

TIBCO BusinessEvents® WebStudio User's Guide

*Software Release 5.3
May 2016*

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 BusinessEvents, TIBCO Designer, TIBCO Enterprise Message Service, TIBCO Enterprise Administrator, TIBCO Hawk, TIBCO Runtime Agent, TIBCO Rendezvous, and Two-Second Advantage are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

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

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

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

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

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

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

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

Copyright © 2004-2016 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Figures	6
TIBCO Documentation and Support Services	7
TIBCO BusinessEvents WebStudio Concepts	8
Rules Management Server (RMS)	8
TIBCO BusinessEvents WebStudio Overview	8
Virtual Rule Functions and Decision Tables	9
Rule Building with Decision Tables	9
Rule Template and Business Rule	10
Rule Building with Business Rules	10
WebStudio and RMS User Workflow	11
Display Models in WebStudio	11
Configuring RMS	15
Adding a Project to RMS	15
Configuring RMS Server Properties	15
Enabling Remote Connection to RMS from WebStudio	16
Enabling SSL Authentication in RMS	17
Enabling One-way SSL Authentication in RMS	17
Enabling Two-way SSL Authentication in RMS	18
Updating Shared JDBC Connection in Runtime	18
Enabling WebStudio Client Side Logging	19
RMS Server Configuration Property Reference	19
RMS Projects and Artifacts	35
Starting and Logging in to RMS	35
Status Check for RMS Connection	35
Checking Out a Project	36
Updating (Synchronizing) a Project	36
Synchronizing Artifacts Changes into RMS	37
Locking an Artifact	38
Unlocking a Locked Artifact	38
Managing Locks on Artifacts	38
Committing Artifacts for Approval	39
Reverting Artifact's Update	40
Viewing History of an Artifact	40
History Window Reference	41
Revisions Panel Reference	41
Details Panel Reference	42

Removing a Project from Workspace	42
Removing a Project from Workspace Using the My Workspace Tab	42
Removing a Project from Workspace Using the Dashboard Tab	43
Validating Multiple Artifacts of a Project	43
Deleting Errors and Warnings	43
Keyboard Shortcuts	44
Decision Table	45
Creating a Decision Table	46
Deleting a Decision Table	46
Renaming a Decision Table	46
Setting the Decision Table Properties	47
Enabling or Disabling the Page View	48
Analyzing a Decision Table	48
Validating a Decision Table	49
Validating a Decision Table Using Test Data	49
Exporting a Decision Table	49
Importing a Decision Table	50
Adding Condition and Action Columns from Arguments	52
Adding Custom Condition and Action Columns	53
Defining Custom Columns Using Substitution	53
Setting Default Value in Decision Table Column	54
Viewing or Hiding Columns in the Decision Table	54
Resizing a Column	55
Removing a Column	55
Sorting the Decision Table	55
Setting Row Priorities	56
Filtering the Rows	56
Creating Duplicate Rows	57
Removing a Row	57
Supported Operators	57
Editing Domain Models	60
WebStudio Domain Model Editor Reference	60
Business Rules	63
Adding Business Rules	63
Adding a Business Rule Using the HTML Form	63
Adding a Business Rule Using the Builder Interface	63
Creating a New Business Rule	63
Adding a Conditional Clause Using the Condition Builder	64
Adding an Action Using the Command Builder	65

Exporting a Business Rule	65
Deleting a Business Rule	66
Renaming a Business Rule	66
Business Rule Operators	67
Process	70
Adding a New Process	70
Deleting a Process	71
Process Palette Reference	72
Renaming a Process	73
Approval Workflow Overview	74
Worklist	75
Viewing Differences between Committed and Previous Version of Artifact	75
Approving or Rejecting a Commit Request from Worklist	75
Delegating a Workitem from Worklist	76
Deleting Workitems from Worklist	77
Generating Deployable Files (EAR and Class Files)	78
Generating the Project EAR or All Project Class Files	78
Generating and Deploying the Decision Table's Class File	79
Generating and Deploying Business Rules	80
Generated Files Location	81
Settings	82
Preferences Settings	82
Application Preferences Settings	83
User Interface Customization	84
Customizing the Company Logo	84
Customizing CSS	84
Customizing Images	85
Customizing Messages and Properties	85
Changing WebStudio Language	85
Adding a New Custom Language Pack	86

Figures

Display Text in Business Rule Editor	12
Display Text in Decision Table Editor	13
German Display Model	14
Importing a Decision Table Flowchart	51
Process Editor in TIBCO BusinessEvents WebStudio	70
WebStudio German Language Login Page	86

TIBCO Documentation and Support Services

Documentation for this and other TIBCO products is available on the TIBCO Documentation site. This site is updated more frequently than any documentation that might be included with the product. To ensure that you are accessing the latest available help topics, please visit:

<https://docs.tibco.com>

Product-Specific Documentation

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

`TIBCO_HOME/release_notes/TIB_businessevents-standard_version_docinfo.html` where `TIBCO_HOME` is the top-level directory in which TIBCO products are installed. On Windows, the default `TIBCO_HOME` is `C:\tibco`. On UNIX systems, the default `TIBCO_HOME` is `/opt/tibco`.

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

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

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support:

- 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.

How to Join TIBCOCommunity

TIBCOCommunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOCommunity offers forums, blogs, and access to a variety of resources. To register, go to the following web address:

<https://www.tibcommunity.com>

TIBCO BusinessEvents WebStudio Concepts

The TIBCO BusinessEvents WebStudio along with the Rule Management Server (RMS) provides an online tool to create and manage business rules and decision tables.

Rules Management Server (RMS)

Rules Management Server (RMS) is a server-based component that manages the repository of the TIBCO BusinessEvents projects, lifecycle of the WebStudio artifacts and WebStudio user access.

RMS provides an easy, secure, and scripted deployment lifecycle. RMS is supported on all platforms that support TIBCO BusinessEvents. Users can access RMS remotely. RMS does not have to be installed on users' machines.

Under certain circumstances, the use of RMS is optional. (see *TIBCO BusinessEvents Decision Manager User's Guide*).

You may require some basic configuration before you use RMS. You also set up the project resources required for business users to create decision tables or business rules for each TIBCO BusinessEvents project. See [Configuring RMS Server Properties](#). Administrators can configure authentication and access control as needed. See *TIBCO BusinessEvents Administration* for authentication and access control configuration.

RMS enables you to check out projects from the repository. The necessary project resources are loaded into the cache. If you have the TIBCO BusinessEvents Decision Manager add-on installed, then the artifacts are also copied into the client's machine. After you are finished working with the projects, you check in the modified artifacts to RMS for approval. See [WebStudio and RMS User Workflow](#) for a more detailed overview.

RMS is implemented using TIBCO BusinessEvents. Knowledgeable TIBCO BusinessEvents users can customize it. The example projects are located in the `BE_HOME/examples/standard/WebStudio` directory, you can configure the RMS to take projects from SVN location as well. Note that documentation is provided for the product as it is shipped. If you customize RMS some documentation may not apply.

TIBCO BusinessEvents WebStudio Overview

As the TIBCO BusinessEvents WebStudio software is an online component, business users can create or manage business rules in a web browser.

In the TIBCO BusinessEvents WebStudio software you can define an executable rule (business rule) based on the rule template and the rule template view defined by the developer in TIBCO BusinessEvents Studio. Similar to TIBCO BusinessEvents Decision Manager, a decision table in TIBCO BusinessEvents WebStudio is defined using the virtual rule function. RMS also supports lifecycle of domain model, where developers create the domain models in BusinessEvents Studio and can edit them in Studio as well as WebStudio. Rule templates, rule template views, and virtual functions are created in TIBCO BusinessEvents Studio by developers and are stored in the Rule Management Server (RMS) repositories. Also, similar to the Process Orchestration, you can design or update a process diagram in TIBCO BusinessEvents WebStudio; however, you cannot perform input and output mapping in WebStudio. You can add details specific to General and Documentation Tabs. The idea here is that you can design the process in WebStudio, add the basic details in the **General** tab, and specify the implementation details in the **Documentation** tab. The developers can use these details for implementation through TIBCO Business Events Studio.



The decision tables and process can be viewed or edited only if TIBCO BusinessEvents Decision Manager and TIBCO Process Orchestration add-ons are installed, respectively.

Administrators can define access control settings and can approve or reject commit requests from Decision Manager, Process Orchestration, and TIBCO BusinessEvents WebStudio users, check on the

status of all such requests, and keep track of all project versions. See [Working with the Worklist Items](#) for more details.

As an administrator you can also deploy decision tables or business rules when they are ready for use, either in an enterprise archive (EAR) file, or as class files. You can deploy class files for use at startup, or perform hot deployment. See [Generating Deployable Files \(EAR and Class Files\)](#) for more details.

RMS REST API

You can use RMS through a REST based webservice using the APIs provided with WebStudio. Using these APIs you can perform complete WebStudio functionalities:

- for creating, modifying, or deleting artifacts (business rule, decision table, process)
- for lifecycle operations, such as, to checkout, commit, or approve
- for worklist operations, such as, approval, rejection, or delegation of a worklist item
- for validating artifacts or analyzing decision tables.

After starting the RMS server, you can access API documentation, using the following URL:

`http://<hostname>:<port>/WebStudio/apidoc`

For example, the default URL to access the WebStudio API documentation is:

`http://localhost:8090/WebStudio/apidoc/`

Virtual Rule Functions and Decision Tables

In TIBCO BusinessEvents WebStudio, business users add decision table resources to VRFs. The decision table provides the body to the VRF, also known as the VRF implementation.

In TIBCO BusinessEvents Studio, technical users add *virtual rule functions* (VRFs) to a TIBCO BusinessEvents project. A VRF has no body, similar to a Java interface. Its implementation is provided using decision tables authored in TIBCO BusinessEvents WebStudio. VRFs are used in the TIBCO BusinessEvents project like any other rule function; they can be called from rules or other rule functions.

For example:

```
/**
 * @description
 */
virtual void rulefunction Virtual_RF.Applicant_VirtualRuleFunction {
    attribute {
        validity = ACTION;
    }
    scope {
        Concepts.Applicant    applicant;
        Events.ApplicationReceived    applicationreceived;
    }
    body {
    }
}
```

One VRF can have more than one decision table. If a VRF has more than one decision table, functions in TIBCO BusinessEvents determine how the tables are used.

Rule Building with Decision Tables

As a business user, you check out projects from RMS, build decision tables, and submit them for approval in TIBCO BusinessEvents WebStudio.



Decision tables are available in TIBCO BusinessEvents WebStudio in the view or edit mode, only if you have TIBCO BusinessEvents Decision Manager add-on installed on your system.

Decision tables provide a graphical way to build complex business rules. You create table columns by dragging and dropping predefined properties onto the decision table editor. The properties belong to

ontology resources defined in the TIBCO BusinessEvents project. However, you can only use the properties specified in the VRF. Columns can be created in other ways too. You then define threshold values (conditions) and actions in the cells of the table. Each row can be thought of as one rule in a table made up of many rules. The individual rules are often straightforward, as in the following examples.

Three Rule Conditions:

```
Person.age < Max(20, Parent.age)
Person.creditscore >= Math.function(...)
Person.gender == "female"
```

Three Rule Actions:

```
Application.status = "ACCEPTED"
Application.credit = 4000
sendNotification()
```

However, one decision table can consist of hundreds, even thousands of rules each of which is executed only when its specific conditions are satisfied. In WebStudio a decision table could be displayed in multiple pages to make them easily manageable.

Exception Tables

Each decision table can optionally have another table known as an exception table. The purpose of the exception table is purely organizational: it enables you to separate the business logic of the main decision table (added by business users) from any non-business logic (generally added by technical users). For example, in the exception table, you could capture situations where fields are blank or contain invalid values, and define actions that send notifications or set return values. The rows of a decision table along with the rows of its exception table are considered in an RTC. If you prefer, you can put non-business logic in the main table instead of using an exception table.

Table Analyzer

The Table Analyzer feature analyzes decision tables and reports problems, such as uncovered ranges for conditions, uncovered domain entries, different set of actions for identical conditions.

Rule Template and Business Rule

A rule template is a specialized rule that contains a pre-conditions section which defines the pre-conditions that must be met in addition to the conditions defined by the business user in a business rule.

The Action Context section of a rule template defines all possible actions that can be taken by a business rule (after all conditions are met). Only the action context statements that the WebStudio user selects and defines as commands in the business rules are actually taken (depending on the rule evaluation at runtime). For a business rule execution to succeed, a business rule definition must include all Actions and the order should be maintained. Completing the definition of an action is a WebStudio user task. If bindings are used (and a view) then in WebStudio, the business rule writer has to enter only the binding values to complete the definition. Action context statements are of three types: create, modify, and call, plus arbitrary actions.

Rule Building with Business Rules

As a business user, you check out projects from RMS, build business rules, and submit them for approval in TIBCO BusinessEvents WebStudio.

Business rules provide builder or a user-friendly HTML form to build rules. In a builder you specify the conditions in the “when” section and the actions in the “then” section. You can use artifacts and supported operators for building conditions and actions. In the user-friendly HTML form, you fill up the values for the conditions and actions.

WebStudio and RMS User Workflow

The rules authored in TIBCO BusinessEvents WebStudio are made available to a TIBCO BusinessEvents application after following a complete approval workflow.

Step 1: RMS Project Setup

A TIBCO BusinessEvents user creates a TIBCO BusinessEvents project, adding the ontology, and writing rules that make use of virtual rule functions (VRFs) and Rule Templates.

The TIBCO BusinessEvents project for RMS is stored at the location defined by a property in the RMS server configuration. The RMS project requires an access control file for authorization.

The RMS server must be running so that the login and workflow actions are available in the TIBCO BusinessEvents WebStudio component.

Step 2: TIBCO BusinessEvents WebStudio

A business user starts the TIBCO BusinessEvents WebStudio component, logs on to RMS, and checks out the project. The business user creates one or more decision tables, business rules, or process and saves the modified project locally, then commits them for approval.

Step 3: Approval

An RMS user (with the "approval" access control rights) working in TIBCO BusinessEvents WebStudio receives the request and reviews the checked-in artifacts and then approves or rejects them. The approved artifacts are available for subsequent checkouts or updates.

Step 4: Generating Deployable Files

An RMS user generates deployable files for resources that are ready for deployment. You can generate EAR files or class files.

Step 5: Deployment

EAR files are deployed in the usual way, as explained in TIBCO BusinessEvents Administration. Class files for decision tables and rule template instance files for business rules are deployed by placing them in a configured location recognized by the TIBCO BusinessEvents engines at start up. They can also be hot-deployed.

Display Models in WebStudio

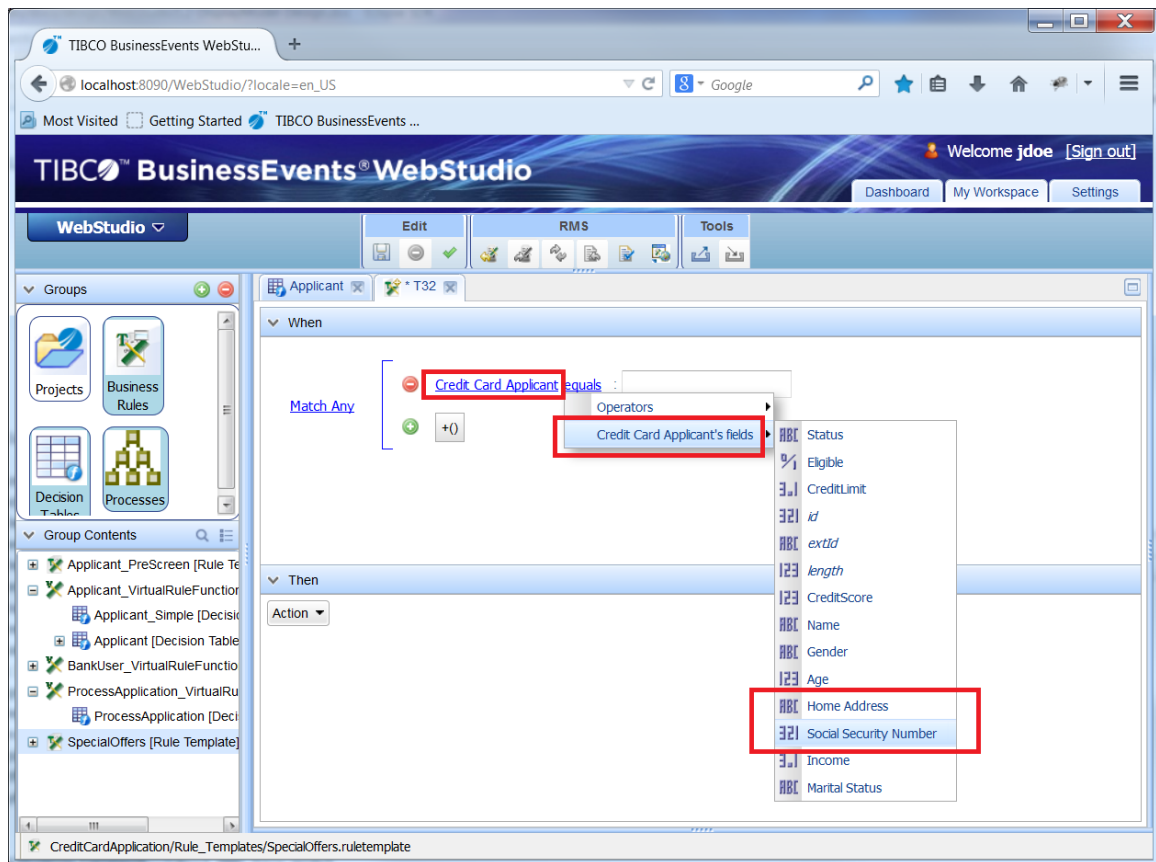
Using display models, you can display text for the artifact and its properties in a simpler form and your preferred language. You can also hide some properties of the artifact using display models.

In the WebStudio run time, the display text for the properties, which are defined in the display model, are displayed instead of the element name. The display model for the artifact and its properties are created in the TIBCO BusinessEvents Studio. See the *TIBCO BusinessEvents Developer's Guide* for more details on how to create display models.

Business Rule Editor

In the Business Rule editor, the display text for the fields are displayed instead of their element name. If the property is left unspecified in the display model, the element name is displayed. Also, those properties are displayed for which the hidden check box was not selected.

