



OpenL Tablets WebStudio User's Guide

OpenL Tablets 5.9.3

OpenL Tablets BRMS

Document number: TP_EPB_7.1.0_OpenLWSt_User_3.0_SN
Revised: 07-16-2012



OpenL Tablets Documentation is licensed under a [Creative Commons Attribution 3.0 United States License](https://creativecommons.org/licenses/by/3.0/us/).

Table of Contents

Preface.....	5
Audience	5
Related Information	5
Typographic Conventions.....	5
Chapter 1: Introducing OpenL Tablets WebStudio.....	7
What Is OpenL Tablets WebStudio?	7
Working with Projects in OpenL Tablets WebStudio.....	7
OpenL Tablets WebStudio Components	8
Security Overview	9
Chapter 2: Getting Started	10
Logging In to OpenL Tablets WebStudio	10
Understanding the User Interface	10
Rule Editor	10
Repository Editor	16
Chapter 3: Using Rule Editor	17
Filtering Projects	17
Opening a Module	18
Managing Projects.....	18
Checking Out and Checking In a Project.....	18
Reverting Project Changes.....	18
Viewing Tables.....	21
Modifying Tables	22
Performing a Search	23
Simple Search	24
Advanced Search	24
Creating New Table.....	26
Datatype Table Wizard	27
Data Table Wizard	28
Chapter 4: Editing and Testing Functionality	31
Editing Tables	31
Unit Tests.....	35
Navigation.....	36
Run Tests.....	37
Creating New Test.....	39
Tracing.....	40
Benchmarking	43
Chapter 5: Using Repository Editor	45
Browsing Design Time Repository.....	45
Filtering the Project Tree	46
Uploading a Project.....	46
Creating a Project	47
Opening a Project	48
Closing a Project	48

Checking Out a Project	48
Checking In a Project	49
Defining Project Dependencies	49
Modifying a Project.....	50
Modifying Project Properties.....	51
Modifying Project Contents.....	52
Copying a Project.....	55
Removing a Project.....	55
Deleting a Project	55
Erasing a Project	55
Deploying Projects	56
Creating a Deployment Project.....	56
Defining Deployment Project Descriptors.....	56
Deploying a Deployment Project	57
Opening Deployed Projects	57
Redeploying Projects.....	57
Comparing Project Versions.....	58
Using Administration Tools	60
Common Settings.....	60
Security Settings	61
Repository Settings	61
System Settings	61
Index	63

Preface

This preface is an introduction to the *OpenL Tablets WebStudio User's Guide*.

The following topics are included in this preface:

- [Audience](#)
- [Related Information](#)
- [Typographic Conventions](#)

Audience

This guide is intended for the following users:

Audience		
User type	Purpose	Required knowledge
Business users	View and modify company business rules stored in tables.	Knowledge of decision tables is required.
Developers	<ul style="list-style-type: none"> • Manage technical details of rule tables. • Organize and deploy rule projects. 	Knowledge of OpenL Tablets technology is required.

Related Information

OpenL Tablets WebStudio is a tool from OpenL Tablets product. For information on OpenL Tablets Rules, see [OpenL Tablets Reference Guide](#).

Typographic Conventions

The following styles and conventions are used in this guide:

Typographic styles and conventions	
Convention	Description
Bold	<ul style="list-style-type: none"> • Represents user interface items such as check boxes, command buttons, dialog boxes, drop-down list values, field names, menu commands, menus, option buttons, perspectives, tabs, tooltip labels, tree elements, views, and windows. • Represents keys, such as F9 or CTRL+A. • Represents a term the first time it is defined.
<code>Courier</code>	Represents file and directory names, code, system messages, and command-line commands.
Courier Bold	Represents emphasized text in code.
Select File > Save As	Represents a command to perform, such as opening the File menu and selecting Save As .

Typographic styles and conventions	
Convention	Description
<i>Italic</i>	<ul style="list-style-type: none">• Represents any information to be entered in a field.• Represents documentation titles.
< >	Represents placeholder values to be substituted with user specific values.
Hyperlink	Represents a hyperlink. Clicking a hyperlink displays the information topic or external source.

Chapter 1: Introducing OpenL Tablets WebStudio

This section introduces the main OpenL Tablets WebStudio concepts.

The following topics are included in this section:

- [What Is OpenL Tablets WebStudio?](#)
- [Working with Projects in OpenL Tablets WebStudio](#)
- [OpenL Tablets WebStudio Components](#)
- [Security Overview](#)
- [User Perspectives](#)

What Is OpenL Tablets WebStudio?

OpenL Tablets WebStudio (further also referred as **WebStudio**) is a web application employed by business users and developers to view, edit, and manage business rules and rule projects created using OpenL Tablets technology. For information on OpenL Tablets, see [OpenL Tablets Reference Guide](#).

By using OpenL Tablets WebStudio, users can modify rules directly in a web browser without installing additional tools. OpenL Tablets WebStudio provides better functionality than the OpenL Tablets Eclipse feature in terms of browsing projects, modifying rules, viewing errors, and executing tests. However, for more advanced activities, such as compiling Java code, generating static wrappers, and running Ant scripts, users must use Eclipse.

