” and Password as “admin”.

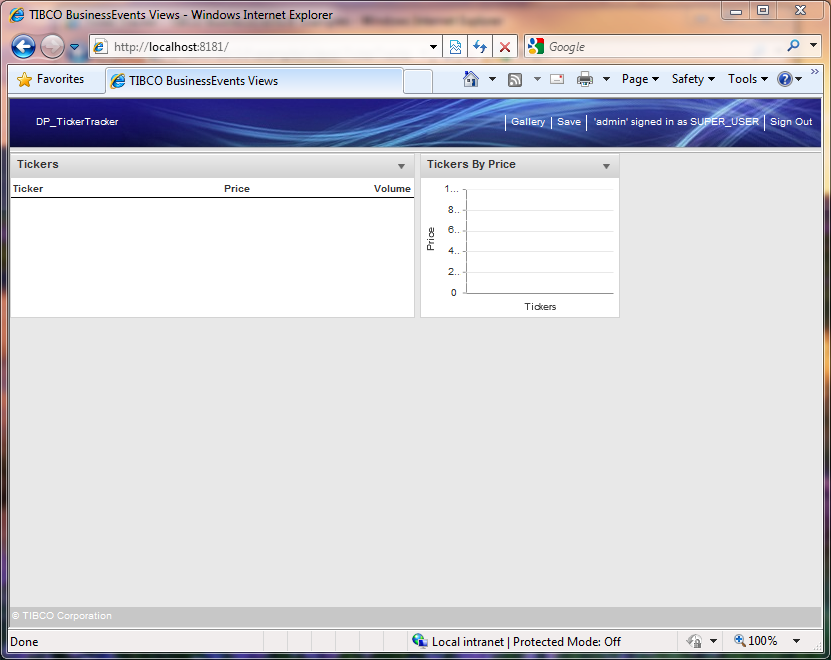


Figure Empty Dashboard

### Sending events

We have successfully started our system and have logged in to the dashboard. Now we will send events to the system over the HTTP Channel.

1. Launch Internet Explorer.
2. Open “readme.html” provided in the learning kit.

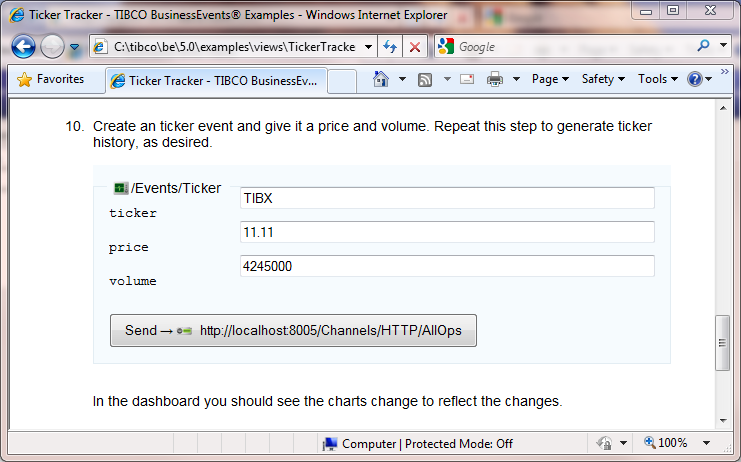


Figure Send Event VIA Http Form

1. Click the “Send -> <http://localhost:8005/Channels/HTTP/AllOps>” button.  
   The dashboard should be updated to show the TIBX/11.11/4245000