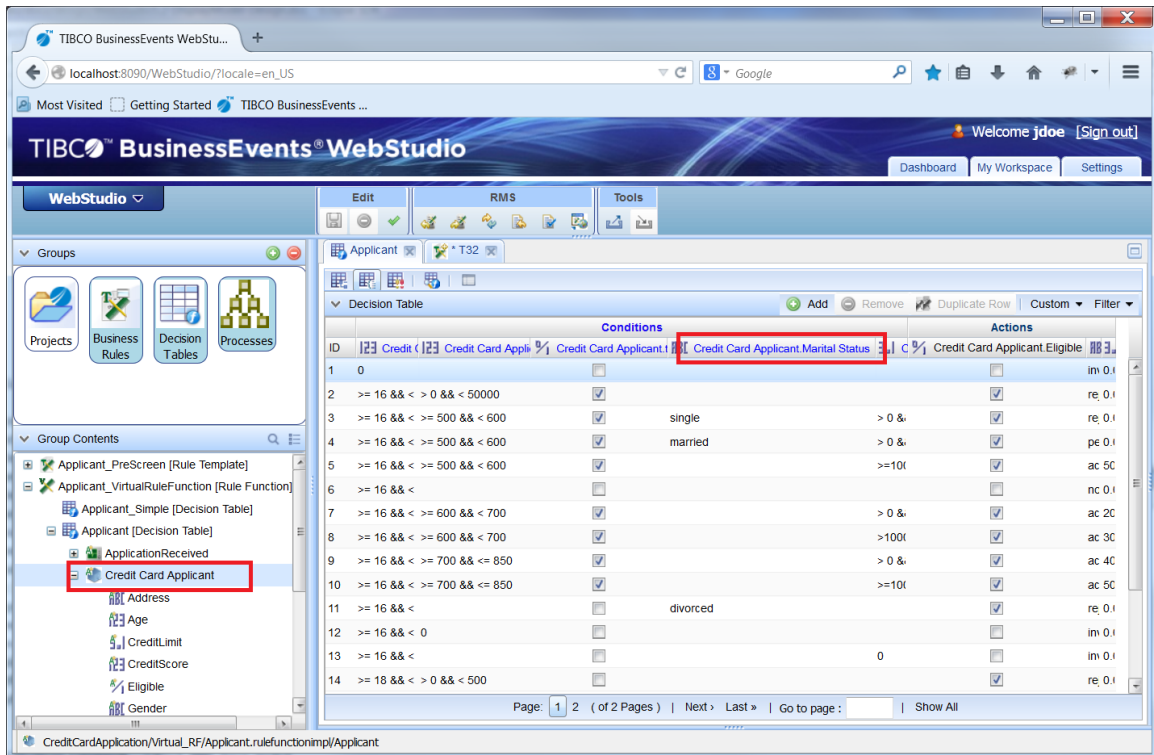
Display Text in Business Rule Editor



Decision Table Editor

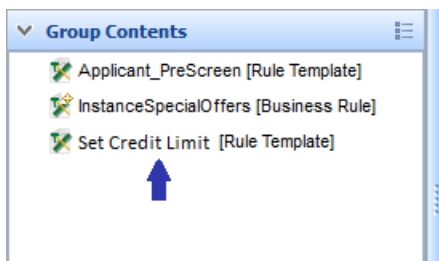
In the Decision Table editor, the display text is displayed in the column heading and also for the arguments under the Decision Table in the Group Content section. All the arguments which are marked as hidden in the display model are not displayed.

Display Text in Decision Table Editor



Group Contents

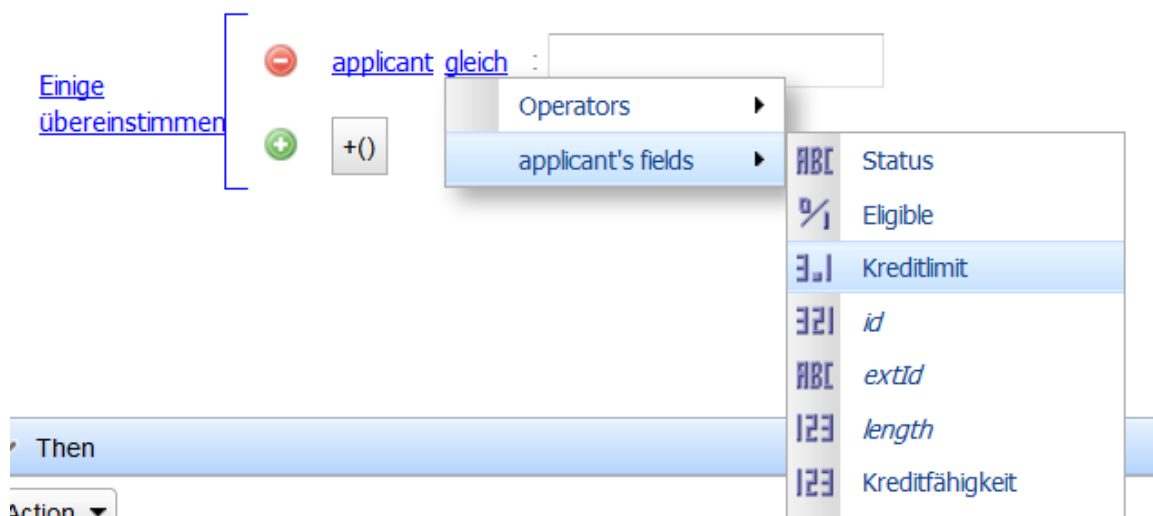
In the **Group Contents** section, all the base artifacts, which have display models defined, have their display text displayed instead of the artifact name.



Other Language

When the locale for Webstudio is set to a non-default language in the URL, the display model for that language is loaded in the WebStudio. For example, if the locale is set for German in the WebStudio URL, such as, <http://localhost:8090/WebStudio/?locale=de>, the display model with the language defined as German is picked up and used.

German Display Model



If the display model with the language and country does not match any language packs, only the display model for that language is loaded in the WebStudio. The default language 'en_US' will be picked up for WebStudio display.

Configuring RMS

Artifacts need to be added to RMS, so that they can be used by TIBCO BusinessEvents WebStudio. Advanced configuration might also be required to enable certain features, such as, SSL authentication.

Adding a Project to RMS

TIBCO BusinessEvents projects, with VRF and rule templates, need to be added to the RMS repository to make them accessible in the TIBCO BusinessEvents WebStudio.

All RMS projects exist in a repository under a *root location* directory and their access control list (ACL) file exist under a *RMS config* location, which are `BE_HOME/examples/standard/WebStudio` and `BE_HOME/rms/config/security` respectively, as shipped. The root location and RMS config location are configurable using `ws.scs.rootURL` and `ws.projects.acl.location` properties in the `RMS.cdd` file, respectively.

Prerequisites

A TIBCO BusinessEvents project created in TIBCO BusinessEvents Studio.

Procedure

1. Under the RMS project root location directory, create a directory with a name appropriate for the project.
The root location is configurable. See [Configuring RMS Server Properties](#).
2. Copy the TIBCO BusinessEvents Studio project contents that will be used in Decision Manager add-on or in TIBCO BusinessEvents WebStudio to the project directory.
3. If the project contains custom functions, create the *ProjectName* directory under the directory identified by the `ws.projects.customLib.location` property in the `RMS.cdd` file and then place the custom function jars under it. See [RMS Server Configuration Property Reference](#) for more information on the property.
4. Copy the access control file for the project (`RMSProjectName.ac`) to the RMS configuration directory. See *TIBCO BusinessEvents Administration* for authentication topics.
The RMS configuration location is configurable. See [Configuring RMS Server Properties](#).
5. Restart the RMS server.

Configuring RMS Server Properties

RMS works out of the box on a local machine. Configuration is required to enable access by remote TIBCO BusinessEvents WebStudio and the Decision Manager clients.

You may need to make other changes to the server configuration. For example, if you change the server location or the location of the RMS project repository, you must update the RMS server properties accordingly.

To configure the RMS server properties, edit the `RMS.cdd` file. You cannot edit a CDD file in TIBCO BusinessEvents Studio outside of its project context. To edit the CDD in TIBCO BusinessEvents Studio, import the BRMS project into your workspace. After that, copy the `RMS.cdd` file to its installed locations, in the following way



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

See [RMS Server Configuration Property Reference](#) for information about each property.



- As with any procedure that changes files, first backup any files that could be affected. The `RMS.cdd` file is located in two places, in the product as shipped:
 - `BE_HOME/rms/project/BRMS/Deployment/RMS.cdd`
 - `BE_HOME/rms/bin/RMS.cdd`
- After you update the `RMS.cdd` file in your workspace, overwrite the existing files with the updated ones, in the above locations (or the ones currently in use, if different).
- Ensure that the CDD files in the BRMS project and the RMS startup directory are kept in sync.

Procedure

1. In TIBCO BusinessEvents Studio, choose **File > Import > Existing Projects into Workspace**.
2. Ensure the **Copy the projects into workspace** check box is selected.
3. Select the following project:
 - `BE_HOME/rms/project/BRMS`
4. In Studio Explorer, double-click **RMS.cdd** to open it in the CDD editor.
5. In the CDD editor Cluster tab, click **Properties**. In the Configuration panel on the right, expand groups to see the individual properties. Update the properties value, as per your requirement, as explained in [RMS Server Configuration Property Reference](#).
6. When you have finished editing, save the `RMS.cdd` file.
7. In the file system, copy the `RMS.cdd` file from your workspace to the BRMS project and to the RMS server startup location.:

The locations are:

 - `BE_HOME/rms/project/BRMS/Deployment/RMS.cdd`
 - `BE_HOME/rms/bin/RMS.cdd`
8. Restart the RMS server.

Enabling Remote Connection to RMS from WebStudio

For production settings, specify the RMS server hostname and port to ensure that TIBCO BusinessEvents WebStudio users can connect to RMS from remote machines.

As shipped, the `tibco.clientVar.Webstudio/hostname` property is set to `localhost`. This setting enables the product to be used on a single machine.

Procedure

1. Import the BRMS project into your workspace and open the `RMS.cdd` file for editing. See [Configuring RMS Server Properties](#) for details.
2. In the CDD editor Processing Units tab, click **WS-Inference**. In the Properties panel on the right, expand the WS property group.
3. Specify the TIBCO BusinessEvents WebStudio hostname and port using the following properties:
 - `tibco.clientVar.Webstudio/hostname`
 - `tibco.clientVar.Webstudio/port`
4. Save the `RMS.cdd` file.
5. In the file system, copy the `RMS.cdd` file from your workspace to the BRMS project and to the RMS server startup location. The locations are:

- `BE_HOME/rms/project/BRMS/Deployment/RMS.cdd`
 - `BE_HOME/rms/bin/RMS.cdd`
6. Restart the RMS server.

Enabling SSL Authentication in RMS

The SSL authentication process uses certificates that are issued by a certificate authority. The same process applies if the certificates are issued by a certificate generation utility or if self-signed certificates are used.

To establish a SSL connection an application acting as an SSL client contacts an application acting as an SSL server. You can establish either a one-way SSL authentication or two-way SSL authentication for the application.

Enabling One-way SSL Authentication in RMS

One-way SSL authentication enables the application operating as the SSL client, which in this case is web browser, to verify the identity of the application operating as the SSL server (RMS). The SSL-client application is not verified by the SSL-server application.

Procedure

1. Import the BRMS project in the TIBCO BusinessEvents Studio and copy it into your workspace. Now open the `RMS.cdd` file for editing. See [Configuring RMS Server Properties](#) for details.
2. In the **CDD editor Processing Units** tab, click **default**.
In the **Properties** panel on the right, you can see the `be.channel.deactivate` property.
3. Remove the `/WebStudio/Core/Channels/WS_CH_Secure_WebstudioChannel` value from the `be.channel.deactivate` property.
4. Save the `RMS.cdd` file.
5. In the file system, copy the `RMS.cdd` file from your workspace to the BRMS project and to the RMS server startup location.
The locations are:
 - `BE_HOME/rms/project/BRMS/Deployment/RMS.cdd`
 - `BE_HOME/rms/bin/RMS.cdd`
6. Restart the RMS server.

Result

To verify, type the URL `https://localhost/WebStudio/` in a web browser and press **Enter**. No port is required as default SSL port is used. The web browser notifies you about the untrusted certificate from the SSL server.

Enabling Two-way SSL Authentication in RMS

In two-way SSL authentication, the SSL client application, which in this case is a web browser, verifies the identity of the SSL server application (RMS), and then the SSL server application verifies the identity of the SSL-client application.

Procedure

1. Follow the steps mentioned in the [Enabling One-way SSL Authentication in RMS](#) to enable the SSL server verification, but do not restart RMS server yet.
2. In Studio Explorer, double click **BRMS > WebStudio > Core > Transports > WS_TR_Secure_WebstudioHTTPConnection.sharedhttp** to open it in the resource editor.
3. In the resource editor, click **Configure SSL**.
The SSL Configuration for HTTPS Connections window is displayed.
4. In the SSL Configuration for HTTPS Connections window, select the **Requires Client Authentication** checkbox and click **OK**.
5. In the BusinessEvents Studio, regenerate the BRMS project EAR file (RMS.ear) and save it at *BE_HOME/rms/project/*. See *TIBCO BusinessEvents Developer's Guide* for more details.
6. Restart the RMS server.
7. Open your web browser and import the *WSClientStore.p12* certificate file in the web browser from the location *BE_HOME/rms/config/security/*. Refer to your web browser's help on how to import certificate in it.

Result

To verify, type the URL <https://localhost/WebStudio/> in the same web browser and press **Enter**. No port is required as default SSL port is used. The web browser displays a confirmation window to select your SSL client certificate and send it to the SSL server. After that the web browser notifies you about the untrusted certificate from the SSL server.

Updating Shared JDBC Connection in Runtime

Edit the *RMS.cdd* file to add the JDBC connection to the backing store from RMS at runtime.

Procedure

1. Import the BRMS project in the TIBCO BusinessEvents Studio and copy it into your workspace.
Now open the *RMS.cdd* file for editing.
See [Configuring RMS Server Properties](#) for details.
2. In the **CDD editor Cluster** tab, click the **Properties** in the Cluster pane and expand **node** under the Configuration pane.
3. Expand the **BackingStore** property group and edit the value for backing store properties as required.
See [Backing Store Property Group](#) for more details on the properties.
4. Save the *RMS.cdd* file
5. In the file system, copy the *RMS.cdd* file from your workspace to the BRMS project and to the RMS server startup location.
The locations are:
 - *BE_HOME/rms/project/BRMS/Deployment/RMS.cdd*

- `BE_HOME/rms/bin/RMS.cdd`

6. Restart the RMS server.

Enabling WebStudio Client Side Logging

You can enable the logs for the WebStudio client for logging messages that are submitted by the client. Webstudio client-side logging can be enabled by configuring the intended logging flags in the `WebStudio.html` file.



WebStudio client-side logging is different from RMS logging, which can be configured using the `RMS.cdd` file.

Procedure

1. Go to the `BE_HOME/rms/bin/` folder and extract the `WebStudio.war` package in a separate folder.
2. From the extracted files, open the `WebStudio.html` file in the edit mode.
3. In the `WebStudio.html` file, set the javascript variable `webstudio_logging_enabled` to `true`. The WebStudio client-side logging is now available.
4. Save the `WebStudio.html` file and repackage all the extracted files to the `WebStudio.war` package.
5. Restart RMS.
The WebStudio client log is created at `BE_HOME/rms/bin/logs/webstudio_client.log`.

RMS Server Configuration Property Reference

To configure the RMS server properties, edit the `RMS.cdd` file.

See [Configuring RMS Server Properties](#) for the related procedure.

Properties not documented in this table are either for internal use or are for other types of configuration explained elsewhere in the documentation.

Common Property Group

Property	Default Value	Description
<code>rms.checkin.revisionId.initValue</code>	10000	Initial value of the revision number to use for project check in.
<code>ws.artifact.deploy.location</code>	<code>BE_HOME/rms/shared</code>	Name of the directory for storing EAR files that are generated for deployment. Separate directory for each project is created, where EAR files for the project are stored. See also <code>be.codegen.rootDirectory</code> .

Property	Default Value	Description
<code>be.codegen.rootDirectory</code>	Codegen	<p>Name of the directory for storing class files generated by the Generate Deployable menu option. The directory location is relative to the <code>ws.artifact.deploy.location</code> directory.</p> <p>If this property is not present, then files are stored in the deployment directory.</p> <p>If this property is present but has no value, a directory is created with the default name Codegen.</p>
<code>be.codegen.useLegacyCompilation</code>	False	<p>Set the value to true, if you want to use the file-based legacy compiler to build the EAR file. By default, the value is false, that means EAR files are built in memory.</p>
<code>ws.projects.customLibLocation</code>	<code>BE_HOME/rms/lib/ext</code>	<p>Path of the directory for storing custom function libraries for a WebStudio project. The custom function libraries are saved in a folder (with the same name as of the project name) inside the directory identified by the <code>ws.projects.customLibLocation</code> property.</p> <p>For example, if the value of the <code>ws.projects.customLibLocation</code> property is <code>C:/Tibco/BE52/be/5.2/rms/lib/ext</code>, then the custom function libraries for the MyTestProject project should be saved in the location <code>C:/Tibco/BE52/be/5.2/rms/lib/ext/MyTestProject</code>.</p>

Property	Default Value	Description
<code>ws.projects.projectLib.location</code>	<code>BE_HOME/rms/project-lib</code>	<p>Path of the directory for storing project libraries for a WebStudio project. The project libraries are saved in a folder (with the same name as of the project name) inside the directory identified by the <code>ws.projects.projectLib.location</code> property.</p> <p>For example, if the value of the <code>ws.projects.projectLib.location</code> property is <code>C:/Tibco/BE52/be/5.2/rms/project-lib</code>, then the project libraries for the MyTestProject project should be saved in the location <code>C:/Tibco/BE52/be/5.2/rms/project-lib/MyTestProject</code>.</p>
<code>ws.validateDT.ignoreDeploymentableStatus</code>	<code>false</code>	<p>Specifies whether to ignore the deployment status of the artifact at server restart and perform decision table validation without regenerating deployables.</p> <p>This property is required when persistence is not enabled for the system. When the persistence is enabled, this property is not needed as the deployment status is available across server restarts.</p>

RMS Property Group

Property	Default Value	Description
<code>rms.project.workspace</code>	<code>workspace</code>	Not used in this release
<code>rms.project.decisiondata</code>	<code>decisiondata</code>	Not used in this release

Property	Default Value	Description
<code>rms.project.deployement</code>	deployment	Not used in this release
<code>rms.workflowstages.config.file</code>	<code>BE_HOME/rms/config/RoleWorkflowStages.xml</code>	Not used in this release
<code>rms.lockworkflowstages.config.file</code>	<code>BE_HOME/rms/config/LockWorkflowStages.xml</code>	Not used in this release Default is <code>BE_HOME/rms/config/LockWorkflowStages.xml</code>
<code>rms.roleArtifactTypes.config.file</code>	<code>BE_HOME/rms/config/RoleApplicableArtifactTypesConfig.xml</code>	Not used in this release Default is <code>BE_HOME/rms/config/RoleApplicableArtifactTypesConfig.xml</code>
<code>rms.external.entities.autodetect</code>		Used internally. Do not change the value.