Working with Projects in OpenL Tablets WebStudio

OpenL Tablets WebStudio is intended for a multi-user environment. It provides a centralized storage of rule projects called **rules repository**. Rules repository is stored on the OpenL Tablets WebStudio server and can be accessed by all users. However, users cannot modify projects directly in rules repository. Instead, to make modifications to a project, users must execute the following procedure:

Procedure for modifying a project		
Step	Action	Description
1	Check out a project.	Checking out a project from rules repository creates a copy in user's workspace, a specific location on the OpenL Tablets WebStudio server. Working copies of projects checked out by the particular user are stored here. Users can only access their personal workspaces. A checked out project is locked in rules repository to avoid loss of information. Other users cannot check it out until the project is checked in. Other users can only open checked out projects in read only mode.
2	Modify a project.	Modifications to a checked out project are performed on the working copy stored in user's workspace. Modifications are not immediately visible to other users.

Procedure for modifying a project		
Step	Action	Description
3	Check in a project.	<p>Checking in a project copies user's workspace modified copy to rules repository. A new version of the project is created in rules repository. A project can be restored to any of its previous versions.</p> <p>From this point, changes are visible to other users and the project is available for check out.</p>

In addition to checking out and checking in projects, users can also open and close them. An open project is copied from rules repository to user's workspace, but the user cannot modify its contents. If a user only wants to view contents of a project, opening the project is recommended instead of checking it out. A checked out project is locked for editing by other users.

Closing a project deletes it from user's workspace but does not affect the version in rules repository. Closed projects can be browsed in repository editor but are not available in rule editor.

OpenL Tablets WebStudio Components

OpenL Tablets WebStudio consists of the following main components:

OpenL Tablets WebStudio components	
Component	Description
Rule editor	<p>Graphic user interface running in a web browser allowing users to browse rule modules, modify table data, and run tests.</p> <p>Rule editor is the default user interface displayed when user opens OpenL Tablets WebStudio.</p> <p>Rule editor does not display all rule module files but provides a logical view of rules stored in a module. This view is convenient for users who modify business rules.</p> <p>Rule editor displays only modules available in projects stored in user's workspace. To retrieve a project to user's workspace, the project must be opened or checked out. For information on opening and checking out projects, see Working with Projects in OpenL Tablets WebStudio.</p> <p>For detailed information on using rule editor, see Chapter 3: Using Rule Editor.</p>
Repository editor	<p>Graphic user interface running in a web browser allowing users to browse and manage projects in rules repository.</p> <p>Unlike rule editor, repository editor displays physical contents of rule projects.</p> <p>Users can easily switch between rule editor and repository editor in user interface.</p> <p>Repository editor provides the following main functions:</p> <ul style="list-style-type: none"> • uploading projects from the file system to rules repository • checking out, checking in, opening, and closing projects • modifying project structure and properties • managing project versions and dependencies • copying and deleting projects in rules repository • managing and tracing project deployments <p>For detailed information on using repository editor, see Chapter 5: Using Repository Editor.</p>

OpenL Tablets WebStudio components	
Component	Description
Rules repository	<p>Centralized storage of rule projects accessible by all OpenL Tablets WebStudio users. Projects uploaded to rules repository are visible to other users.</p> <p>Rules repository creates a separate project version each time a project is checked in. A project can be restored to any of its previous versions if it is checked in with incorrect data.</p>
Deployments repository	<p>Centralized storage of final rule projects delivered to the production environment where solution applications use them.</p> <p>Projects can be deployed to deployments repository from rules repository using deployment projects. A deployment project is a specific OpenL Tablets WebStudio project type. It stipulates which rule projects and project versions must be deployed to deployments repository. Deployment projects are saved and versioned so that developers can identify which specific rule project versions are deployed.</p>
User workspace	<p>Project storage on the server containing projects checked out by users. Each user has a personal workspace not accessible by others.</p>

Security Overview

OpenL Tablets WebStudio supports a security mechanism restricting access to certain product functions based on user access rights. Each OpenL Tablets WebStudio user is identified by a unique name. Users can have varied levels of access in OpenL Tablets WebStudio. For example, system administrators usually have full access to all OpenL Tablets WebStudio functions, whereas business users only have access rights to modify business rules.

Usually, when a user opens OpenL Tablets WebStudio in the web browser, the user is automatically logged in using the Windows® account. If automatic logging in is not supported, a login window appears and the user name and password must be specified.

Chapter 2: Getting Started

This section explains logging in to OpenL Tablets WebStudio and briefly introduces the user interface.

The following topics are included in this section:

- [Logging In to OpenL Tablets WebStudio](#)
- [Understanding the User Interface](#)

Logging In to OpenL Tablets WebStudio

To log in to OpenL Tablets WebStudio, proceed as follows:

1. In the web browser address bar, enter the OpenL Tablets WebStudio URL provided by the system administrator.

The OpenL Tablets WebStudio URL has the following pattern:

`http://<server>:<port>/webstudio`

Usually, the user is automatically logged in using the Windows account. However, depending on the solution configuration, the login window can appear.



Figure 1: Login window

2. If the login window appears, enter your user name and password provided by the system administrator and click **Login**.