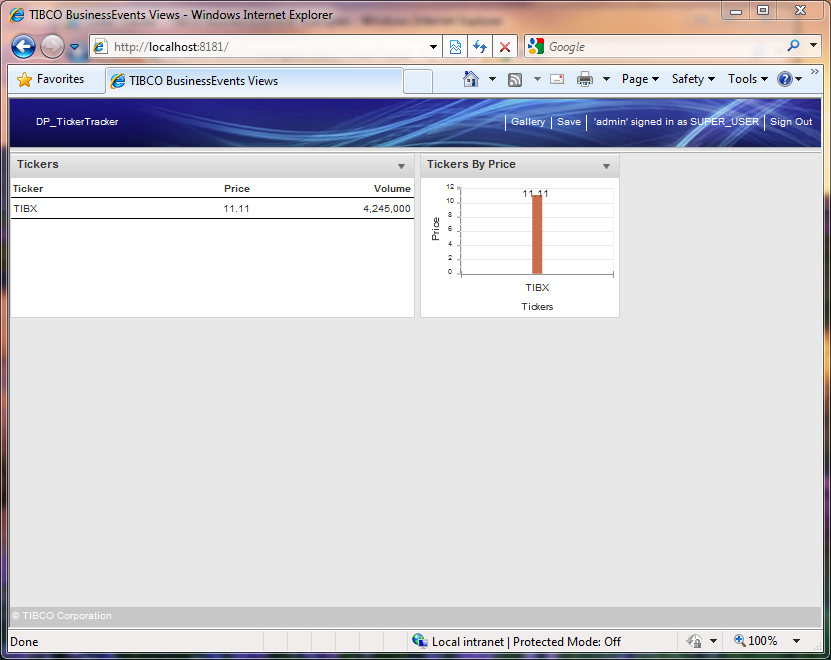


Figure Updated Dashboard

1. Now change the values for price and volume to 15.00 and 5005000 respectively in the “readme.html” and again click the “Send -> <http://localhost:8005/Channels/HTTP/AllOps>” button.

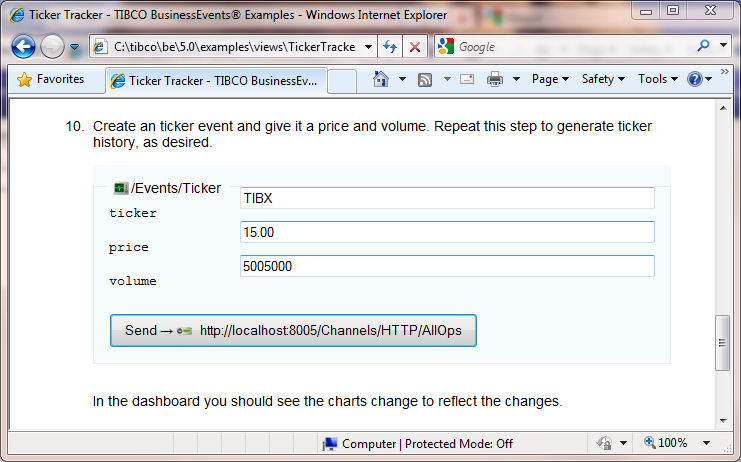


Figure Changed Values for TIBX Event

Notice how the values have changed in the dashboard

You will see the TIBX row getting updated. That is because we have marked the ticker field as group-by in the metric, which means any event for TIBX ticker will cause the same instance to get updated.