Authentication and Authorization (ACL) Properties

User authentication topics are common to TIBCO BusinessEvents and the add-on products that use authentication. Common options are fully documented in *TIBCO BusinessEvents Administration*. The relevant properties are also listed as follows for your convenience.

Property	Default Value	Description
<code>be.auth.type</code>		Specifies the authentication mechanism used. Additional configuration is required. The values are: <ul style="list-style-type: none"> file ldap openldap Note: Authentication topics are documented in the <i>TIBCO BusinessEvents Administration</i> guide.

Property	Default Value	Description
be.auth.file.location	BE_HOME/rms/config/security/users.pwd	Location of the authentication file used for file-based authentication. Note: Authentication topics are documented in the <i>TIBCO BusinessEvents Administration</i> guide.
java.security.auth.login.config	BE_HOME/rms/config/security/jaas-config.config	Location of the JAAS login configuration file. You can substitute a different implementation of the JAAS login module than the one provided. Note: Authentication topics are documented in the <i>TIBCO BusinessEvents Administration</i> guide.
ws.projects.acl.location	BE_HOME/rms/config/security	Location of the directory used for all ACL (authorization) files. Files must be named using the format <i>RMSProjectName.ac</i> .

RMS-GVs Property Group

This group of properties override the values of global variables that are provided in the TIBCO BusinessEvents Studio BRMS project. These properties are made available as global variables so that the properties can be overridden at runtime, as needed.

You can override the default global variable values in the `RMS.cdd` file for command-line startup. For details on overriding global variable values when deploying with TIBCO BusinessEvents Monitoring and Management or with TIBCO Administrator, see *TIBCO BusinessEvents Administration*.

For defining and using global variables in TIBCO BusinessEvents Studio projects, see *TIBCO BusinessEvents Developer's Guide*.

Property	Default Value	Description
tibco.clientVar.RMS/hostname	localhost	Specifies the host name or IP address of the machine where RMS is hosted. Remote clients can connect to the server at this location.

Property	Default Value	Description
<code>tibco.clientVar.RMS/port</code>	5000	Specifies the port number of the machine where RMS hosted. This port is used for listening to client requests. See notes for <code>tibco.clientVar.RMS/hostname</code> .
<code>tibco.clientVar.RMS/Approval/adminRole</code>	Administrator, Business-User	<p>The multiple roles specified here have Administrator role permissions. When you change the default value, the Administrator role still has Administrator permissions.</p> <p>You can also specify project specific administrator roles. To specify project specific admin roles, append the role to the project name separated by a colon (:). Ensure that the role has the approve permission set for the project.</p> <p>For example,</p> <pre>tibco.clientVar.RMS/Approval/adminRole = Administrator,CreditCardApplication:Business-User</pre> <p>This shows the worklist items for the CreditCardApplication project, for users with the Business_User roles, while for all other projects, the worklist items show up for users under the Administrator role.</p> <p>Ensure that you replace the XML special character in the administrator role name value (if present) with the following characters:</p> <ul style="list-style-type: none"> • "&" by "&"; • "' " by "&apos;";

WebStudio Property Group

Property	Default Value	Description
tibco.clientVar.Webstudio/ warDir	rms/bin/WebStudio.war	Path to the Web archive for the client side representation of the TIBCO BusinessEvents WebStudio.
tibco.clientVar.Webstudio/ hostName	localhost	Specifies the host name or IP address of the machine where RMS server is hosted for WebStudio.
tibco.clientVar.Webstudio/ port	8090	Specifies the port number of the machine where RMS server is hosted for WebStudio. This port is used for listening to client requests. See notes for tibco.clientVar.RMS/hostname.
tibco.clientVar.Webstudio/ Connection/timeout	60000	Timeout interval for the HTTP channel. This value is in milliseconds.
tibco.clientVar.Webstudio/ sessionTimeOut	30	Time interval for which an inactive user session will be valid. This value is in minutes.
tibco.clientVar.Webstudio/ DecisionTable/pageSize	20	Number of rows of the decision table to be displayed in a page.
ws.validateDT.temp.dir	rms/temp-dir	Directory which is used as an interim storage of decision table contents during the validation process.

Global Variable Properties for One-Way SSL Between RMS Server and Decision Manager Clients

Property	Default Value	Description
tibco.clientVar.RMS/ security/securePort	8443	The port to be used for secure HTTP communication in BusinessEvents Studio.

Property	Default Value	Description
<code>tibco.clientVar.RMS/security/sslCertificateStore</code>	<code>BE_HOME/rms/config/security/SSLKeyStore.ks</code>	Absolute path of the JKS keystore which contains the RMS server's public/private keys. If the certificate used is CA signed, include the CA's certificate.
<code>tibco.clientVar.RMS/security/sslCertificateStorePassword</code>	<code>changeit</code>	The password for the keystore.
<code>tibco.clientVar.RMS/security/secureWebPort</code>	<code>443</code>	The port to be used for secure HTTP communication in WebStudio.

Properties for One-Way SSL Between LDAP Server, RMS Server, and Decision Manager Clients



- For one-way SSL between LDAP server and RMS server, you must also change the value of `be.auth.ldap.port`.
- For one-way SSL between RMS server and Decision Manager clients, you must also set the value of `be.auth.ldap.port`.

Property	Default Value	Description
<code>be.auth.ldap.ssl</code>		If set to true, enables SSL between the RMS server and the LDAP server If set to false or not set, the LDAP authentication uses a non-secure channel.
<code>javax.net.ssl.trustStore</code>		The absolute path of the keystore containing the LDAP server certificate chain.
<code>javax.net.ssl.trustStorePassword</code>		The password for the keystore.
<code>javax.net.ssl.trustStoreType</code>		The keystore type for keystores other than JKS type, for example, PKCS12.

Email Notification Property Group

Email notification properties are used to send email notification when an artifact undergoes a status change in approval workflow, such as, artifact commit and artifact review (Approve, Reject, BuildAndDeploy).

In addition to these properties, you can define any SMTP protocol specific properties depending on the mail server configuration. These properties should be defined with a prefix `ws.notify.prop`, for example, `ws.notify.prop.mail.smtps.auth`, `ws.notify.prop.mail.smtps.ssl.enable`.

Property	Default Value	Description
<code>ws.notify.enabled</code>	False	If set to true, send email notification. If set to false, do not send email notification.
<code>ws.notify.file.location</code>	<code>BE_HOME/rms/config/notify/UserNotify.Ids</code>	Specifies the location of the <code>.IDS</code> file which contains the mapping of WebStudio user IDs to the email IDs. This field is used in the case of file based authentication.
<code>ws.notify.ldap.userNotifyIdAttr</code>	<code>userPrincipalName</code>	Specifies the name of the LDAP attribute for the email ID. This field is used in the case of LDAP based authentication.
<code>ws.notify.impl.class</code>	<code>com.tibco.be.ws.notification.impl.EmailNotification</code>	Specifies the implementation class for email notification support.
<code>ws.notify.context.impl.class</code>	<code>com.tibco.be.ws.notification.impl.EmailNotificationContext</code>	Specifies the context implementation class for email notification support.
<code>ws.notify.message.template.file</code>	<code>BE_HOME/rms/config/notify/message.stg</code>	Specifies the location of the message template file, which contains template (subject and content) for the commit, reject, and approve notification emails.

Property	Default Value	Description
<code>ws.notify.mail.domain</code>		Specifies the domain on the email. This is used with the RMS username to construct the email address to which the notification is to be sent. For example, if RMS username is "admin" and the property is set to "tibco.com", the email address for the notification is admin@tibco.com.
<code>ws.notify.mail.receiver.cc.emails</code>		<p>Comma separated list of email IDs or WebStudio user IDs to whom any commit, approve, or reject notifications need to be sent. In the case of CC notifications to the WebStudio users, specify their login IDs. The email IDs are picked up from the .IDS file identified by the ws.notify.file.location property, if present; otherwise, the email ID is evaluated as <i><login-ID>@<mail-domain></i>.</p> <p>In the case of LDAP based authentication, the email ID is picked up from the user's LDAP entry.</p>
<code>ws.notify.prop.MAIL_PROTOCOL</code>	SMTPS	Specifies the email protocol to be used to send the email. SMTP and SMTPS are valid values.
<code>ws.notify.prop.MAIL_SERVER_HOST</code>		Specifies the host name of the email server.
<code>ws.notify.prop.MAIL_SERVER_PORT</code>		Specifies the port at which the mail server is listening.
<code>ws.notify.prop.SENDER_EMAIL</code>		Specifies the email address to be used for sending the notification email.

Property	Default Value	Description
<code>ws.notify.prop.SENDER_USERNAME</code>		Specifies the sender's username for authentication to email server.
<code>ws.notify.prop.SENDER_PASSWORD</code>		Specifies the sender's email password for authentication to email server.
<code>ws.notify.prop.mail.smtps.auth</code>	<code>true</code>	This is a SMTP protocol-specific property. Specifies whether the authentication is checked or not for the email server.
<code>ws.notify.prop.mail.smtps.ssl.enable</code>	<code>true</code>	This is a SMTP protocol-specific property. Specifies whether the email server is SSL enabled.

Hot Deployment Property Group

Use the hot deployment property group to configure JMX connection details in the CDD file and hot deploy the business rule and the decision table directly from WebStudio. You can configure JMX connection details for each project for each environment. Using these properties you can deploy multiple projects in multiple environments. For example, for deploying the CreditCardApplication project in the "QA" environment, set the QA environment in the `CreditCardApplication.ws.applicableEnvironments` property and set the JMX connection details in the `CreditCardApplication.QA.ws.jmx.*` properties.

Property	Default Value	Description
<code><ProjectName>.ws.applicableEnvironments</code>	<code>QA, PROD</code>	<p>Specifies the environment where the artifacts can be deployed.</p> <p>Add this property for each project you want to deploy. For example, if you want to deploy the CreditCardApplication project in the PROD and the QA environment, set the <code>CreditCardApplication.ws.applicableEnvironments</code> property value to "PROD,QA".</p>
<code><ProjectName>.<EnvironmentName>.ws.jmx.hotDeploy.enable</code>	<code>true</code>	Specifies whether to enable or disable hot-deployment (as part of BuildAndDeploy operation) for the specified project in the specified environment.

Property	Default Value	Description
<code><ProjectName>.<EnvironmentName>.ws.jmx.host</code>	localhost	The name of the JMX host
<code><ProjectName>.<EnvironmentName>.ws.jmx.port</code>	9990	The unused port number, through which you want to enable the JMX connection
<code><ProjectName>.<EnvironmentName>.ws.jmx.user</code>		The username for the JMX connection
<code><ProjectName>.<EnvironmentName>.ws.jmx.password</code>		The password for the user name for <code><ProjectName>.<EnvironmentName>.ws.jmx.user</code>
<code><ProjectName>.<EnvironmentName>.ws.jmx.clusterName</code>		The name of the cluster to be monitored
<code><ProjectName>.<EnvironmentName>.ws.jmx.agentName</code>	inference-class	The name of the TIBCO BusinessEvents agent

Backing Store Property Group

Use the backing store property group to add a shared JDBC connection without the need to build the new EAR file (RMS.ear) for the BRMS project.

Property	Default Value	Description
<code>tibco.clientVar.Webstudio/DB/driverName</code>	<code>oracle.jdbc.OracleDriver</code>	The name of the JDBC driver class.
<code>tibco.clientVar.Webstudio/DB/url</code>	<code>jdbc:oracle:thin:@localhost:1521:orcl</code>	The URL to use to connect to the database. You must supply the portions of the URL, for example, the host, port number, and database instance name. .
<code>tibco.clientVar.Webstudio/DB/maxConnections</code>	5	The maximum number of database connections to allocate. The minimum value that can be specified is 1.
<code>tibco.clientVar.Webstudio/DB/userName</code>		User name to use when connecting to the database.
<code>tibco.clientVar.Webstudio/DB/password</code>		Password to use when connecting to the database.

Property	Default Value	Description
tibco.clientVar.Webstudio/ DB/timeout	zero	Time (in seconds) to wait for a successful database connection. Only JDBC drivers that support connection timeouts can use this configuration field. If the JDBC driver does not support connection timeouts, the value of this field is ignored. Most JDBC drivers support connection timeouts.

Source Control System (SCS) Property Group

SCS property group enables you to checkout a WebStudio project resources from a SCS location and commit the updates.

Property	Default Value	Description
ws.scs.rootURL	BE_HOME/examples/standard/ WebStudio	Specifies the RMS repository location. This is the root location for RMS project directories. If repository is present in source control, then provide the repository URL of RMS projects as the value of ws.scs.rootURL .
ws.scs.impl.type	file	Identifies the type of repository being referred. The values that configured in the product out-of-the-box are: <ul style="list-style-type: none"> file (default) svn
ws.scs.repo.dir	BE_HOME/rms/repo	The local repository directory on your system. All checked out project resources are downloaded at this location.

Property	Default Value	Description
<code>ws.scs.command.path</code>		<p>Absolute path to the source control command-line client.</p> <p>For example, if SVN is used then RMS server uses the SVN command-line tool to connect to the SVN repository. Any SVN command-line tool can be used, one commonly used one being Tortoise SVN CLI.</p>
<code>ws.scs.default.username</code>		<p>Default username to connect to the repository mentioned in the <code>ws.scs.rootURL</code> property.</p>
<code>ws.scs.default.password</code>		<p>Default password, encoded into the Base64 format, to connect to the repository mentioned in the <code>ws.scs.rootURL</code> property. For example, Password123 should be encoded with Base64 format and entered as UGFzc3dvcmQxMjM=. </p> <p>Note: WebStudio uses the default username and password (after decoding from Base64 format) to check out and check in the objects from the svn repository. You also have the option to enter your SCS username and password (in the Preference portlet under the Settings tab) in the WebStudio UI which overrides the default username and password. Enter the actual password in the WebStudio UI and not a Base64 format encoded one. See Preferences Settings for more details.</p>

Property	Default Value	Description
<code>ws.scs.locking.enable</code>	<code>false</code>	You can lock the artifacts for the WebStudio. If the lock is applied on an artifact, the other users cannot modify or delete the artifacts until the lock is released.
<code>ws.scs.lock.timeout</code>	<code>0</code>	Time in seconds before the lock is automatically released. Set the property value greater than zero to apply timeout on locks.

Cross-Origin Resource Sharing (CORS) Property Group

Using the CORS property group, you can specify which external applications can use the WebStudio APIs.

Property	Default Value	Description
<code>be.http.filter.cors.class</code>	<code>org.apache.catalina.filters.CorsFilter</code>	The filter class name for the CORS filter.
<code>be.http.filter.cors.param.cors.allowed.origins</code>	<code>*</code>	<p>A list of Origin request header values that are allowed to access the resource..</p> <p>BusinessEvents matches the value of Origin header of the HTTP request with this list. If the Origin request header value is present in the list, then the Access-Control-Allow-Origin header in the HTTP response returns the Origin request header value.</p> <p>The default value is <code>"*"</code> which means all URLs are valid, and the Access-Control-Allow-Origin header in the HTTP response returns <code>"*"</code> for all requests.</p>
<code>be.http.filter.cors.param.cors.allowed.methods</code>	<code>GET, POST, DELETE, PUT</code>	A list of HTTP methods that can be used to access the resource, using cross-origin requests. These are the methods which will also be included as part of Access-Control-Allow-Methods header in pre-flight response.

Property	Default Value	Description
be.http.filter.cors.urlpattern.1	/*	The URL pattern that triggers the filter.

RMS Projects and Artifacts

The projects and artifacts that were added to RMS can be used in WebStudio to create business rules.

Starting and Logging in to RMS

Before you can work on the RMS projects, you need to start and login to the RMS.

Procedure

1. Start the RMS server using one of the following ways:
 - Execute `BE_HOME/rms/bin/be-rms.exe` (or `be-rms.sh`, depending on your operating system), with valid arguments. For example, on Windows you would open a command window in `BE_HOME/rms/bin` and execute:


```
be-rms.exe
```
 - (Windows only) Select Start > All Programs > TIBCO > TIBCO_HOME > TIBCO BusinessEvents *version* > Start Rules Management Server.

The server is ready when you see a message in the command window such as the following:

```
Info [HTTP-Channel-Startup] - [driver.http] Channel server for HTTP Channel [Port:5000] successfully started
```
2. Type the URL `http://host:port/WebStudio/` in a web browser. Default host is localhost and default port is 8090. Host and port are configurable using properties in `RMS.cdd`. Ensure that the RMS is running before logging to TIBCO BusinessEvents WebStudio.
3. Type the username and password for TIBCO BusinessEvents WebStudio.
4. Click **Login** to sign in to RMS.

Result

The application displays the TIBCO BusinessEvents WebStudio dashboard.



The system restricts multiple simultaneous login of the same user name. However, if you have previous unclosed sessions of the same username you can select the **Force Login** check box to close all the other sessions of the user. The check box is only visible if multiple login for the user is detected.

In TIBCO BusinessEvents WebStudio, click **Sign out** to log out. In the case of session timeout, the user is automatically logged out.