Understanding the User Interface

The OpenL Tablets WebStudio user interface consists of the following main parts:

- [Rule Editor](#)
- [Repository Editor](#)

Rule Editor

This section briefly introduces rule editor. For detailed information on tasks that can be performed in rule editor, see [Chapter 3: Using Rule Editor](#).

The following topics are included in this section:

- [Rule Editor Overview](#)
- [View Modes](#)

Rule Editor Overview

Rule editor provides controls for users to browse rule modules and modify table data. This is the default editor opened when a user logs in.

Rule editor resembles the following:

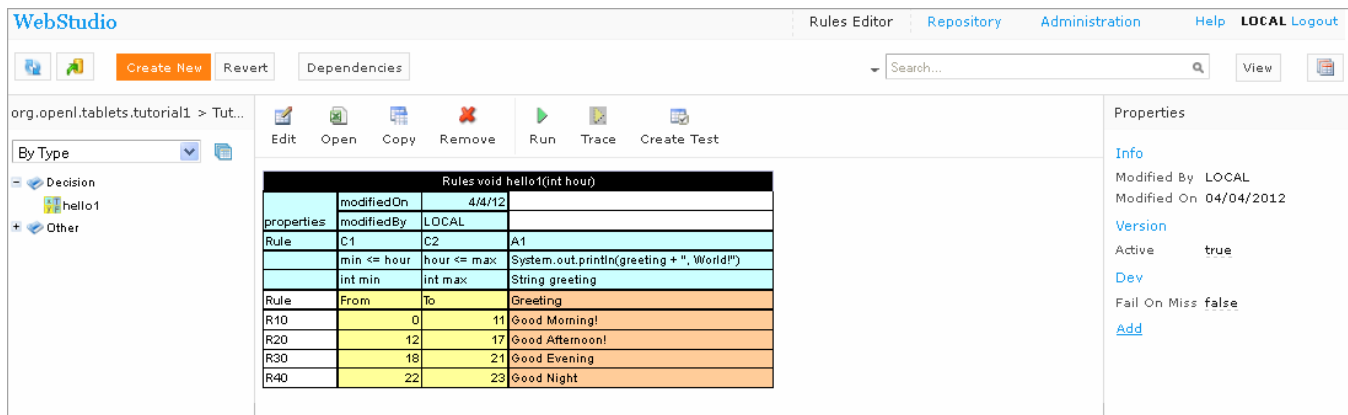


Figure 2: OpenL Tablets WebStudio rule editor








Rule editor displays one module at a time. To switch between modules, the user must select a module in the **Projects** tree. One rule project can contain several modules.








The left pane displays the module tree providing a view of elements in the currently displayed rule module.

The middle pane displays contents of the table selected in the left pane and provides controls for modifying table data, running tests, and checking test results.

The right pane displays properties of the currently displayed table.

The upper part of the window contains toolbars with the following controls:

Rule editor toolbar buttons	
Button	Description
	Switches to view with table headers and/or Excel formulas. For information on view modes, see View Modes .
	Opens a page for reverting project changes
	Opens the search window. For information on performing searches, see Performing a Search .
	Refreshes OpenL Tablets WebStudio with latest changes in Excel files.
	Initiates the table creation wizard.
	Displays recently visited tables instead of the module tree.
	Returns to the module tree view.

Rule editor toolbar buttons	
Button	Description
	Initiates a dialog for comparing Excel files.
	Opens Dependency graph
	Switches user interface to repository editor. For general information on repository editor, see Repository Editor .
	Opens OpenL Tablets WebStudio help.
	Logs the user out of OpenL Tablets WebStudio.
	Switches user interface to rules editor For general information on rules editor, see Rule Editor
	Switches user interface to the Administration mode

View Modes

OpenL Tablets WebStudio provides the following display modes for showing rule elements:

Project display modes in rule editor	
Mode	Description
Simple view	Project view is more business oriented displaying only those project elements relevant to a business user. Structure of the tree is logical rather than physical. Rule tables are organized into categories based on the Category table property or, if the property is not defined, based on Excel table sheet names. An example of a module tree displayed in simple view sorted by the Category parameter is as follows:

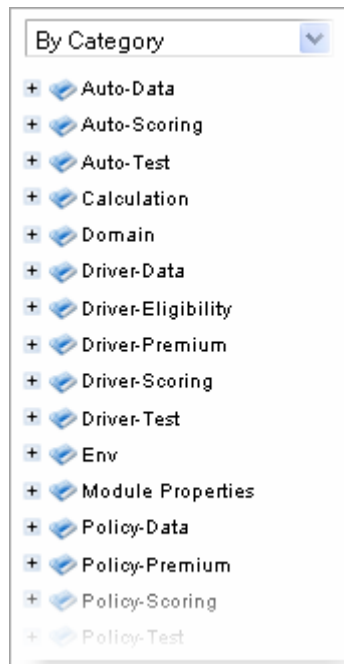


Figure 3: Module tree sorted by category

The **By Category Detailed** view displays modules sorted by the first value of the Category property. In the example the same module tree is sorted by **Category Detailed**. The modules that have the Auto-Data category are shown in the **Auto** node and **Data** sub-node. The modules with, for example, Calculation category value, are displayed in the **Calculation** node, **Calculation** sub-node:

Project display modes in rule editor	
Mode	Description

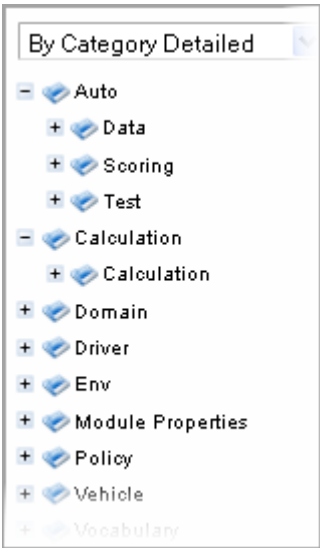


Figure 4:Module tree sorted by Category Detailed

An example below provides the module tree sorted by **Category Inversed** (where the modules are sorted by the second value of the **Category** property):

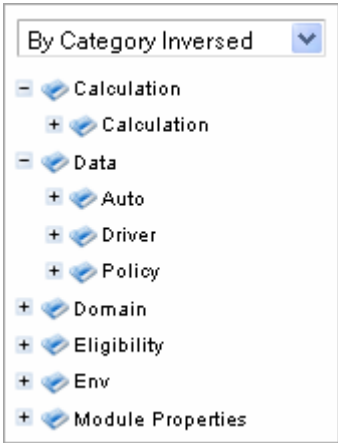


Figure 5:Module tree sorted by Category Inversed

Note: If in a Properties table 'scope' is defined as 'Module' then in the **By Category** view this table is shown in the **Module Properties** sub-node (see illustration above). If the 'scope' is defined as 'Category' then the table will be displayed in the Category **Properties** sub-node.

OpenL Tablets WebStudio hides various technical table details when a table is opened in simple view. The following is an example of a table opened in simple view:

Project display modes in rule editor

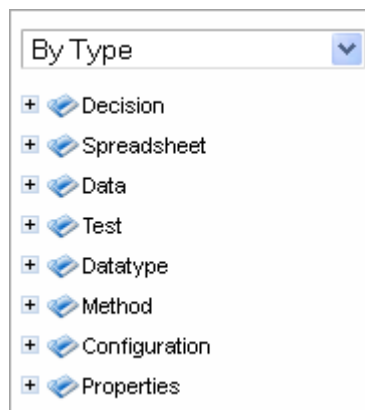
Mode	Description
------	-------------

Vehicle Age	Increase
<1	\$400
1-4	\$300
5-10	\$250

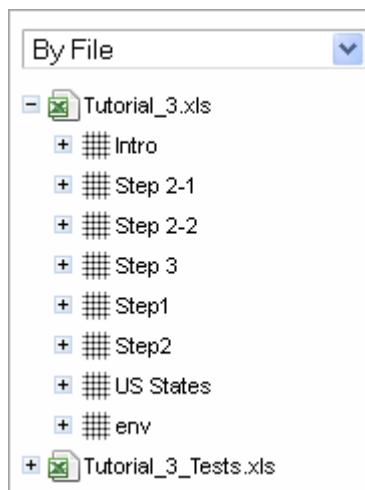
Figure 6: Rule table in simple view

The user can switch to the simple view by clicking the **View** button and deselecting the **Header** check box.

Extended view Project is displayed in a way convenient to experienced users with module tree elements organized by type rather than logically. The following is an example of a module tree displayed in extended view and sorted by type:

*Figure 7: Module tree sorted by type*

The following module tree is sorted **By File** and by spreadsheets within the file:

*Figure 8: Module tree sorted by file*

OpenL Tablets WebStudio shows various technical table details important for integration with code when a table is opened in extended view. The following is an example of a table opened in

Project display modes in rule editor**Mode** **Description**

extended view:

Rules DoubleValue ageSurcharge(Vehicle vehicle)		
properties	name	Vehicle Age Surcharge
C1		RET1
ageRange.contains(vehicle.age)		ageSurcharge
IntRange ageRange	DoubleValue ageSurcharge	
Vehicle Age		Increase
<1		\$400
1-4		\$300
5-10		\$250

Figure 9: Rule table in extended view

User can switch to extended view by clicking the **View** button and selecting the **Header** check box.

Repository Editor

Repository editor provides controls for browsing and managing design time repository. User can switch to repository editor by clicking the **Repository** control. Repository editor resembles the following:

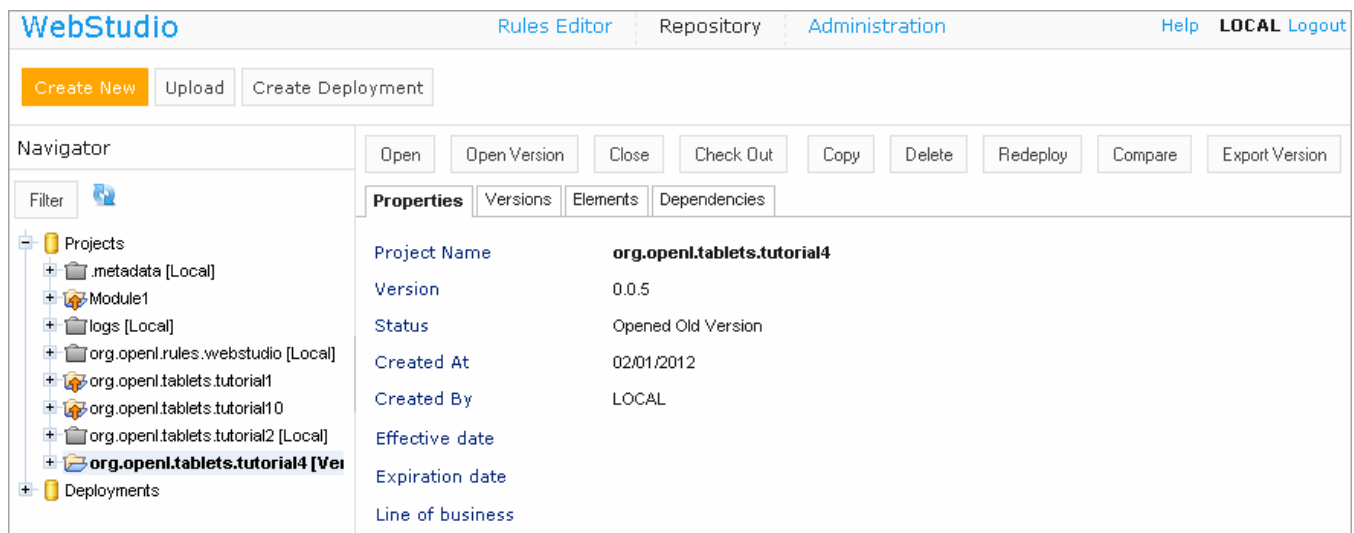


Figure 10: OpenL Tablets WebStudio repository editor

The left pane contains a tree of projects stored in design time repository and user's workspace. Unlike rule editor, repository editor displays physical project contents in terms of files and folders.

The right pane of repository editor differs depending on the element selected in the tree.

The user can switch to rule editor by clicking the **Rules Editor** control.

For detailed information on tasks that can be performed in repository editor, see [Chapter 5: Using Repository Editor](#).

Chapter 3: Using Rule Editor

This section describes the basic tasks that can be performed in rule editor. For general information on rule editor, see [Rule Editor](#).

The following topics are included in this section:

- [Filtering Projects](#)
- [Opening a Module](#)
- [Managing Projects](#)
- [Viewing Tables](#)
- [Modifying Tables](#)
- [Performing a Search](#)
- [Creating New Table](#)

Filtering Projects

The system enables you to limit a list of projects to be displayed in the **Projects** list. For that, you should start typing a project name in the field located above the list of projects.

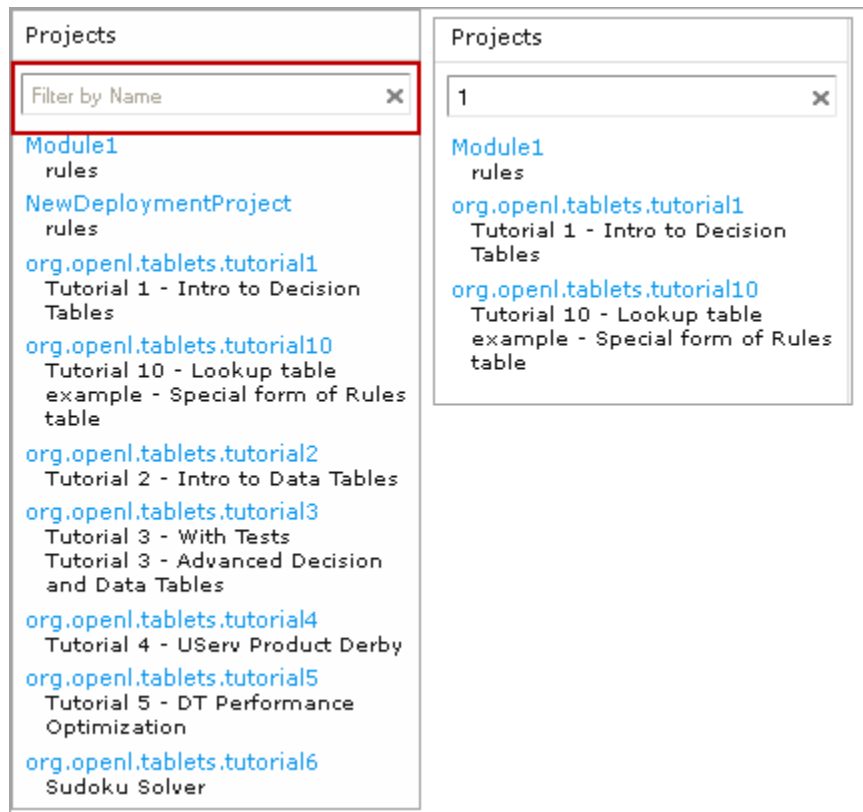


Figure 11: Filtering projects by Name

To get a full list of project, click the cross on the right.

Opening a Module

Rule editor allows a user to work with one module at a time. To select a module, in the **Projects** tree select the module name. Selected module appears as a tree in the left pane displaying its tables. If a particular module is not available, the project in which it is defined must be opened. For information on opening a project, see [Opening a Project](#).