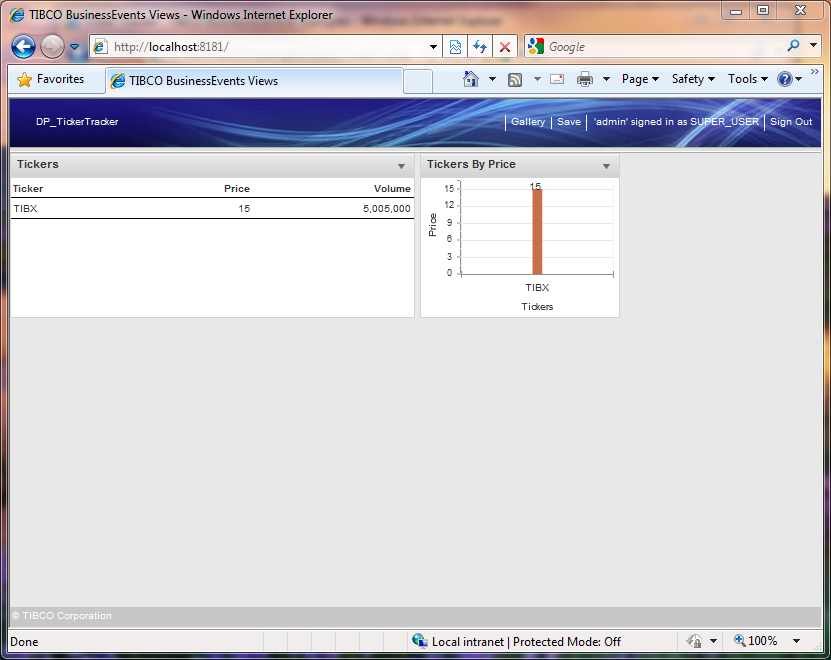


Figure Updated Values for TIBX

1. Now change the values for ticker, price and volume to CSCO, 24.00 and 63000000 respectively in the “readme.html” and again click the “Send -> <http://localhost:8005/Channels/HTTP/AllOps>” button.

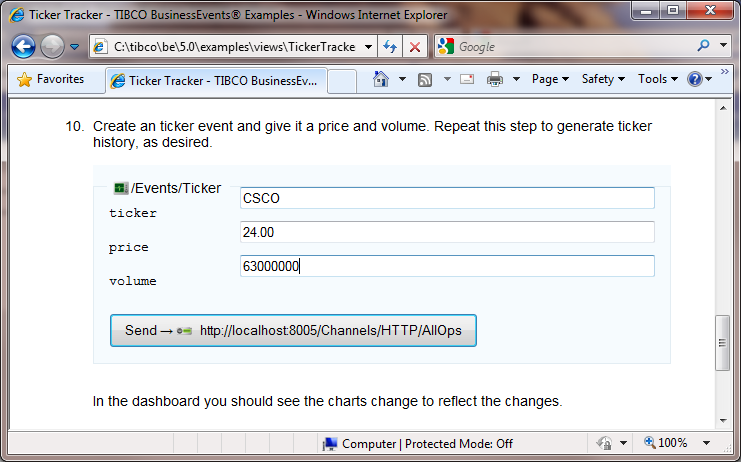


Figure New Ticker Event

Notice how you see a new row in the table for CSCO. You will also notice that CSCO is above TIBX since CSCO’s price (24.00) is higher than TIBX’s price (15.00)

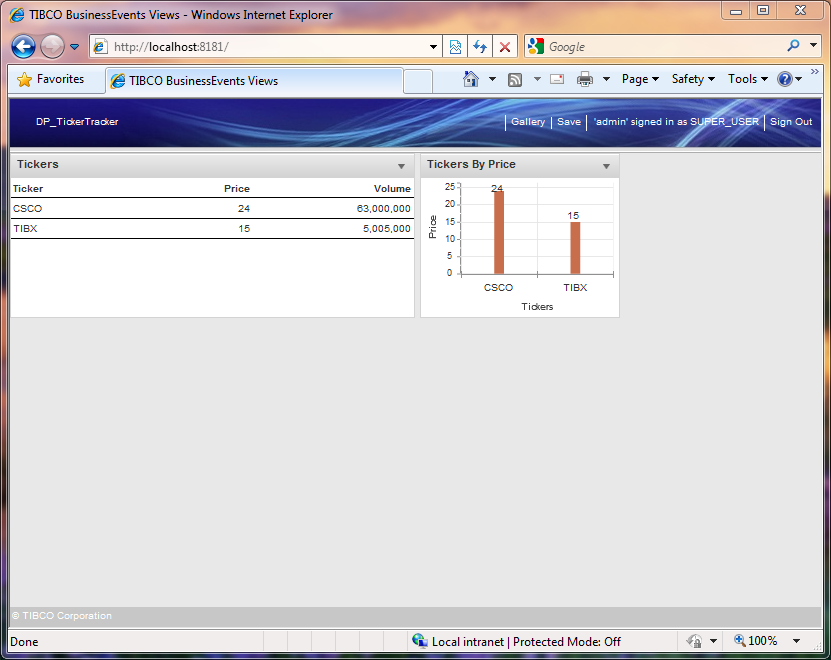


Figure New Values for CSCO

1. The default be.agent.dashboard.pullrequestport is 8080. If you have some other service running on 8080, then you should specify the be.agent.dashboard.pullrequestport with a different port number. [↑](#footnote-ref-1)