Status Check for RMS Connection

Once a TIBCO BusinessEvents Studio, a TIBCO BusinessEvents Decision Manager, or a TIBCO BusinessEvents WebStudio client logs in to RMS, a Dashboard background activity periodically polls the RMS server to refresh the worklist items in WebStudio and `rms.heartbeat.relay` property (in `studio.tra`) is used by Studio to configure the heartbeat time.

By default, the delay is of two seconds.

If the server goes down, the menu options for RMS are automatically disabled. However, if you are already performing an RMS operation and the server goes down, the operation fails. The login option is enabled when the server comes up again.

Checking Out a Project

If RMS is used in your environment, then you must log into RMS and check out a project from RMS before you can work with the project resources in the TIBCO BusinessEvents WebStudio UI.


When generating an EAR file using the Generate Deployable option, the project resources from the RMS repository and not from the workspace are used.

Depending on your role or roles, you may not have permission to check out all resources or to do all of the documented tasks. For instructions on updating a project you already checked out, see [Updating \(Synchronizing\) a Project](#).



The checkout dialog does not list the imported 4.x release projects, if the project name contains a dot(.). These imported projects should be renamed inside TIBCO BusinessEvents Studio to remove the dot(.) from the project name.

Procedure



1. Open the checkout window using any of the following methods:
 - In the **Dashboard** tab, click the link in the **My Projects** section if no projects are checked out.
 - In the **My Workspace** tab, click **WebStudio**, with the down-pointing triangle in front of it, and select **RMS > Checkout**.
 - In the **My Workspace** tab, right-click inside the **Group Contents** section and select **RMS > Checkout**.
 - In the **My Workspace** tab, click the checkout icon  in the RMS toolbar.
2. In the checkout window, select the project you want to checkout from the **Project List** drop-down.
3. Select the checkboxes of the files you want to check out and click **OK**. Alternatively, select the header checkbox to select all the files for checkout and click **OK**.



You can perform partial checkouts as per your requirement, such as, checkout some files in the start and then later on checkout more files as required.

Result

In TIBCO BusinessEvents WebStudio the checked out project is displayed under Group Contents and the project files are loaded to your workspace. The Group Contents section displays the artifacts based on the groups (Projects/Business Rules/Decision Tables/Process) selected in the Groups section. The Business Rules, Processes and the Decision Tables group are selected by default after you check out a project. In the **Group Content** tab, business rules are displayed under their parent rule templates and decision tables are displayed under their parent virtual rule functions.

The TIBCO BusinessEvents WebStudio displays the project files in the Group Contents section either as a list or in a tree structure. In the Group Contents section click the project view icon to select the **Show items as tree** or **Show items as list** option to toggle between the tree view () and the list view (). The list view is the default view.

Updating (Synchronizing) a Project

After you log into the RMS server, and it is recommended that you update (synchronize) your local copy of the project, with the most recent changes from the RMS server, before continuing to work on it.


Other users may have checked in changes, additions, and deletions that are now available for checkout. There may be other changes too. You can select what updates to accept. Changes and additions to individual decision tables and business rules are listed.

Procedure

1. In the **Group Contents** section, select the artifact you need to synchronize.



The synchronize command does not support multiple artifact selection using Shift or Ctrl key. Alternatively, select a folder/project to select all artifacts under that folder/project, when project files are displayed in a tree structure.

2. Open the synchronize window using any of the following methods:
 - Click **WebStudio**, with the down pointing triangle in front of it, and select **RMS > Synchronize**.
 - Right-click on the selected artifact and then select **RMS > Synchronize**.
 - Click the synchronize icon  in the RMS toolbar.
3. If there are local changes as well as the server side changes, then WebStudio highlights the conflicted artifacts. Double click on the artifact to see the changes. Right click on the artifact and select **Use Local** to use your copy of the artifact, or select **Use Base** to use the copy on the server.
4. Select the checkboxes of the files you want to synchronize and click **OK**. Alternatively, select the header checkbox to synchronize all the listed project files and click **OK**.
If there are some local changes for the artifact and then you can double click the artifact to view the highlighted server side change. Hover the mouse over the highlighted (conflicted) field to see the server side value. Right click on the conflict field and select **Use Local** to use your copy of the artifact, or select **Use Base** to use the copy on the server and discard the local changes.

The selected artifacts are updated with the latest project files from RMS.

Synchronizing Artifacts Changes into RMS

In WebStudio you can synchronize the project artifacts changes, that you have made externally after performing check out in WebStudio, into RMS.

Using the **Sync To Repository** function, you can synchronize artifacts which cannot be managed in WebStudio. The project artifacts which can be synchronized are:

- Concepts
- Events
- Virtual rulefunction
- Rule template.

During synchronization, only additive changes are synchronized to existing WebStudio managed entities, which are referencing those changed artifacts. The changes related to deletion are not synchronized. After synchronization, you should use the **Validate** function to identify the entities which are affected by the deletion changes. The WebStudio managed entities are:

- Decision table
- Business rule
- Process.

Procedure

1. Open the synchronize window using any of the following methods:
 - Click **WebStudio**, with the down pointing triangle in front of it, and select **RMS > Sync To Repository**.

- In the **Group Contents** section, right-click anywhere and select **RMS > Sync To Repository**.
The Sync To Repository window is displayed.
- 2. In the Sync To Repository window, select the project that you want to synchronize from **Project List**. All the artifacts of the allowed types for the selected project are listed.
- 3. Select the artifacts that you want to sync to RMS and click **OK**.

Locking an Artifact

An exclusive lock ensures that while you are modifying the artifact, no other user is able to edit or delete it.

When other user tries to save the edit done to the locked artifact, or lock the locked artifact, WebStudio displays an error stating that the artifact is already locked by a user.

Prerequisites

Enable the artifact locking mechanism for WebStudio using the `ws.scs.locking.enable` property in the `RMS.cdd` file. See [Source Control System \(SCS\) Property Group](#) for more details.

Procedure

- In the **Group Contents** section, right-click on the artifact you need to lock, and select **RMS > Lock**. The artifact is locked and a lock is displayed on the artifact icon.

Unlocking a Locked Artifact

Release the locks on the artifact after completing work on the artifact, so that other users can also work on the artifact.

Alternatively, you can also select the **Auto UnLock on Approve/Reject** property check box to release the lock on the artifact as soon as the commit on the artifact is approved or rejected. The property can be set in the **Preference** portlet under the **Settings** tab in the WebStudio UI. See [Preferences Settings](#) for more details.

You can also set up a timeout for locks using the `ws.scs.lock.timeout` property in `RMS.cdd`, after which the locks are released. See [Source Control System \(SCS\) Property Group](#) for more details.

Procedure

- In the **Group Contents** section, right-click on the locked artifact you need to unlock, and select **RMS > UnLock**. The artifact is unlocked now and the lock disappears from the artifact icon.

Managing Locks on Artifacts



Administrators sometimes need to release locks, established by other users. For example, when other user is not available and the locked artifact is needed for some other updates.

Prerequisites

The user who is managing the lock should have the `ALLOW` permission set for the **manage_locks** action for the project in the RMS project's access control file. See the *TIBCO BusinessEvents® Administration* guide for more details on how to configure the access control file.

Procedure

1. Open the Manage Locks window using any of the following methods:

- Click **WebStudio**, with the down pointing triangle in front of it, and select **RMS > Manage Locks**  .
- In the **Group Contents** section, right-click any artifact and select **RMS > Manage Locks**  .
The Manage Locks window is displayed.
- 2. Select the project from the **Project List** dropdown.
The Manage Locks window displays all the locked artifacts by all users for the project.
- 3. Select the check boxes for the artifact for which you must release the lock and click **UnLock**.
All the selected artifacts are now unlocked and can be updated by all users.

Committing Artifacts for Approval

After you complete the additions, changes, or deletions to a project you checked out of RMS, commit the artifacts for approval. You can make a commit request for a selection of artifacts, or all.


Copies of submitted artifacts are saved in cache, with the names of the users who submitted them, version numbers, and the status `Committed`. A task is created for the Approver role to review the check-in requests. (see [Working with the Worklist Items](#)).

A Revision ID is assigned to each commit request. This ID is used in other dialogs to allow you to view the details on the request.

If the project is checked out from the SCS repository, then, similar to the file mode, the committed artifacts are saved in the local cache, till the approval. After the approval, the committed artifacts from the cache are saved to the local repository, identified by the value of the `ws.scs.repo.dir` property in `RMS.cdd` (see [Configuring RMS Server Properties](#)), and checked in to the SVN repository as well. The comments entered for commit are used as checkin comments for the SVN repository, after approval.

Procedure

1. In the **Group Contents** section, select the artifact you need to commit.

Commit command does not support multiple artifact selection using the Shift or Ctrl key. Alternatively, select a folder/project to select all the artifacts under that folder/project, when the project files are displayed in a tree structure.
2. Open the commit window using any of the following methods:
 - Click **WebStudio**, with the down pointing triangle in front of it, and select **RMS > Commit** .
 - Right-click on the selected artifact and select **RMS > Commit** .
 - Click the commit icon  in the RMS toolbar.
3. In the Commit window enter the revision description in the **Comments** text box.
Comments field is not a mandatory field but it is recommended to comments before committing. This helps the approver to understand the nature of changes.
4. Click the checkboxes for the files you want to commit to the RMS project and click **OK**.
Alternatively, click the header checkbox to select all the listed project files.
5. Click **Finish** after the system displays the checkin successful message with the revision ID.

Reverting Artifact's Update

The Revert command can revert the updates done to an artifact in the WebStudio and restore it to a previous version.

The Revert command also reverses the Delete command. It can reverse the update to the following artifacts:

- Decision Tables
- Business Rules
- Processes
- Domain Models

The Revert command restores the artifact to the last version that was committed in the repository. If multiple changes are made to the artifact and no commit is done to the repository, the Revert command reverses all the changes. This command does not work on the changes which are committed to the repository. Thus, ensure that the updates, which you are sure of, are committed to the repository on a regular basis. Also ensure that the Revert command is used only on the updates, of which you are unsure.

You can make a revert request for one or for a selection of artifacts.

Procedure

1. In BusinessEvents WebStudio, open the Revert window using one of the following ways:
 - Click **WebStudio**, with the down pointing triangle in front of it, and select **RMS > Revert**.
 - Right-click the selected artifact and select **RMS > Revert**.
 - Click the **Revert** icon in the RMS toolbar.
2. Select the check boxes for the artifacts for which you want to revert the updates and click **OK**. Alternatively, select the header check box to select all the listed artifacts.
3. Click **Finish** after the system displays the "revert successful" message with the revision ID.

Result

The artifact is reverted to its previous version, which was last committed to the repository.

Viewing History of an Artifact

You could view the update performed in earlier revisions of the artifact, and also view the update performed to other artifacts in the those revisions.

Procedure

1. In the **Group Contents** section, right-click the artifact, of which you want to view revision history, and select **RMS > Show History**.
2. The **History** window displays all the previous revisions of the artifact. Use the following search options and click **Go** to filter the results:

Option	Description
From	Select the from date using the date picker. Only revisions which are committed on or after this date are displayed.

Option	Description
To	Select the to date using the date picker. Only revisions which are committed on or before this date are displayed.
Search	Type the keyword to perform search in the user-name and comments fields .

3. In the **Revisions** panel, select the revision of which you want to view the details. See [Revisions Panel Reference](#) for more details on the Revisions panel columns.

A detailed list of all updated artifacts for the selected revision is displayed in the Details panel with the artifact of interest in bold. See [Details Panel Reference](#) for more details on the Details panel columns.

4. In the **Details** panel, right-click the artifact and select **Diff with Previous Version** to view the update made to the artifact in the selected revision from the previous approved revision.

The updates on the updated artifact are displayed highlighted in different colors for addition, deletion, and modification. Refer to the legend displayed on the editor for the significance of each highlighted color. Hover the mouse pointer over the highlighted (modified) part to see the previous value.

The History window is minimized while the version difference for the artifact is displayed. Click the **Restore Window** icon of the minimized History window to restore it.

5. Finally, click **Close** to close the History window.

History Window Reference

The history window lists the revisions for the selected artifact in the Revision panel and details of the selected revision in the Details panel.

Revisions Panel Reference

The revision panel lists revisions associated with the selected artifact. The list can be filtered using the search options in the History window.

Column	Description
Revision	The unique ID assigned to each revision by WebStudio.
User Name	The user name of the user who had performed the commit.
Date	The RMS system date and time of the commit. Format of Date is <i>DD/MM/YYYY HH:MM</i> . For example, 06/08/2013 14:35.
Comments	The comments provided by the user at the time of the commit.

Details Panel Reference

The Details panel displays the list of artifacts committed in the selected revision. The panel also lists various details about the artifacts update specific to that revision. You can also view the differences between the selected revision of the artifact and its previous approved version.

Column	Description
Action	<p>The update action performed on the artifact.</p> <p>The values are:</p> <ul style="list-style-type: none"> • Create • Modify • Delete
Type	The icon identifying the type of artifact, whether business rule or decision table.
Path	The location of the artifact with respect to the project's root directory.
File type	<p>The file extension of the artifact.</p> <p>The values are:</p> <ul style="list-style-type: none"> • ruletemplateinstance (for Business Rules) • rulefunctionimpl (for Decision Tables)
Approval Status	<p>The approval status of the artifact for the revision. See Approval Status Values for more details on approval status values.</p> <p>The values are:</p> <ul style="list-style-type: none"> • Committed • Approve • Reject • BuildAndDeploy

Removing a Project from Workspace

You can remove the project from the WebStudio workspace after completing work on the project.

The project could be removed either from the **My Workspace** tab, where all its artifacts are displayed, or from the **Dashboard** tab, where all open projects are listed under the My Projects dashboard portlet.

Removing a Project from Workspace Using the My Workspace Tab

The **My Workspace** tab displays the checked-out project resources.

Procedure

1. In the **My Workspace** tab, under the **Group Content** section, click the **Change View** icon and select **Show items as a tree** (if not already selected).

The project artifacts are now displayed in the tree structure.

2. Right-click the project name and select **Delete** or click on the **Delete** icon in the toolbar.

The Confirm window is displayed to confirm project deletion.

3. Click **OK** to remove the project from the WebStudio workspace.

Removing a Project from Workspace Using the Dashboard Tab

You can remove the project from the workspace using the **Dashboard** tab.

Procedure

1. In the **Dashboard** tab, under the **My Projects** dashboard portlet, click the **Delete** icon in front of the project name.

The Confirm window is displayed to confirm project deletion.

2. Click **OK** to remove the project from the WebStudio workspace.

Validating Multiple Artifacts of a Project

You can save time and effort by validating multiple decision tables and business rules of an RMS project at the project level.


In TIBCO BusinessEvents WebStudio, validate the artifacts, for any access control violations or syntax errors, using the validate command.

Procedure

1. In the **My Workspace** tab, under the **Group Content** section, click the **Change View** icon and select

Show items as a tree  (if not already selected).

The project artifacts are now displayed in the tree structure.

2. Select the project and click the **Validate** icon  to validate multiple decision tables and business rules of the project.

All the errors are displayed in the **Problems View** tab at the bottom of the application.

3. Double-click errors to see the problematic view.

Take any needed corrective actions and then validate the artifacts again until all errors are resolved.

Deleting Errors and Warnings

After validation, when there are many errors and warnings in the **Problems View** tab, delete obsolete warnings and errors, so that you can browse the errors easily.

Procedure

- In the **Problems View** tab, right-click the obsolete warning or errors and select **Delete**.

The selected error or warning is now deleted from the tab.

Keyboard Shortcuts

Keyboard shortcuts can make it easier to interact with TIBCO BusinessEvents WebStudio, saving you time and effort as you work with RMS projects in WebStudio.

Keyboard Shortcuts in WebStudio

Command	Keyboard Shortcut
Shift+Alt+K	Checkout
Shift+Alt+P	Commit
Shift+Alt+R	Revert
Shift+Alt+W	WorkList
Shift+Alt+G	Generate Deployable
Shift+Alt+J	Validate
Shift+Alt+O	Export
Shift+Alt+U	Import
Shift+Alt+L	Sync

Decision Table

A decision table row represents a business rule. It has one or more conditions and one or more actions.

Each condition cell is equivalent to one condition (one line) in the TIBCO BusinessEvents rule editor condition area. Similarly each action cell is equivalent to one action (one line) in the TIBCO BusinessEvents rule editor action area (that is, one rule). Decision table rules are like business rules. The rule that calls the virtual rule function that implements the decision table participates in inferencing in the usual way. When the virtual rule function is called, the decision table rules are applied.

Conditions and Actions

The columns of a decision table are made up of condition columns on the left, and action columns on the right. Each column represents one condition or one action.

A condition is a test that must evaluate to true before the action is executed. If a decision table rule uses multiple conditions, all the conditions for a row must evaluate to true for the action to execute.

In each row (rule) you define the specific conditions and actions. For example, if a condition column is Age (using a concept property of that name), then each row can define a different age range. The action for each row would define what action to take if a given concept instance contains an age property within the specified range.

If you add a second condition column called Income, then before the action is taken, a concept instance would be tested to see if both the age and the income are within the ranges specified in the rule's conditions.



Conditions that are blank, contain an asterisk, or are disabled are ignored, and are treated as if they evaluate to true.

Regular and Custom Conditions and Actions

Regular condition

It is the value of an entity specified in the virtual rule function scope, or a simple comparison with the value - is greater than, is less than, is greater than or equal to, is less than or equal to.

Custom condition

It can use the rule language, standard functions, and data in the scope of the function at runtime (for example, scorecards and global variables). It can contain complex formulas.

Regular action

It sets the value of an entity specified in the VRF scope.

Custom action

The action can use the rule language, standard functions, and data in the scope of the function at runtime to do whatever is desired. For example, the action could be to send an event out to a different system for follow-up.

Using non-literal values in a regular condition or regular action

Suppose event A and event B are in scope, and event A has property ZZ, and event B has property YY. Both properties belong to the same data type. Suppose you then drag property ZZ to a condition or action column. In the cell, you can then specify a value as `b.YY`. The effect is different depending on the type of column:

- In a condition column, this means: compare the value of property ZZ with the value of property YY.
- In an action column this means: set the property ZZ to the value of YY.