Managing Projects

This section explains the following tasks that can be performed on projects in rule editor:

- [Checking Out and Checking In a Project](#)
- [Uploading Projects to Design Time Repository](#)
- [Reverting Project Changes](#)

Checking Out and Checking In a Project

A project can be checked out and checked in directly in rule editor.

NOTE: It is impossible to check out a local project in the Rules Editor.

To check out an open project, above the module tree, click **Check Out Project** . If the project is checked out, to check it in, click **Check In Project** .

Reverting Project Changes

OpenL Tablets WebStudio provides functionality allowing users to compare or revert project changes against a specific date.

To compare project versions, select the desired project in the **Projects** tree and follow the steps below.

1. Click the **Revert** button in the top line menu. The **Revert Changes** page appears where all the project versions are displayed.

Revert Changes			
#	Modified On	Tutorial_2.xls	Compare
1	Initial	✓	<input type="checkbox"/>
2	07/05/2012 at 01:13:17 PM	✓	<input type="checkbox"/>
3	07/06/2012 at 11:43:37 AM	✓	<input type="checkbox"/>
4	07/06/2012 at 11:58:30 AM	✓	<input type="checkbox"/>
5	07/06/2012 at 12:39:12 PM	✓	<input type="checkbox"/>
6	07/06/2012 at 12:42:11 PM	✓	<input type="checkbox"/>
7	07/06/2012 at 12:46:26 PM	✓	<input type="checkbox"/>
8	07/09/2012 at 12:36:27 PM	✓	<input type="checkbox"/>
9	07/09/2012 at 08:43:50 PM	✓	<input type="checkbox"/>

Figure 12: Compare / Revert project changes – Step 1

2. To compare the changes, select checkboxes for the required dates and click the **Compare** button.

Revert Changes			
#	Modified On	Tutorial_2.xls	Compare
1	Initial	✓	<input type="checkbox"/>
2	07/05/2012 at 01:13:17 PM	✓	<input type="checkbox"/>
3	07/06/2012 at 11:43:37 AM	✓	<input checked="" type="checkbox"/>
4	07/06/2012 at 11:58:30 AM	✓	<input checked="" type="checkbox"/>
5	07/06/2012 at 12:39:12 PM	✓	<input type="checkbox"/>
6	07/06/2012 at 12:42:11 PM	✓	<input type="checkbox"/>
7	07/06/2012 at 12:46:26 PM	✓	<input type="checkbox"/>
8	07/09/2012 at 12:36:27 PM	✓	<input type="checkbox"/>
9	07/09/2012 at 08:43:50 PM	✓	<input type="checkbox"/>

Figure 13: Compare / Revert project changes – Step 2

The system displays the project in a separate browser window where changed table(s) are marked as shown in the example below.

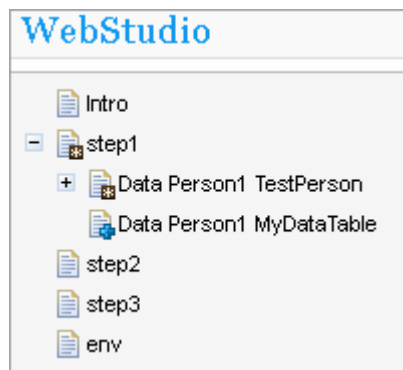


Figure 14: Table(s) with changes

3. Click the change to be viewed. The comparison result will be shown in the bottom of the window.

File 1 fragment

Data Person1 TestPerson				
modifiedOn	7/6/12			
modifiedBy	LOCAL			
createdBy	LOCAL			
createdOn	7/6/12			
name	ssn	dob	gender	maritalStatus
Name1	SSN1	DOB1	Gender1	Marital Status1
Helena	123-45-0001	1/1/1980	Female	Single
Bill	123-45-0002	1/1/1981	Male	Married

File 2 fragment

Data Person1 TestPerson				
properties	createdBy	LOCAL		
properties	createdOn	7/6/12		
name	ssn	dob	gender	maritalStatus
Name	SSN	DOB	Gender	Marital Status
Helena	123-45-0001	1/1/1980	Female	Single
Bill	123-45-0002	1/1/1981	Male	Married

© OpenL Tablets WebStudio v5.9.3, 2004-2012

Figure 15: The result of the comparison

To revert module changes:

1. Follow Step 1 and Step 2 described above.
2. Click the **Revert** button under the list of changes.
















Revert Changes				
#	Modified On	Tutorial_3.xls	Tutorial_3_Tests.xls	Compare
1	Initial		✓	<input type="checkbox"/>
2	02/23/2012 at 10:50:09 PM		✓	<input checked="" type="checkbox"/>
3	05/10/2012 at 03:16:00 PM		✓	<input checked="" type="checkbox"/>
<input type="button" value="Revert"/> <input type="button" value="Compare"/>				

Figure 16: Revert project changes

- Click **OK** in the confirmation window.

Viewing Tables

OpenL Tablets module tables are listed in the module tree. Table types are represented by different icons in rule editor. The following table describes table type icons:

Table type icons	
Icon	Table type
	Decision table.
	Decision table with unit tests.
	Column match table.
	Column match table with unit tests.
	Tbasic table.
	Tbasic table with unit tests.
	Data table.
	Datatype table.
	Method table.
	Unit test table.
	Run method table.
	Environment table.
	Property table.
	Table not corresponding to any preceding types. Such tables are considered comments.
	Spreadsheet table.