Decision tables are available in TIBCO BusinessEvents WebStudio in the view or edit mode only if you have the TIBCO BusinessEvents Decision Manager add-on installed on your system.

Creating a Decision Table

A decision table can only be created using a virtual rule function (VRF).

For instructions on adding a VRF, see *TIBCO BusinessEvents Developer's Guide*.

Procedure


1. Select the **Projects** group.
2. In the **Group Contents** section, right click the virtual rule function (VRF) you want to use and click **New Decision Table**.
3. Enter the decision table name in the Decision Table pop-up window and click **OK**.
If needed, you can enter a new folder name in the **Parent Folder** field and the new folder is created in the project.
The new empty decision table is displayed in the work area.

Deleting a Decision Table

You can remove a newly added table as well as any existing table from the project.

If the table you want to delete has been checked into the RMS project, then you must submit the deletion for approval. After approval, the deleted table is removed from the RMS project.

Procedure

1. Do one of the following:
 - In **Group Contents**, right-click the decision table name and select **Delete**.
 - In **Group Contents**, select the decision table name. Then click the **Delete** icon  in the **Edit** toolbar.
2. If the table has been committed to the RMS project, select one of the following options as appropriate:
 - Commit the project or the parent folder so that the RMS master copy of the table is also deleted.
 - Synchronize the project to replace the deleted copy with the RMS project version of the table.

What to do next

In order to delete the decision table classes, after approving the deletion of decision table, select the **BuildAndDeploy** status of worklist for the deleted decision table. See [Generating and Deploying the Decision Table's Class File](#) for more details on how to generate decision table classes.

Renaming a Decision Table

You can rename a decision table in WebStudio to a more suited name, which identifies its functionality more appropriately.

Procedure

1. In **Group Contents**, right-click the decision table name and select **Rename**.



A decision table name can contain letters, numbers, and underscore only. It should start with a letter and be less than 64 characters.

WebStudio displays the **Rename** completely successful message.


2. If the table has been committed to the RMS project, commit the project or the parent folder so that the RMS master copy of the table is also renamed.

The changes appear in two entries: one with the **Change Type** as **Added** and other with the **Change Type** as **Removed** in the Commit window and the Worklist window. Commit both the entries together as a single commit; otherwise WebStudio displays an error.

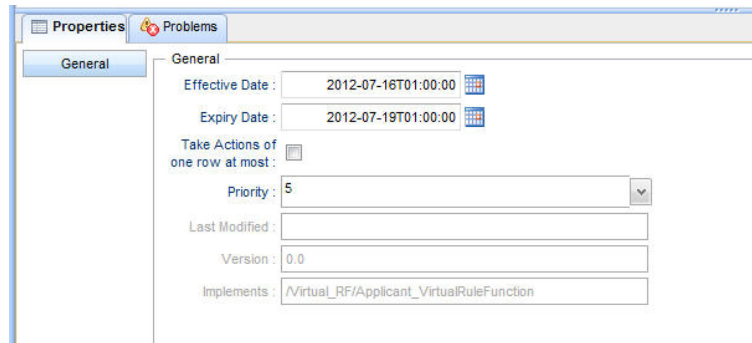
Setting the Decision Table Properties

In the **Group Contents** section, double-click the decision table you want to open. You can now edit, analyze, and export the open decision table.

Procedure

1. In the decision table work area, click the **Properties** icon .

The Properties view is displayed at the bottom of the work area.



2. Edit the following fields to set the Decision table properties.

Option	Description
Effective Date	Define the date and time on which the decision table becomes valid in the runtime application. Click the Calendar icon and use the Calendar dialog to set the date, and then edit the time as needed.
Expiry Date	Define the date and time after which the decision table is no longer valid in the runtime application.
Take Actions of One Row at Most	Select the check box, if you want the decision table to stop after one row (rule) passes the condition tests. Only the actions of that row are considered (until the next time the decision table's rule function is called).
Priority	Set a priority as desired. When a VRF has multiple implementations (decision tables), the decision table's Priority setting determines the order in which the decision tables executes.

3. Click the **Save** icon  in the Edit toolbar to save the changes to the decision table.

Result

The decision table is still open for you to edit and the changes are not applied to RMS unless you commit the changes to RMS.

Enabling or Disabling the Page View

When a large table is displayed in multiple pages, it improves the server response time as well as a better user experience.

A decision table is displayed in pages if the number of rows are more than the value specified for the page size (minimum is 10 and default is 20). The page size can be specified in the following locations:

- The `tibco.clientVar.Webstudio/DecisionTable/pageSize` property in the RMS.cdd file. See [Configuring RMS Server Properties](#) on how to configure the RMS properties.
- The **Decision Table Page Size** field in the Preference portlet under the **Settings** tab in the WebStudio UI. See [Preferences Settings](#) for more details.

Procedure

1. In the decision table editor, click **Show All** to disable the page view for the decision table and display all rows.
2. Click **Show Pages** to return to the page view for the decision table.

The option appears in place of navigation options after clicking the **Show All** option.

Result

Navigate through the pages using navigation options present at the bottom of the decision table. Click on the specific page number to open that page or click **Previous** and **Next** options to navigate through pages sequentially.




Analyzing a Decision Table

In the Table Analyzer you can set example condition values and perform various validation checks on the currently displayed decision table.



Table analyzer supports the Not Equal To (\neq) operator.

Procedure

1. In BusinessEvents WebStudio open the decision table you want to analyze.
2. Click the **Decision Table Analyzer** icon  to view Table Analyzer.
3. In the Table Analyzer section, controls are created dynamically for setting values and ranges for each condition in the table. Enter or select the example values for which you want to analyze the table.
For example, if the condition is a range, for instance “< 40” or “> 10 && < 100”, then the corresponding control in the Table Analyzer section is a Slider that spans the range from the minimum value to the maximum value. You can then select a range of values within the range.
4. Click the **Analyze** icon  to analyze the decision table based on the selected values.
A pop-up window displays the count of any issues. The Problems section at the bottom displays more information on issues. Click the issue to highlight the row.
5. You can click the **Show Coverage** icon  in the table analyzer section to highlight all rows that meet the criteria. In page view, browse through the pages to see the highlighted rows.
For example, if you select Male for Gender; true for Eligible; and 50000–60000 for Income, clicking **Show Coverage** displays all rows that have Male, true, and an income between 50000 and 60000.

Validating a Decision Table


In TIBCO BusinessEvents WebStudio, validate a decision table, for any access control violations or syntax errors in the table, using the validate command.



The Validate icon is activated when the decision table is opened in the editor. EAR file for the project should be present in the deploy location; otherwise, the validate command fails. If the EAR file is not present at the deploy location, then you receive the following error while validating the decision table:

EAR for projects: <project_name> either not present or not in sync at deploy location.

Procedure

1. Select the decision table and click the **Validate** icon  to validate the table.
All the errors are displayed in the **Problems View** tab at the bottom of the application. New errors in syntax are added to these existing errors.

2. Double-click errors to see the problematic view.

Take any needed corrective actions and then validate the table again until all errors are resolved.

Validating a Decision Table Using Test Data

Using test data you can verify if the decision table covers all the data that might be passed through the decision table. This enables you to add more rules to the decision table, if the decision table is not able to cover the test data.

Prerequisites

Create the test data for at least one argument from BusinessEvents Studio. See *TIBCO BusinessEvents Developer's Guide* for more information on creation of test data. The test data must be in the project in your repository.



The test data are not managed in the RMS repository.

Procedure

1. In the **Table Analyzer**, click the **Check Test Data Coverage** icon .



You can select the test data for only a single argument at a time.

The Check Test Data Coverage window is displayed with the available test data.


2. Select the test data, for which you want to validate the decision table and click **OK**.
The table analyzer validates the decision table based on the test data and displays the **Test Data Coverage** tab at the bottom pane. The Test Data Coverage pane highlights the rows of test data, which are covered in the decision table.

Exporting a Decision Table

You can export decision tables from TIBCO BusinessEvents WebStudio to Microsoft Excel spreadsheets.

Procedure

1. Open the decision table you want to export.
2. Execute the export command in either of the following ways:

- Click the **Export** icon  in the **Tools** toolbar. You see the Save As dialog box for the exported file.
 - Right click the decision table in the Group Contents section and select **Export**.
- Browse to the directory in which you want to save the excel file and provide a file name.
 - Click **Save**.

For example, a sample exported Microsoft Excel Sheet is given the following figure.

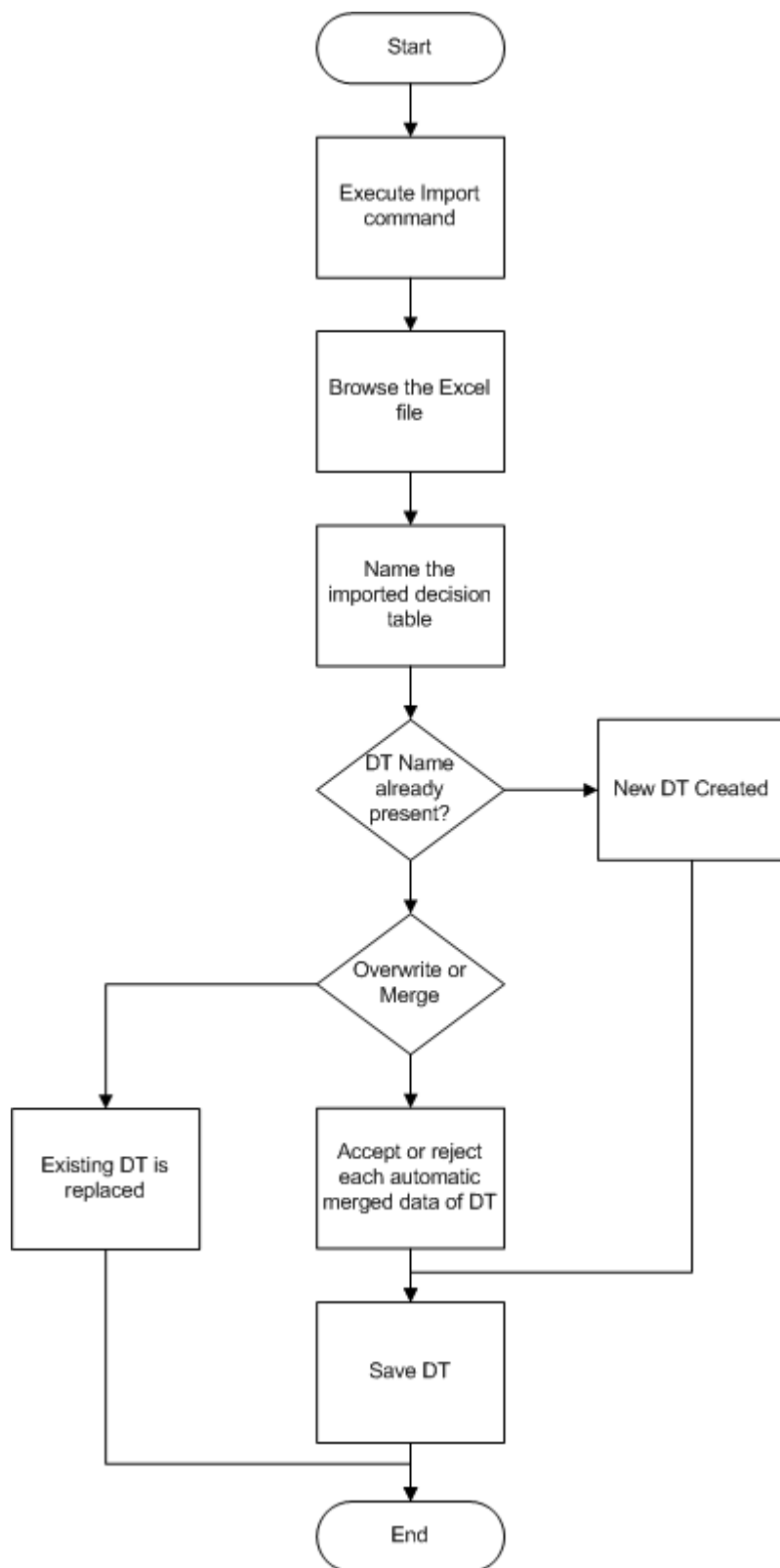
A1		Version									
	A	B	C	D	E	F	G	H	I	J	
1	Version	2									
2											
3	Declarations										
4											
5	Path	Alias	Property								
6	/Concepts/applicant	BOTH									
7	/Events/Applicant	BOTH									
8											
9	DecisionTable										
10											
11	Id	Condition	Condition	CustomAction	Action (5)	Action (11)	Action (9)	Description	Priority		
12	applicant	applicant	Custom A	applicant	applicant	applicant	applicant	Status			
13	1	"John Doe" >=20 && < System.de true			2500			"VISA Granted"			
14	2	"Sandra W" >=20 && < System.de false			0			"Loan Rejected"			
15	3	"Prakash" >=20 && < System.de true			7500			"Pending"			
16	4	"Jane Doe" > 30	System.de true		25000			"Platinum Status"			
17											
18	ExceptionTable										
19											
20	Id	Description	Priority								
21											
22											

Importing a Decision Table


You can import correctly formatted Microsoft Excel file to create a decision table or to merge to an existing decision table in TIBCO BusinessEvents WebStudio.



- Only import from the Microsoft Excel files is supported. You cannot import from CSV (comma-separated values) files or tab-delimited text files. You can open such files in Excel and save as Excel format binary files.
- Before importing the Microsoft Excel file, ensure that cells are not protected (locked). The protected cells, after import, appear disabled in WebStudio.

Importing a Decision Table Flowchart**Procedure**

1. Open the Import dialog box in either of the following ways:

- Select the virtual rule function and click the **Import** icon  in the **Tools** toolbar.
 - Right click the virtual rule function in the Group Contents section and select **Import**.
2. Click **Choose Files** in the Import Decision Table dialog box.
 3. Browse the excel file you want to import and click **Open**.
 4. In the **Name** field, do either of the following:
 - Specify the new decision table name and click **OK**.
 - Specify the name of an existing decision table name and click **OK**.

Depending on the decision table name, either a new decision table is created or a prompt is displayed that a decision table with the existing name already exists. The prompt provides the option to either overwrite existing table or merge the imported and the existing table.

5. If you have given an existing decision table name, in the prompt, select either of the following options and click **OK**.

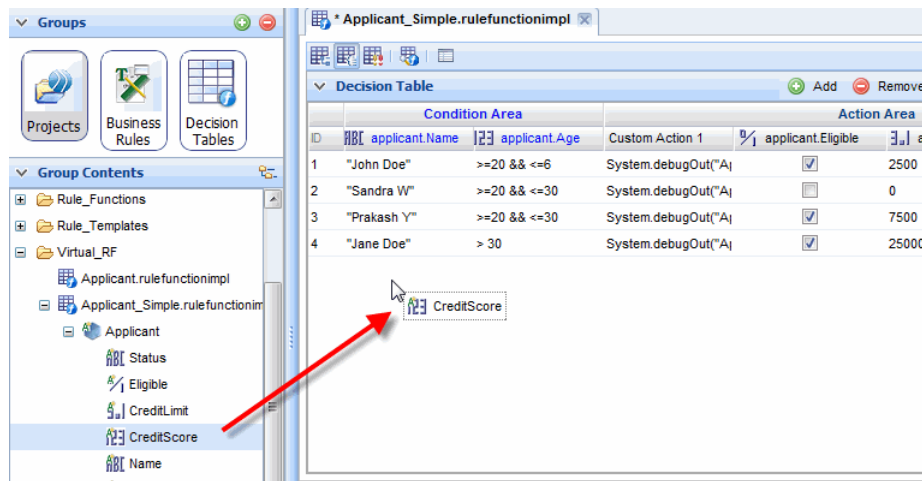
Option	Description
Overwrite the existing decision table	The imported table overwrites the existing decision table with the same name.
Manually merge the imported decision table with the existing one, in a diff editor	<p>WebStudio automatically merges both the decision tables and displays the merged decision table in the decision table editor. All the differences are highlighted using the following legends:</p> <ul style="list-style-type: none"> • Green for added items • Red for deleted items • Yellow for modified items <p>In the editor, to restore any data to the local value, click the modified item and select Use Local. Click the Save icon to save the changes to the decision table.</p>

Adding Condition and Action Columns from Arguments

Add the condition and action columns using the arguments under the virtual rule function scope.

Procedure

1. Open the decision table you want to edit.
2. In the **Group Contents** section, right click the decision table and select **Show Arguments**.
A plus icon is displayed in front of the decision table for argument list expansion.
3. Click the plus icon to expand the arguments list. If needed, click the plus icons for concepts and rules as well.
The decision table arguments are listed under the decision table.
4. Drag a property from the argument list to the decision table editor.
A Create Column pop-up window appears to create the condition or action column.



- Click the **Condition** or **Action** radio button and click **OK** to create a Condition or Action column. A new Condition or Action column is added at the end of the Condition or Action column-sets respectively.

Adding Custom Condition and Action Columns

Add the condition and action columns using the rule language, standard functions, and data in the scope of the function at runtime (for example, scorecards and global variables).

Procedure

- Open the decision table you want to edit.
- Click **Custom** in the decision table editor menu and select either of the following options:
 - Select **Add a custom Condition** to insert a custom condition column at the end of condition column set.
 - Select **Add a custom Action** to insert a custom action column at the end of action column set.

Defining Custom Columns Using Substitution

You can specify complex entries in a column by using column substitution. This feature works where all the entries in a column follow a specific pattern.

As an example, suppose all entries in a column use a comparison such as this, but with different values in each case:

```
Math.maxInt(MyAlias.Value, MyAlias.MaxValue) < 50
```

Instead of typing such entries in each cell, you could use substitution.

Procedure

- In the column header, specify (right-click the column and select **Field Settings**) the pattern:
`Math.maxInt({0}, {1}) < {2}`
- In each of the column cells, specify parameter values, using a semicolon delimiter:
`MyAlias.Value; MyAlias.MaxValue; 50`
- Press Enter so that the cell contents are substituted with the pattern of the column name.

Setting Default Value in Decision Table Column

Using the **Field Settings** option of the column, you can set a default expression for the column for all new rules.

Procedure

1. Open the column's options menu by either of the following ways:
 - Right-click any one of the column headers.
 - Hover the mouse pointer over any one of the column headers to view the down-pointing triangle and click the triangle.
2. Select the **Field Settings** option.
The Field Settings dialog is displayed.
3. Enter the default value for the column in the **Default Expr** text box.
This default value is used for the column, whenever you add a new row.
The **Include existing rules** check box is activated.
4. Select the **Include existing rules** check box to fill all the blank cells for the column with the default value specified in the **Default Expr** text box.
5. Click **OK**.
The Field Settings dialog closes.

Result

Now the value of the **Default Expr** field is used as the default value for the column for new rules. If you have selected the **Include existing rules** check box, all empty cells for the column in the decision table are replaced by the default value.

Viewing or Hiding Columns in the Decision Table

To avoid scrolling tables with large number of columns, you can view only the columns that you want in the decision table and hide all others.

Procedure

1. Open the column's options menu by either of the following ways:
 - Right-click any one of the column headers.
 - Hover the mouse pointer to any one of the column headers to view the down-pointing triangle and click the triangle.
2. Select the **Columns** option and select or clear the column that you want to hide or view.
The check marks in front of the column names identify whether the column is visible or hidden. You can select or clear the entire condition and action columns by selecting or clearing the **Conditions** and **Actions** option in the menu.

Resizing a Column

To view the conditions and actions value better, you can resize the column of a decision table.

Procedure

- Perform any of the following actions to resize a column in the TIBCO BusinessEvents WebStudio:
 - Drag the right boundary of a column to manually resize them.
 - Right click on the column header you want to resize and click **Auto Fit** to resize the column to fit to the content size.
 - Hover the mouse pointer to the column header to view the down-pointing triangle. Click the triangle and click **Auto Fit** to resize the column to fit to the content size.
 - Double click the right boundary of a column to make it auto fit to the content size.
 - Right click on the column header you want to resize and click **Auto Fit All Columns** to resize all columns to fit their content size
 - Hover the mouse pointer to the column header to view the down-pointing triangle. Click the triangle and click **Auto Fit All Columns** to resize all columns to fit their content size.

You now have the ability to expand and collapse columns. No more scrolling is required if you want to see selected columns from the decision table which has a large number of columns. Right clicking the decision table columns will show you the "Columns" option to either hide or show these columns.

Removing a Column

Remove a column when you need to remove the condition or action for the entire decision table.



You cannot remove the last condition column because a minimum of one condition column must exist if you have any action columns.

Procedure

- Right click the column header and click **Remove** to remove the column from the decision table.

Sorting the Decision Table

The rows of a decision table can be viewed in a ascending or descending order of the selected column.

In page view, the sorting is performed only for the page and not the entire decision table. If you wish to perform the sorting for the entire decision table, then first click **Show All** to disable the page view and then perform the sorting.



Sort options are not available for columns with boolean entry values and for custom condition and action columns.

Procedure

- There are more than one way to sort the rows of a decision table in the BusinessEvents WebStudio.
 - Click once on the column header to sort the table in ascending order of the column and click again on the column header to re-sort the table in descending order of the column.
 - Hover the mouse pointer over the column header to view the down-pointing triangle. Click the triangle and select **Sort Ascending** or **Sort Descending** to sort the table in ascending or descending order of the column.

- Right click on the column header you want use for sorting and select **Sort Ascending** or **Sort Descending** to sort the table in ascending or descending order of the column.

Setting Row Priorities



Each row in a decision table represents a separate business rule. You can control the order in which sets of rows are executed using the row priority setting.

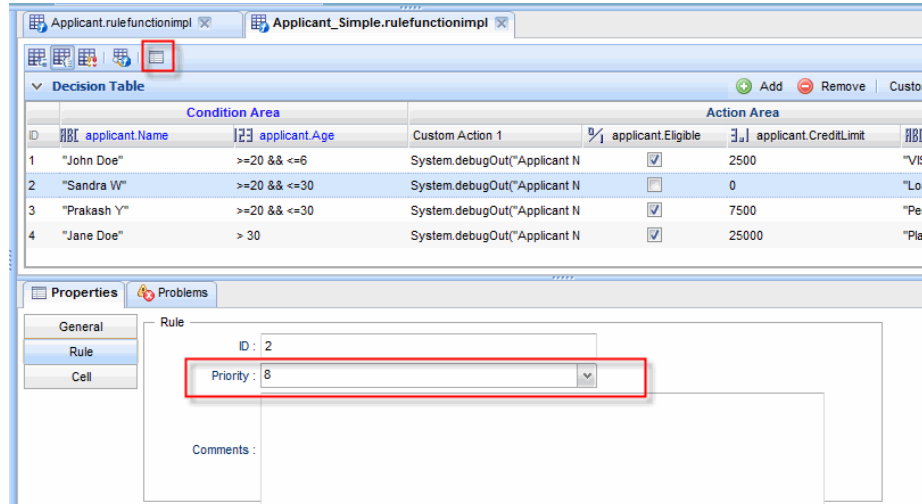
Rows with higher priorities are executed before those with lower priorities, as follows:

- First, all conditions are checked for all rows that have the highest priority. The checking order within a set of rows with the same priority is not determinate.
- Then the rule actions for all of those rows whose conditions evaluate to true are executed. The execution order is not determinate. The runtime engine optimizes rule execution.)
- The process is repeated for all rows with the next highest priority, and so on.

Ten is the lowest priority and one is the highest. Five is the default priority.

Procedure

1. Select the row whose priority you want to set.
2. Click the **Properties** icon  to display the Properties view.
3. In the **Properties** view, and select the **Rule** side-tab.
4. In the **Priority** field, select the appropriate value.
5. Click the **Save** icon  in the **Edit** toolbar to save the changes to the decision table.



Filtering the Rows

Apply filters to focus on certain rows instead of entire table.

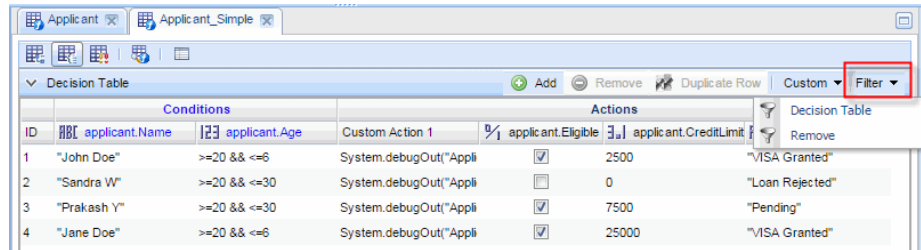
Apply filters on multiple columns for more refined results. In TIBCO BusinessEvents WebStudio decision table filter makes only textual comparison between the row values and filter values.



If you are viewing the decision table in page view, you can only filter values on the displayed page and not the complete decision table. If you wish to use the filter for the entire decision table, then first click **Show All** to disable the page view and then use the filter.

Procedure

1. To apply the filter, click **Filter > Decision Table** . Enter the values in filter cells (above column header) to filter the table.
2. To remove the filter, click **Filter > Remove** in the decision table editor.



Creating Duplicate Rows

You might want to create Duplicate rows, when you have only minor condition or action of the existing rows to create a new rule, instead of creating the entire row from start.



When you duplicate rows in the page view, the duplicate rows are still added to the end of the decision table, but are shown on the same page as original rows. The table is reorganized when it is closed and reopened, thus the duplicate rows are then be displayed at the end of the decision table.

Procedure

1. In the Decision Table editor, select the Decision Table rows that you want to duplicate and click **Duplicate Row**.
New rows are added, at the end of decision table, with the same conditions and actions of selected rows, but with a different ID.
2. Edit the rows as needed.

Removing a Row

Remove the row if you want to delete the rule entirely instead of editing it. You can remove only one row at time.

Procedure

1. In the decision table editor, select the rows you need to remove and click **Remove**.
2. In the Remove Decision Table Record window, click **OK** to confirm the removal of the row.

Supported Operators

TIBCO BusinessEvents WebStudio supports several operators for Decision Tables that you can use in the expressions.

Named or Scoped Conditions must Evaluate to Boolean

For logical operators, named conditions will have to use them explicitly from the second operand onwards.

For example, if the LHS of the condition is the column name, bankuser . Age, the following are valid entries:

```
50 || ==60 , 20 || >=80
```

Custom Conditions must evaluate to Boolean

Same as above, except that you must specify the entire Boolean expression in the cell, including LHS.

Actions or Custom Actions

Named actions perform assignment of the cell expression to the column name property.

Custom actions are treated as normal strings and can be anything that you would enter in the THEN section of rules (LHS included).

Supported Operators

Operator	Description
==	Equals to
!=	Does not equal to
>	Greater than
>=	Greater than or equals to
<	Less than
<=	Less than or equals to
	Logical OR
&&	Logical AND
+	Addition or String concatenation (depending on context)
!	Logical NOT (only with boolean)
-	Subtraction
*	“Don’t Care” (that is, ignore), or multiplication (depending on context)
/	Division
=	Assignment
.	Scope resolution
()	Operator precedence order or function call (depending on context)
[]	Array declaration or array indexing (depending on context)
{ }	Block resolution or array Initialization (depending on context)
?:	Ternary conditional
++	Increment (used only with custom actions)

Operator	Description
--	Decrement (used only with custom actions)

Editing Domain Models

Using RMS, you can manage the lifecycle of domain models, and edit them in TIBCO BusinessEvents WebStudio.

You can create a domain model only in BusinessEvents Studio; however, you can edit them in Studio as well as WebStudio. See [WebStudio Domain Model Editor Reference](#) for more details on the domain model editor fields. The updates are committed in the RMS and are available to all user after the approval.

See *TIBCO BusinessEvents Developer's Guide* for more information on domain models.

Prerequisites

Ensure that **Project** is selected under the **Groups** section in WebStudio.

Procedure

1. Double click the domain model you want to edit. Edit the domain model as required in the editor. The domain model editor is displayed.
2. Edit the domain model as required in the editor.
3. Click the **Save** icon to save the updates.
4. Click the **Commit** icon to commit the changes to RMS for approval. Once approved all users can see the updates after synchronization.


WebStudio Domain Model Editor Reference

In WebStudio, using the domain model editor you can edit the domain model entries. You can also configure the domain model to inherit entries from another domain model in the project.

Configuration

In the Configuration section, you can configure the basic domain model information.

WebStudio Domain Model Editor Configuration Properties

Field	Description
Description	Description of the domain model.
Domain Type	<p>The data type of the domain model. The values are:</p> <ul style="list-style-type: none"> • String • Int • Long • Double • Boolean • DateTime <div>  <p>You can edit the Domain Type field only if there are no domain entries in the domain model.</p> </div>

Field	Description
Inherits From	Specifies the domain model from which the domain entries are inherited. Click Browse to select another domain model in the project from where you can inherit the domain entries.

Domain Entries

The Domain Entries section lists all the domain entries, that you can edit.

WebStudio Domain Model Editor Domain Entries Commands

Commands	Description
Add	Add a new domain entry to the domain model. The new entry is appended at the end of the domain entries list.
Remove	Remove the selected domain entry from the domain model.
Duplicate	Create a duplicate of the selected domain entry at the of the domain entries list.

WebStudio Domain Model Editor Domain Entries Columns

Column Name	Description
Description	Name of the domain entry.
Value	Displays the value of the domain entry. Value can be edited in the Details section.

Details

In the Details section you can enter the value of the domain entry based on the **Domain Type**.

WebStudio Domain Model Editor Details

Data Type	Description
String	String entries are simple text strings.
Integer, Double, Long	In a single domain model, you can enter single values, range values, or a mixture of both. Acceptable values for integer, long, and double domain entries are the same as for the equivalent Java datatypes.

Data Type	Description
Boolean	Boolean entry values are always <code>true</code> or <code>false</code> . The description can give the meaning of the pair of choices, such as male or female, supported or unsupported, eligible or ineligible and so on.
DateTime	In a single domain model, you can enter single values, range values, or a mixture of both. You specify a date and a time. Using the date picker you can select the date.

Business Rules

Business Rules artifact allows non-technical business users to build complex business rules in TIBCO BusinessEvents WebStudio.

Adding Business Rules

Business Rules are created using the Rule templates developed in TIBCO BusinessEvents Studio. If a rule template view is associated for the rule template, the business rule has an easy-to-use form based interface to specify conditions and actions; otherwise, conditions and actions are added using the builder.

See *TIBCO BusinessEvents Developer's Guide* for overview on rule template, and rule template view.


A business rule in a project is identified by the `.ruletemplateinstance` file extension. The rule template instance file is generated when the BuildAndDeploy option is selected for the business rule in approvals. See [Generating and Deploying the Business Rule's Rule Template Instance](#) for more details.

Adding a Business Rule Using the HTML Form

If a rule template view is defined for the rule template, then you can provide values for the bindings, defined in TIBCO BusinessEvents Studio, in a HTML form.

The form interface is the visual presentation of the rule template defined by rule template views. See *TIBCO BusinessEvents Developer's Guide* for more details on rule template views.

Procedure

1. In TIBCO BusinessEvent WebStudio, under the **Group Contents** section, right click the rule template you want to use and select **Create Business Rule**.
2. Type the business rule name in the dialogue box and click **OK**.
If needed, you can enter a new folder name in the **Parent Folder** field and the new folder is created in the project.
If the rule template view is defined for the rule template, then the HTML form is presented in the business rule editor.
3. Enter the threshold values in the input fields of the HTML form.
4. Click the **Save** icon  to save updates to the business rule.

Adding a Business Rule Using the Builder Interface

If the rule template view is not defined for the rule template, then you can build conditions and actions using various operators in the builder style of user interface .

The builder for business rule in the TIBCO BusinessEvents WebStudio consists of two sections: When section (condition builder) and Then section (command builder).

Creating a New Business Rule

A new business rule needs to be created from the rule template, without the rule template view, to add conditions and actions to it.

Procedure

1. In TIBCO BusinessEvent WebStudio, under the **Group Contents** section, right click the rule template you want to use and select **Create Business Rule**.

2. Type the business rule name in the dialogue box and click **OK**.

If needed, you can enter a new folder name in the **Parent Folder** field and the new folder is created in the project.

If the view is not associated with the rule template, a builder interface opens up in the editor.

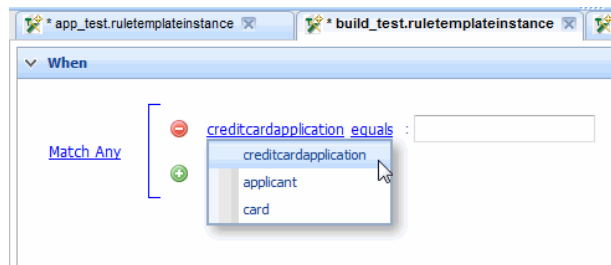
Adding a Conditional Clause Using the Condition Builder

The When section lets the user define additional conditions that must be met, in addition to the pre-condition defined in the Rule Template.

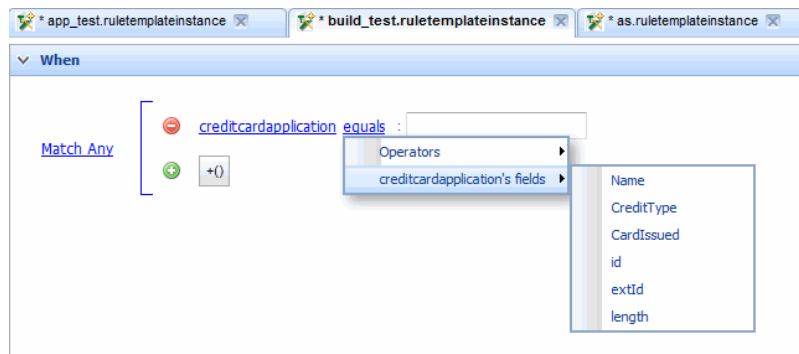
The builder determines the scope of the rule template, and allows the user to define individual conditional clauses based on that scope. Each conditional clause has one or more sub-clause, and a match operator determines whether the conditional clause is true when all sub-clause are true, or any one sub-clause is true, or none of the sub-clauses evaluate to true.

Procedure

1. In the business rule editor, under the **When** section, click the artifact-name link and select the artifact you require.




2. Click the operator-name link to select another operator, or a child artifact of the previously selected artifact, as you require. See [Condition Builder Operators](#) for list of supported operators.



3. Enter the conditional value for the artifact in the empty box.
4. Click the **Add** icon to another condition to the conditional clause, or click the **Remove** icon to remove the existing condition from the conditional clause, if you want.
5. Click the **Add Sub-clause** icon to add a conditional sub-clause to the conditional clause, if you want.
6. Click the match-operator link to select the matching condition for which the conditional clause evaluates to true. The values are:

Option	Description
Match Any	Conditional clause is true if any of the condition evaluates to true.

Option	Description
Match All	Conditional clause is true only if all of the conditions evaluates to true.
Match None	Conditional clause is true only if none of the conditions evaluates to true.

- Click the **Save** icon  to save updates to the business rule.

Adding an Action Using the Command Builder

The Then section defines additional actions to be taken based on the conditions defined.

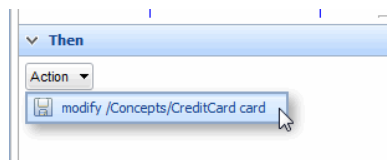
The builder lets the user define actions based on the statements defined in the Action Context section of the rule template.



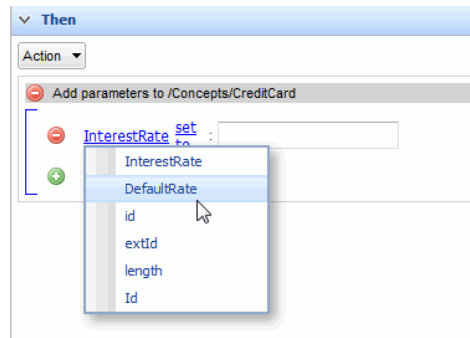
Ensure that all the actions in the business rule are defined and the order should match that of the rule template. Otherwise, the business rule will be invalid.




Procedure

- In the business rule editor, under the Then section, click the **Action** dropdown list to select the actions available for the rule template.



- Click the artifact-name link and select the artifact you want modify.



- Click the operator-name link and select the command for the artifact. See [Action Builder Operators](#) for list of available on operators.
- Type the value, you want to set for the artifact, in the empty box.
- Click the **Add** icon  to add another parameter for the action, or click the **Remove** icon  to remove the parameter for action, if you want.
- Click the **Save** icon  to save updates to the business rule.