For information on each table type, see *OpenL Tablets Reference Guide*. If a table contains an error, a small red cross is displayed in the corner of the icon.









To view contents of a particular table, in the module tree, select the table. The table is displayed in the middle pane. If the project is not checked out, the table can be viewed but not modified.

Modifying Tables

OpenL Tablets WebStudio provides embedded tools for modifying table data directly in the web browser. To modify a table, proceed as follows:

1. If the project is not checked out, check it out as described in [Checking Out and Checking In a Project](#).
2. In the module tree, select the required table.

The selected table is displayed in the middle pane in the read mode.

								Available Tests/Runs Driver Age Type Test (2 test cases)
Edit	Open	Copy	Remove	Run	Trace	Test	Create Test	

Gender	Age	Status
Male	<25	Young Driver
Female	<20	Young Driver
	71+	Senior Driver
		Standard Driver

Figure 17: Table opened in OpenL Tablets WebStudio































3. If required to switch between simple / extended view, click the **View** button at the top right of the screen and select / deselect the **Header** and/or **Formula** options.
4. To switch the table to edit mode, perform one of the following steps:

- Above the table, click the **Edit** button
- Right-click anywhere in the table and click **Edit**
- Double click the cell to be edited

Note: The table cannot be switched to edit mode if the project is not checked out.

Note: Alternatively, the file can be edited in Excel. In the local mode, the rule file is opened in Excel, and changes become available in OpenL Tablets WebStudio upon Excel file saving. In the remote mode, the file must be saved locally and after modifying, uploaded via the repository.



















The table is switched to edit mode.

Gender	Age	Status
Male	<25	Young Driver
Female	<20	Young Driver
	71+	Senior Driver
		Standard Driver1

Figure 18: Table in edit mode

The edit mode provides the following buttons above the table:

Table editing buttons	
Button	Description
	Saves changes in table.
	Reverses last changes.
	Reapplies reversed changes.
	Inserts a row.
	Deletes a row.
	Inserts a column.
	Deletes a column.
	Aligns text in currently selected cell with left edge.
	Centers text in currently selected cell.
	Aligns text in currently selected cell with right edge.
	Make the text bold
	<i>Italicize</i> the cell text
	<u>Underline</u> the cell text
	<u>Set</u> fill color
	<u>Set</u> font color
	Decreases indent.
	Increases indent.
	Opens help.

5. Modify cell values as required.

A cell can be modified by double clicking it or pressing **Enter** while the cell is selected.

6. To save changes, click **Save** .

Performing a Search

OpenL Tablets WebStudio provides search functionality allowing users to perform a search across data in all module tables for a particular project.

There are the following search modes in the WebStudio application:

- [Simple Search](#)
- [Advanced Search](#)

Simple Search

Simple search looks for a particular word or phrase in all tables within a given project.

To perform a simple search, enter a word or phrase you are looking for into the **Search** field and click



Figure 19: Starting the simple search

OpenL Tablets WebStudio displays all the tables containing the entered text. Above each table there is the **Open Table in Excel** link redirecting to the Excel file containing the entered text; the **Edit Table** link opens the table in Rules Editor with the possibility to edit that table.

6 tables found

[Edit Table](#)
[Open Table in Excel](#)

Datatype Person1	
String	name
String	ssn
Date	dob
String	gender
String	maritalStatus

[Edit Table](#)
[Open Table in Excel](#)

Data Person1 pp1				
name	ssn	dob	gender	maritalStatus
Name	SSN	DOB	Gender	Marital Status
Jonh	555-55-0001	1/1/1980	Male	Single
Paul	555-55-0002	5/7/1981	Male	Single
Peter	555-55-0003	10/20/1982	Male	Single
Mary	555-55-0004	7/7/1987	Female	Married

Figure 20: Search results

You can also use simple search to look for any cell content by right-clicking a cell (the table should be open in read mode) and selecting the **Search** item in the context menu.

Advanced Search

Advanced search allows the user to narrow the search by specifying criteria for tables where the search is to be performed. You can limit the search by specifying one or more table types, text from the table header and table properties as described below.

1. To start advanced search, click the arrow to the left of search window.



Figure 21: Advanced Search – Step 1

The following filter form appears.

Figure 22: Advanced Search – Step 2

2. In the filter form, click in the **Table Types** field and select the desired type of table or mark **Select All** to search in all table types.

Figure 23: Advanced Search – Step 3

3. In the **Header contains** field, enter the word or phrase you are looking for.
4. Expand the **Table Properties** list and select the table property you need and then click the **Add** button on the right. The text field appears where you should enter the property name.
5. Enter the property name ('Driver-Scoring' in our example).

Figure 24: Advanced Search – Step 4

6. Proceed the same way to add more table properties if required. To remove the property you don't need, click the cross at the right of the property.

Figure 25: Advanced Search – remove table property

7. Click the **Search** button to run the search.

As a result, the system displays the tables matching the search criteria along with a link to the relevant Excel file and the **Edit Table** link leading to the table editing page.