Exporting a Business Rule


You can export business rules from TIBCO BusinessEvents WebStudio to rule template instance file.

A business rule in a project is identified by the .ruletemplateinstance file extension. The rule template instance file is generated when the BuildAndDeploy option is selected for the business rule in approvals. See [Generating and Deploying the Business Rule's Rule Template Instance](#) for more details.

The sample file content of the exported rule template instance file is:

```
<?xml version="1.0" encoding="UTF-8"?>
<RuleTemplateInstance id="738cd4ea-8569-4744-aa86-fb88a89bf0a3" implementsPath="/
Rule_Templates/Applicant_PreScreen">
  <actions/>
  <binding id="minimumIncome" value="5000"/>
  <binding id="minimumAge" value="25"/>
  <binding id="creditType"/>
</RuleTemplateInstance>
```

Procedure

1. Open the Export dialog box in either of the following ways:
 - Select the business rule and in the **Tools** tool bar, click the **Export** icon .
 - Right click the business rule in the Group Contents section and select **Export**.
2. Navigate to the directory in which you want to save the rule template instance file and provide a file name.
3. Click **Save**.

Deleting a Business Rule


If you want to delete a business rule that has been checked into the RMS project, then you must submit the deletion for approval.

After approval, the deleted business rule is removed from the RMS project, but it is not removed from the user's workspace.



If you are not sure whether the business rule you deleted was committed to the RMS project in a prior action, select **Project > Commit**. If the deleted business rule does not appear in the Changes Made panel, then it is a local business rule.

Procedure

1. Do one of the following:
 - In **Group Contents**, right-click the business rule name and select **Delete**.
 - In **Group Contents**, select the business rule name. Then click the **Delete** icon  in the **Edit** toolbar.
2. If the business rule has been committed to the RMS project, select one of the following options as appropriate:
 - Commit the project or the parent folder so that the RMS master copy of the business rule can be deleted after the approval (see [Committing Artifacts for Approval](#)). OR
 - Synchronize the project to replace the deleted copy with the RMS project version of the business rule (see [Updating \(Synchronizing\) a Project](#)).



Users who have already checked out the project must manually delete their local copy of the artifact.

Renaming a Business Rule

You can rename a business rule in WebStudio to a more suited name that identifies its functionality more appropriately.

Procedure

1. In **Group Contents**, right-click the business rule name and select **Rename**.



A business rule name can contain letters, numbers, and underscore only. It should start with a letter and be less than 64 characters.

WebStudio displays the **Rename completely successful** message.

2. If the business rule has been committed to the RMS project, commit the project or the parent folder so that the RMS master copy of the business rule is also renamed.

The changes appear in two entries: one with the **Change Type** as **Added** and other with the **Change Type** as **Removed** in the Commit window and the Worklist window. Commit both the entries together as a single commit; otherwise WebStudio displays an error.

Business Rule Operators

Business Rule builder operators consists of the conditional clause operators and the action builder operators. The operator list changes as per the parent entity type in the builder.

Condition Builder Operators

Operator Value on WebStudio Client	Parent Type	Operator Value on RMS Server
Matches Other Field	<ul style="list-style-type: none"> • Concept / Event • Data / Time / Datetime • Integer / Float • Text / String • Boolean 	<code>== <field></code>
Differs From Field	<ul style="list-style-type: none"> • Concept / Event • Data / Time / Datetime • Integer / Float • Text / String • Boolean 	<code>!= <field></code>
Is Null	<ul style="list-style-type: none"> • Concept / Event • Data / Time / Datetime • Integer / Float • Text / String 	<code>== null</code>

Operator Value on WebStudio Client	Parent Type	Operator Value on RMS Server
Is not Null	<ul style="list-style-type: none"> • Concept / Event • Data / Time / Datetime • Integer / Float • Text / String 	!= null
Greater Than	Integer / Float	>
Greater Than Field	Integer / Float	> <field>
Greater Than Equal To	Integer / Float	>=
Greater Than Equal To Field	Integer / Float	>= <field>
Less Than	Integer / Float	<
Less Than Field	Integer / Float	< <field>
Less Than Equal To	Integer / Float	<=
Less Than Equal To Field	Integer / Float	<= <field>
Equals	<ul style="list-style-type: none"> • Integer / Float • Text / String • Boolean 	==
Not Equals	<ul style="list-style-type: none"> • Integer / Float • Text / String • Boolean 	!=

Action Builder Operators

Operator Value on WebStudio Client	Parent Type	Operator Value on RMS Server
Set To	<ul style="list-style-type: none"> • Concept / Event • Data / Time / Datetime • Integer / Float • Text / String 	=

Operator Value on WebStudio Client	Parent Type	Operator Value on RMS Server
Set To Field	<ul style="list-style-type: none"> • Concept / Event • Data / Time / Datetime • Integer / Float • Text / String • Boolean 	= <field>
Set To Null	<ul style="list-style-type: none"> • Concept / Event • Data / Time / Datetime • Integer / Float • Text / String 	= null
Increment By	Integer / Float	+=
Increment By Field	Integer / Float	+= <field>
Decrement By	Integer / Float	-=
Decrement By Field	Integer / Float	-= <field>
Set To True	Boolean	= true
Set To False	Boolean	= false

Process

You can use the WebStudio software to create and design a new process without opening the BusinessEvents Studio.

Using WebStudio, you can design the complete process and set the General and Documentation properties, which instructs the developers on what to implement for the process.

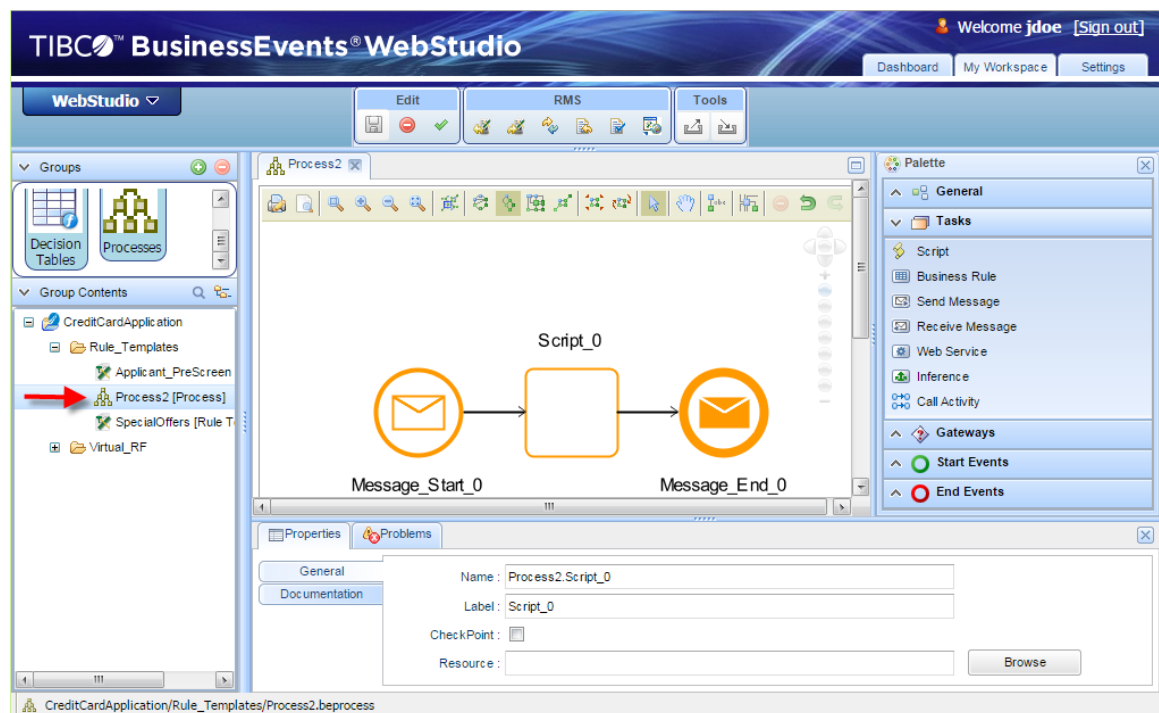


Processes are available in TIBCO BusinessEvents WebStudio in the view mode and the edit mode only if you have the TIBCO BusinessEvents Process Orchestration add-on installed on your system.

The process editor contains three sections:

- **Diagram Editor** - You can design the process in the diagram editor by adding the component from the palette. It has a toolbar, using which you zoom in or zoom out, change how the components are arranged, undo the command, print the command, and so on. Hover your mouse over each tool to see the tooltip, which displays the tool's name.
- **Properties** - The Properties section contains two common tabs: **General** and **Documentation**. Using the **General** tab, you can specify the **Name**, **Label**, and **Resource** for the component. You can write instructions for the developers on what to implement for the process components, using the **Documentation** tab. After the process is committed in RMS and approved, the developers can see these documented instructions in the TIBCO BusinessEvents Studio after logging into RMS.
- **Palette** - All the components that you can use to design the process diagram are categorised in different palettes. See [Process Palette Reference](#) for a complete list of available components in WebStudio.


Process Editor in TIBCO BusinessEvents WebStudio



Adding a New Process

You can create a new process and design the complete process flow in WebStudio.

Procedure

1. In the **My Workspace** tab, under the **Group Content** section, click the **Change View** icon  and select **Show items as a tree** (if not already selected).
The project artifacts are now displayed in the tree structure.
2. Right-click the folder where you want to add the new process and select **New Process**.
The New Process window is displayed.
3. Specify the process name in the **Process** field. If needed, you can change the parent folder name to an existing folder or a new folder in the **Parent Folder** field.
The new process is added to the project and is displayed in the process editor for update.

Result

The new process contains three basic components by default: Message_Start_0, Message_End_0, and Script_0. You can add other components from the palettes. Click the component in the palette and click the editor to add the component.

Deleting a Process


If you want to delete a process that has been checked into the RMS project, you must submit the deletion for approval.

After approval, the deleted process is removed from the RMS project, but it is not removed from the user's workspace.



If you are not sure whether the process you deleted was committed to the RMS project in a prior action, select **Project > Commit**. If the deleted process does not appear in the Changes Made panel, it is a local process.

Procedure

1. Do one of the following:
 - In **Group Contents**, right-click the process name and select **Delete**.
 - In **Group Contents**, select the process name. Then click the **Delete** icon  in the **Edit** toolbar.
2. If the process has been committed to the RMS project, select one of the following options as appropriate:
 - Commit the project or the parent folder so that the RMS master copy of the process can be deleted after the approval (see [Committing Artifacts for Approval](#)). OR
 - Synchronize the project to replace the deleted copy with the RMS project version of the process (see [Updating \(Synchronizing\) a Project](#)).



Users who have already checked out the project must manually delete their local copy of the artifact.

Process Palette Reference

The Palette in WebStudio contains the components you can use to design a process diagram.

General

Name	Description
Note	A free-form text note.
Association	An association is a dotted line that links flow objects with non-flow objects such as text annotations.
Sequence Flow	A link or transition between two activities is called a sequence flow.

Tasks

Name	Description
Script	Executed using a TIBCO BusinessEvents rule function
Business Rule	Executed using a TIBCO BusinessEvents Decision Manager decision table
Send Message	Sends a TIBCO BusinessEvents event to its default destination
Recieve Message	Receives a TIBCO BusinessEvents event
Web Service	Executed using a web service
Inference	Executed using a TIBCO BusinessEvents rule
Call Activity	Executed using a private process

Gateways

Name	Description
Exclusive	Used to execute one sequence flow of execution after evaluating multiple sequence flows in a process
Parallel	Used to execute concurrent multiple flows of execution

Start Events

Name	Description
Start	No trigger.
Message Start	Triggered by a simple event from a queue-based destination.
Timer Start	Triggered by a time event.
Signal Start	Triggered by a simple event from a topic-based destination. Multiple processes can be triggered by the same event.

End Events

Name	Description
End	No action occurs
Message End	Sends a simple event to a queue-based destination
Error End	Throws an appropriate error
Signal End	Sends a simple event to a topic-based destination

Renaming a Process

You can rename a process in WebStudio to a more suited name that identifies its functionality more appropriately.

Procedure

1. In **Group Contents**, right-click the process name and select **Rename**.



A process name can contain letters, numbers, and underscore only. It should start with a letter and be less than 64 characters.

WebStudio displays the `Rename completely successful` message.

2. If the process has been committed to the RMS project, commit the project or the parent folder so that the RMS master copy of the process is also renamed.

The changes appear in two entries: one with the **Change Type** as **Added** and other with the **Change Type** as **Removed** in the Commit window and the Worklist window. Commit both the entries together as a single commit; otherwise WebStudio displays an error.

Approval Workflow Overview

After updating the artifacts (decision tables or business rules), submit them for approval. All business user actions, such as committing or deleting artifacts are committed to the RMS project only after a approver approves those updates.

The approval permissions are set at the project level using the ACL file for the project. Allow the approval action type for the user role in the ACL file to the grant approval permission. See *TIBCO BusinessEvents Administration* for details on setting permission for a user role at the project level.



This manual documents the behavior of the product as shipped. The workflow, the roles used, and the permissions granted to the roles are all configurable. However, the general flow is likely to be similar to that described in this guide.

Use of RMS—and therefore the approval process—is required when TIBCO BusinessEvents WebStudio is used.

See [Committing Artifacts for Approval](#) for business user procedures.

See [Working with the Worklist Items](#) for approver procedures.

Who Can Approve or Reject Commit Requests

In TIBCO BusinessEvents WebStudio users with Administrator role permissions can approve or reject commit requests, or delegate approval. They can also delegate the task of approving or rejecting commit requests to any specified user role and not to a specific user. In this case, the approval or rejection requests will be sent to all the users in that role. See [Delegating a Workitem from Worklist](#).

Approval Status Values

COMMITTED

When a user commits one or more resources for approval, the status of the request is set to Committed. The approver then sets the status as appropriate.

APPROVE

The request was approved. If the submission request was to add or change an artifact, that artifact is copied to the RMS project. If the submission request was to delete an approved artifact, that artifact is deleted from the RMS project.

REJECT

The change was rejected. If the submission request was to add or change an artifact, that artifact is not copied to the RMS project. If the submission request was to delete an approved artifact, that artifact is not deleted. It is up to the business user to make changes in their local project accordingly.

BUILDANDDEPLOY

The artifact is built and hot deployed. The artifact after approval is built and hot deployed at the location specified by a property in the RMS.cdd file. See the following sections on how to build the artifact and hot deploy it:

- [Generating and Deploying the Decision Table's Class File](#)
- [Generating and Deploying the Business Rule's Rule Template Instance](#)



- The **BUILDANDDEPLOY** status is not applicable for the Process artifacts.
- If the `<ProjectName>.<EnvironmentName>.ws.jmx.hotDeploy.enable` property is set to `false` then the **BUILDANDDEPLOY** option only compiles the classes but do not hot deploy them.

Worklist

Approval requests are handled in the Worklist window in TIBCO BusinessEvents WebStudio. You can take the appropriate approval action on single or multiple artifacts at a time in the Worklist window.

Worklist have multiple columns to provide more information about each request:

Revision ID

Unique ID assigned to each revision by WebStudio.

User name

Username of the user who had performed the commit.

Project

Project name of the artifact.

Commit comments

Comments provided by user at the time of the commit.


Commit time

RMS system time of the commit.

Viewing Differences between Committed and Previous Version of Artifact

View the updates that have been made in the committed version, from the previous approved version of the artifact in RMS, before approving or rejecting the request.

Procedure

1. Open the worklist window using any of the following methods:
 - Click **WebStudio**, with the down-pointing triangle in front of it, and select **RMS > Show Worklist**.
 - Right-click any artifact and select **RMS > Show Worklist**.
 - Click the **Show Worklist** icon  in the RMS toolbar.


The worklist window displays all the requests submitted for approval.

2. To view the artifacts submitted for approval, click the triangle in front of the Revision ID.
3. Right-click the modified artifact and select **Diff with Previous Version**.
The updated artifact is displayed with the updates highlighted in different colors for addition, deletion, and modification. Refer to the legends displayed on the editor for the significance of each highlighted color. Hover the mouse pointer over the highlighted (modified) part to see the previous value.
4. After viewing changes, open the worklist again to approve or reject the commit request (see [Approving or Rejecting a Commit Request from Worklist](#)).

Approving or Rejecting a Commit Request from Worklist

Approve or reject requests after they are committed to RMS. You should have allow permission for the approval action type in the project's access control file.

Procedure

1. Open the worklist window using any of the following methods:
 - Click **WebStudio**, with the down-pointing triangle in front of it, and select **RMS > Show Worklist**.
 - Right-click any artifact and select **RMS > Show Worklist**.
 - Click the **Show Worklist** icon  in the RMS toolbar.

The worklist window displays all the requests submitted for approval.

2. To view the artifacts submitted for approval, click the triangle in front of the Revision ID. To examine the artifact before taking an action, double-click the artifact. The decision table or business rule displays in the editor.
3. To take action on an artifact, click the `Committed` status under the Status column, and select a value from the **Status** drop-down list:
 - Approve
 - Reject

A message field displays the error messages in the event of a problem.

4. Click anywhere outside the drop-down menu and click **Ok**.


Delegating a Workitem from Worklist

A user in the Administrator role can delegate a workitem to one or more roles. Users belonging to any of those roles see the workitem in their worklist.



Users in the role specified by the RMS global variable property `tibco.clientVar.RMS/Approval/adminRole` have Administrator role privileges. See [Configuring RMS Server Properties](#).

Procedure

1. Open the worklist window using any of the following methods:
 - Click **WebStudio**, with the down-pointing triangle in front of it, and select **RMS > Show Worklist**.
 - Right-click any artifact and select **RMS > Show Worklist**.
 - Click the **Show Worklist** icon  in the RMS toolbar.

The worklist window displays all the requests submitted for approval.

2. Select the checkbox for one or more Revision IDs you need to delegate.
3. Select one or more user roles from the **DelegateTo** dropdown list and click **Apply**.
You see all roles that you do not belong to, and that have permission to approve or reject the artifact. The workitem disappears from your worklist. It can be seen in the worklist of the users in the roles you selected.

Deleting Workitems from Worklist

You can delete the workitems from your worklist, which are either in the Approve or Reject status.

The WebStudio displays the following error message if you try to delete any workitem revision which has even a single artifact in the Commit status:

Revisions having artifacts which are not yet approved cannot be deleted.




Individual artifacts entry could not be deleted from the worklist. Only complete revisions could be deleted, after all its artifacts are in the Approve or Reject status.

Prerequisites

Approve or reject the commit request before deleting the workitems from the worklist.

Procedure

1. Open the worklist window using any of the following methods:
 - Click **WebStudio**, with the down-pointing triangle in front of it, and select **RMS > Show Worklist**.
 - Right-click any artifact and select **RMS > Show Worklist**.
 - Click the **Show Worklist** icon  in the RMS toolbar.

The worklist window displays all the requests submitted for approval.

2. Select the check box for one or more Revision IDs you need to delete.
3. Select **Delete** from the drop-down list and click **Apply**. All the selected revisions are now deleted from your worklist and WebStudio displays the *Revision(s) [Revision ID] successfully deleted* message.

Generating Deployable Files (EAR and Class Files)

Deployable files are generated using the TIBCO BusinessEvents Studio project files, located in the RMS project's folder.

The deployable files are of two types: EAR files and class files.

EAR files

EAR files are generated in Decision Manager and TIBCO BusinessEvents WebStudio UI.

Class files

You can generate class files using Decision Manager and TIBCO BusinessEvents WebStudio UI, or at the command line. You can generate an entire project's class files, one decision table's class file, or one business rule's rule template instance file. If you generate an entire project's class files, a property in the CDD file enables you to exclude unwanted packages.




You can generate one decision table's class file and business rule's rule template instance file, only using TIBCO BusinessEvents WebStudio only.

Generating the Project EAR or All Project Class Files

You can generate deployable EAR files only if you have the permission to check out all the required project resources (ACLs may limit what you can check out).

Procedure

1. Open the Generate Deployable window using any of the following methods:
 - Click **WebStudio**, with the down pointing triangle in front of it, and select **RMS > Generate Deployable**.
 - Right-click on any artifact and select **RMS > Generate Deployable**.
 - Click the **Generate Deployable** icon  in the RMS toolbar.

The Generate Deployable window is displayed.
2. Do one of the following:
 - To generate class files, select the **Generate Classes Only** checkbox.
 - To generate EAR files, do not select the **Generate Classes Only** checkbox.
3. From the drop-down list, select the project for which you want to generate the deployable files.
4. (For EAR files only.) If you want to generate the debug information, select the **Generate Debug Info** checkbox.
5. (For EAR files only.) If you want to include all service-level global variables, select the **Include all service level global variables** checkbox.
6. Click **OK**.

Result

Files are saved to the preconfigured location (see [Generated Files Location](#)). Any existing files are overwritten.

Generating and Deploying the Decision Table's Class File

In TIBCO BusinessEvents WebStudio, using the BuildAndDeploy status menu option in the Worklist window you can generate the class file and hot deploy the decision table.




You can generate one decision table's class file using TIBCO BusinessEvents WebStudio only.

Prerequisites

- For generation of class file and hot deployment of a decision table, you must have a generated EAR for the project. See [Generating the Project EAR or All Project Class Files](#) for more details.
- Ensure that JMX connection details for the environments where you must deploy are configured for the projects in the hot deploy properties group in the RMS.cdd file. See [Hot Deploy Property Group Reference](#) for more details on the hot deploy properties. Using these properties you can deploy multiple projects in a single environment or a single project in multiple environments.
- For hot deployment of the decision table, ensure that the project's CDD file (for which the decision table is hot deployed) have the `<ProjectName>.<EnvironmentName>.ws.jmx.hotDeploy.enable` property set to `true`. See [Hot Deploy Property Group Reference](#) for more details on the hot deploy properties.
- You can generate the class file and hot deploy a decision table only after it is approved. See [Working with the Worklist Items](#) on how to approve a committed request.
- For hot deployment of the decision table, ensure that the project's CDD file (for which the decision table is hot deployed) have the `be.engine.cluster.externalClasses.path` property under the appropriate processing unit, for example `default`. The property points to the location where artifacts are generated and also identified by the RMS property `ws.artifact.deploy.location` (see [Common Property Group](#) for more details on the property). For example, the decision table class file for the CreditCardApplication project is stored at `BE_HOME\rms\shared\CreditCardApplication`.

Procedure

1. Open the worklist window using any of the following methods:
 - Click **WebStudio**, with the down pointing triangle in front of it, and select **RMS > Show Worklist**.
 - Right-click any artifact and select **RMS > Show Worklist**.
 - Click the **Show Worklist** icon  in the RMS toolbar.

The worklist window displays all the requests submitted for approval as well as approved requests.
2. To view the approved decision tables, click the triangle in front of the **Revision ID**. To examine the decision table before taking an action, double-click the table. The decision table displays in the editor.
3. To generate classes and hot deploy the decision table, click the **Approved** status under the **Status** column, and select **BuildAndDeploy** from the status drop-down menu.



If the `<ProjectName>.<EnvironmentName>.ws.jmx.hotDeploy.enable` property is set to `false` then the **BuildAndDeploy** option only compiles the classes but do not hot deploy them.

WebStudio performs auto-validation of the artifact to ensure that the update does not affect the runtime execution.

A message field displays error messages in the event of a problem.

4. To start generating classes click anywhere outside the drop-down menu and click **Ok**.


Generating and Deploying Business Rules

In TIBCO BusinessEvents WebStudio, using the BuildAndDeploy status menu option in the Worklist window you can generate the rule template instance file and hot deploy the business rule.

Prerequisites

- For generation of rule template instance file and hot deployment of a business rule, you must have a generated EAR for the project. See [Generating the Project EAR or All Project Class Files](#) for more details.
- Ensure that JMX connection details for the environments where you must deploy are configured for the projects in the hot deploy properties group in the RMS.cdd file. See [Hot Deploy Property Group Reference](#) for more details on the hot deploy properties. Using these properties you can deploy multiple projects in a single environment or a single project in multiple environments.
- For hot deployment of the business rule, ensure that the project's CDD file (for which the business rule is hot deployed) have the `<ProjectName>.<EnvironmentName>.ws.jmx.hotDeploy.enable` property set to true. See [Hot Deploy Property Group Reference](#) for more details on the hot deploy properties.
- You can generate the class file and hot deploy a decision table only after it is approved. See [Working with the Worklist Items](#) on how to approve a committed request.
- For hot deployment of the business rule, ensure that the project's CDD file (for which the business rule is hot deployed) have the `be.cluster.ruletemplateinstances.deploy.dir` property under the appropriate processing unit, for example `default`. The property points to the location where artifacts are generated and also identified by the RMS property `ws.artifact.deploy.location` (see [Common Property Group](#) for more details on the property). For example, the rule template instance file for the CreditCardApplication project is stored at `BE_HOME\rms\shared\CreditCardApplication`.

Procedure

1. Open the worklist window using any of the following methods:
 - Click **WebStudio**, with the down-pointing triangle in front of it, and select **RMS > Show Worklist**.
 - Right-click any artifact and select **RMS > Show Worklist**.
 - Click the **Show Worklist** icon  in the RMS toolbar.

The worklist window displays all the requests submitted for approval as well as approved requests.
2. To view the approved business rules, click the triangle in front of the **Revision ID**. To examine the business rule before taking an action, double-click the rule.
The business rule is displayed in the editor.
3. To generate rule template instance file and hot deploy the business rule, click the **Approved** status under the **Status** column, and select **BuildAndDeploy** from the status drop-down menu.



If the `<ProjectName>.<EnvironmentName>.ws.jmx.hotDeploy.enable` property is set to false then the **BuildAndDeploy** option only compiles the classes but do not hot deploy them.

WebStudio performs auto-validation of the artifact to ensure that the update does not affect the runtime execution.

A message field displays error messages in the event of a problem.

4. To start generating rule template instance file, click anywhere outside the drop-down menu and click **Ok**.

Generated Files Location

Location of generated files can be configured using deploy location properties in the `RMS.cdd` file.

See [RMS Server Configuration Property Reference](#) for more details on deploy location properties.

Below are the locations of the generated files as per the preconfigured locations in the shipped product.

- EAR files are saved in the RMS project directory in the RMS's shared directory. For example, EAR file for CreditCardApplication project is stored at `BE_HOME\rms\shared\CreditCardApplication`.
- Project's class files are saved in the `codegen` subdirectory under the RMS project directory in the RMS's shared directory. For example, class files for CreditCardApplication project are stored at `BE_HOME\rms\shared\CreditCardApplication\codegen`.
- Individual decision table's class file and business rule's rule template instance are saved in the RMS project directory in the RMS's shared directory.

Settings

Using the Settings tab you can configure WebStudio settings. This tab has multiple portlets for different purposes.

Preferences Settings

Set various preferences for TIBCO BusinessEvents WebStudio using the options available under the **Preferences** tab.

After updating the values, click **Apply** to set new WebStudio preferences.

Field	Default Value	Description
Number of Recently Opened Items	10	The number of recently open items to be displayed in the Recently Opened dashboard portlet.
Default URL for Custom Webpage	www.tibco.com	The default URL for the Web Page dashboard portlet.
SCS User Name	jdoe	<p>The username to connect to the SVN repository mentioned in the <code>ws.scs.rootURL</code> property of the <code>RMS.cdd</code> file. This username overrides the default username provided in the <code>ws.scs.default.username</code> property of the <code>RMS.cdd</code> file.</p> <p>See Source Control System (SCS) Property Group for more details on the <code>RMS.cdd</code> properties.</p>
SCS Password	jdoe	<p>The password for the SCS User Name field to connect to the SVN repository mentioned in the <code>ws.scs.rootURL</code> property of the <code>RMS.cdd</code> file. This password overrides the default password provided in the <code>ws.scs.default.password</code> property of the <code>RMS.cdd</code> file.</p> <p>See Source Control System (SCS) Property Group for more details on the <code>RMS.cdd</code> properties.</p> <p>Unlike the <code>ws.scs.default.password</code> property of the <code>RMS.cdd</code> file, this field takes the actual password and not the Base64 formatted password.</p>
View Project as	List	The list or tree view of the project artifacts under the Group Content in workspace as well as in Dashboard Portlets.
Decision Table Page Size	20	The number of rows of the decision table to be displayed in a page. Reopen the decision table, or disable and then again enable the page view, to reflect the updated page size for the decision table. The minimum page size is limited to 10.
Auto UnLock on Approve/Reject	selected	If selected, the property enables the artifact to be unlocked once they are either approved or rejected for the committed update.
Group Related Artifacts	selected	If selected the business rules are grouped under their parent rule templates, and decision tables are grouped under their parent virtual rule function. If the check box is clear, the decision table and rule templates are not grouped, but are listed individually.

Field	Default Value	Description
Allow Custom Domain Values	selected	If selected, the custom values can be entered for the attributes associated to the domain. If the check box is cleared, you can only select from the domain values list using the drop-down.
Show column alias if present	cleared	If selected then column alias are displayed for the decision table columns. Column alias can be defined using the column option Field Settings .

Application Preferences Settings

Using the Application Preferences portlet, you can restrict the operators available for creating the Business Rule using the Builder interface in WebStudio. This portlet is available only for the users with the administrator role.

The Operator Preferences section in the portlet is consist of three columns.

Preference	Description
Field Type	<p>Lists the type of the fields available in the condition and command builder. Select the field type for which you want to restrict the available operators. The default field types are:</p> <ul style="list-style-type: none"> • String • Integer • Long • Double • Boolean • DateTime • Concept/Event
Filter Operators	<p>Lists the operators available in the condition (When) builder for the selected Field Type. Select the operators that you want to display in the condition builder for the selected Field Type and click Apply.</p> <p>The operator with the clear checkbox does not appear in the Builder interface while creating conditional clause for the Business Rule.</p>
Command Operators	<p>Lists the operators available in the command (Then) builder for the selected Field Type. Select the operators that you want to display in the command builder for the selected Field Type and click Apply.</p> <p>The operator with the clear checkbox does not appear in the Builder interface while defining actions for the Business Rule.</p>

User Interface Customization

You can customize various aspects of BusinessEvents WebStudio user interface (UI) using the .WAR file.

In WebStudio you can change the CSS styles (colours, borders, and so on), images, background images, message strings, various titles and tooltips. Thus, you can create a new theme for WebStudio that is more in line with your corporate theme. All the updates are completely reversible to the default theme.

You can change the following UI elements in WebStudio:

- Company logo
- CSS
- Images
- Message or properties
- Language

Customizing the Company Logo

WebStudio can be configured to display your company logo on the login page and the banner page.

Procedure

1. Extract the `WebStudio.war` file located at the `BE_HOME/rms/bin/` folder.
2. In the extracted folder, open the `WebStudio.html` file in edit mode.
3. In the `WebStudio.html` file, set the value of the `company_logo_url` variable to the URL of your logo image.

The URL can be a hyperlink to the logo image which might be hosted on some web-server, for example, `http://www.mycompany.com/images/logo.jpg`. The URL can also point to an image on the file system within the WebStudio folder hierarchy, for example, `webstudio/images/logo.png`. Ensure that the image is present in the WAR file at the path specified.



To remove the company logo, set the value of the `company_logo_url` property to the blank string (`company_logo_url = ""`).

4. Save the HTML file and compress the folder back to the `WebStudio.war` file.

Customizing CSS

You can create a new theme, in line with your corporate theme, for WebStudio by configuring the CSS of the Webstudio.

Procedure

1. Extract the `WebStudio.war` file located at the `BE_HOME/rms/bin/` folder.
2. In the extracted folder, open the `styles/custom_styles.css` file in edit mode and populate it with CSS styles that you want to update (override).
Refer to the instructions within the file. Save the file after the update.
3. In the extracted folder, open `styles/WebStudio.css` file in edit mode. Uncomment (remove #) the first line to import `custom_styles.css` and save it.
4. Compress the folder back to the `WebStudio.war` file.

Customizing Images

In WebStudio you can use your own customized images in place of the default images. A typical example is a delete or add icon is represented through a very specific icon throughout all your corporate applications and you want the same to appear in WebStudio.

Procedure

1. Extract the `WebStudio.war` file located at the `BE_HOME/rms/bin/` folder.
2. In the extracted folder, back up the images folder by duplicating the folder at the same level with a name, such as, `my_images` or any name of your choice.
3. Update the images under the duplicate folder (`my_images`) as per your requirement without changing the file names.
4. Open the `WebStudio.html` file in edit mode.
5. In the `WebStudio.html` file, set the value of the `images_dir` variable to point to the `my_images` folder.

For example,

```
var images_dir = "webstudio/my_images/"
```

6. Save the HTML file and compress the folder back to the `WebStudio.war` file.



To revert to the WebStudio's default images, set the value of the `images_dir` variable to "webstudio/images/" in `WebStudio.html` file.

Customizing Messages and Properties

If needed, you can customize messages and labels in WebStudio user interface (UI) to explain the right meaning and context, or add a new language support.

See [Adding a New Custom Language Pack](#) for more details on customizing messages and properties for a new language pack.

Procedure

1. Browse to the `BE_HOME/rms/lib/locales` directory, which contains the locale JAR files. Each JAR (`<locale_name>.jar`) file correspond to a different language. Each JAR file (`filename-<locale_name>.properties`) consist of the different properties files corresponding to various areas of the application.
2. Extract the corresponding locale JAR file to edit one of the existing values of the properties.
3. Identify the files or properties that you need to change. Update the required files and properties and compress it back into the JAR file.

What to do next

Refresh the web browser to see the changes in WebStudio. If you don't see the changes, perform a browser cache cleanup (as sometimes the values are loaded from the cache) and reload the URL.

Changing WebStudio Language

You can change the default language of WebStudio from American English (`en_US`) to other languages.

Prerequisites

Ensure that the language pack of the language, in which you want WebStudio to be displayed, is installed. See *TIBCO BusinessEvents Installation* for more details.

Procedure

1. Clear the web browser cache.
2. Open WebStudio URL in the web browser. Put the `locale=<language>` or `locale=<language_country>` as query string of URL and reload URL.



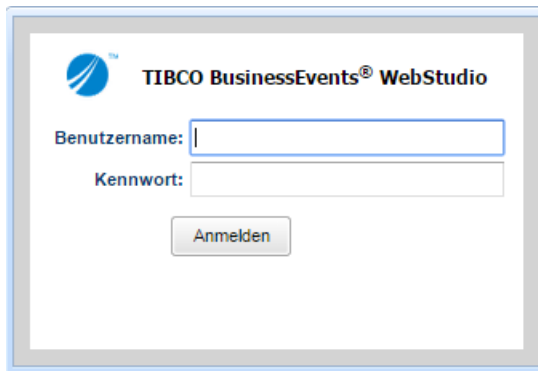
If you load the language that is not supported, the English language is picked up.

For example, to load WebStudio with the German language, put the URL as the following:

`http://localhost:8090/WebStudio/?locale=de`

The URL is case sensitive.

WebStudio German Language Login Page



Adding a New Custom Language Pack

You can manually translate WebStudio to use a new language, if the language you need is not available in the language pack.

There are about seven properties inside the locale JAR corresponding to various areas of the application. The format is defined as `filename_<language>.properties` or `filename_<language_country>.properties`.

Procedure

1. Browse to the `BE_HOME/rms/lib/locales` directory, which contains the locale JAR files. Each JAR file correspond to a different language. The format of the locale JAR file is `<locale_name>.jar`.
2. Create temporary directory and un-package the English JAR (`en_US.jar`) files into the temporary directory.
3. Rename each of the property file with the corresponding language specific values. There are about seven properties inside the locale JAR corresponding to various areas of the application. The format is defined as `filename_<language>.properties` or `filename_<language_country>.properties`. For example, rename the property file from `DTMessages.properties` to `DTMessages_de.properties` for the German language.
4. Open each of renamed property file and replace the values in each of the property file with the corresponding language specific values.
5. Re-package the files back into the JAR file. Ensure the correct naming convention for the property file as well as the JAR file.

What to do next

Refresh the web browser to see the changes in WebStudio. If you don't see the changes, perform a browser cache cleanup (as sometimes the values are loaded from the cache) and reload the URL.