[Edit Table](#)
[Open Table in Excel](#)

Rules Double\value driverTypeScore(String driverAgeType, String driverEligibility)		
properties	name	Driver Type Score Table
	category	Driver-Scoring
C1	C2	RET1
ageType == driverAgeType	eligibility == driverEligibility	ageScore
String ageType	String eligibility	Double\value ageScore
Driver Age	Eligible	Score
Young Driver	Not Eligible	30
Senior Driver	Not Eligible	20
		0

Figure 26: Advanced Search result

Creating New Table

OpenL Tablets WebStudio allows creating new tables of the following types:

- Datatype
- Datatype Alias
- Data
- Test Method
- Properties

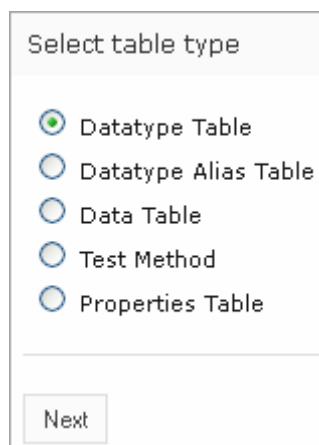
To open table wizards page, click the **Create new** button:

Create New

Datatype Table Wizard

To create new Datatype table, follow these steps:

1. Select the **Datatype Table** item as shown in picture below and press the **Next** button.



Select table type

☒ Datatype Table

☐ Datatype Alias Table

☐ Data Table

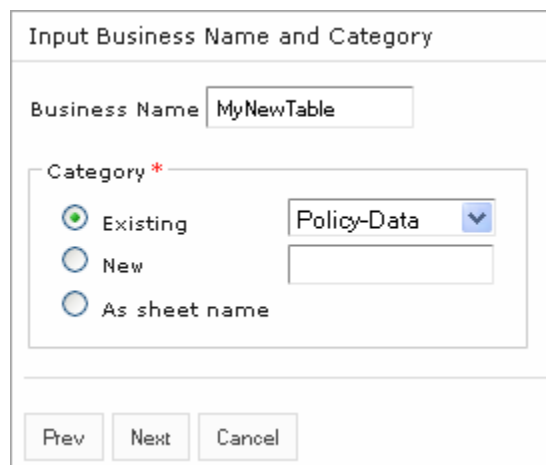
☐ Test Method

☐ Properties Table

Next

Figure 27: Table wizards

2. Input Business Name and Category



Input Business Name and Category

Business Name

Category *

☒ Existing

☐ New

☐ As sheet name

Prev Next Cancel

Figure 28: Datatype table wizard – Step 1

3. Provide name for your new type and define parent type if necessary.

Figure 29: Datatype table wizard – Step 2

4. Define type fields.

Type	Is Array	Name
Date	<input type="checkbox"/>	test1
IntRange	<input type="checkbox"/>	test2

Figure 30: Datatype table wizard – Step 3

5. Provide workbook and worksheet to save new table and click **Save** to complete.

Figure 31: Datatype table wizard – Step 4

Data Table Wizard

Creating a Data table in WebStudio is similar to creating a Datatype table described in the previous section except for several steps described in this section in details.

1. Select the **Data Table** item in and click **Next** to proceed.

Select table type

☐ Datatype Table

☐ Datatype Alias Table

☒ Data Table

☐ Test Method

☐ Properties Table

Figure 32: Data Table wizard - Step 1

2. Enter the rule table properties - **Business Name** (optional) and **Category**.

Input Business Name and Category

Business Name

Category *

☐ Existing

☐ New

☒ As sheet name

Figure 33: Data Table wizard – Step 2

3. From the **Type** list, select the table Data Type and enter the table name in the **Name** field.

Enter data table type and name

Type *

Name *

Figure 34: Data Table wizard – Step 3

4. Define the table columns configuration. In our example we create Data Table for the custom Data Type 'Person' so the screen will look similar to the next one. You can also select the **Foreign key** option and specify the table to create relation with.

Data table columns configuration

name :	Business Name =	NAME	Foreign key	<input checked="" type="checkbox"/>	TestTable
ssn :	Business Name =	SSN	Foreign key	<input type="checkbox"/>	
dob :	Business Name =	DOB			
gender :	Business Name =	GENDER	Foreign key	<input type="checkbox"/>	
maritalStatus :	Business Name =	MARITALSTATUS	Foreign key	<input type="checkbox"/>	

Figure 35: Data Table wizard – Step 4

5. Indicate where your new Data Table will be located. For that select desired Excel file from the **Workbook** list and select existing worksheet (step1 in the example) or create a new one.

Select destination

Workbook * Tutorial_2.xls

Worksheet *

☒ Existing step1
 ☐ New

Prev Save Cancel

Figure 36: Data Table wizard – Step 5

6. Click **Save** to complete your work.

From that point your new Data Table is created and you can enter data and modify the table as needed.

Chapter 4: Editing and Testing Functionality

This section provides an overview of more advanced OpenL Tablets WebStudio functions.

The following topics are included in this section:

- [Editing Tables](#)
- [Unit Tests](#)
- [Tracing](#)
- [Benchmarking](#)

Editing Tables

Editing comma array values from UI

A multi selection window, displaying all values, appears when editing the field defined as an array via commas.

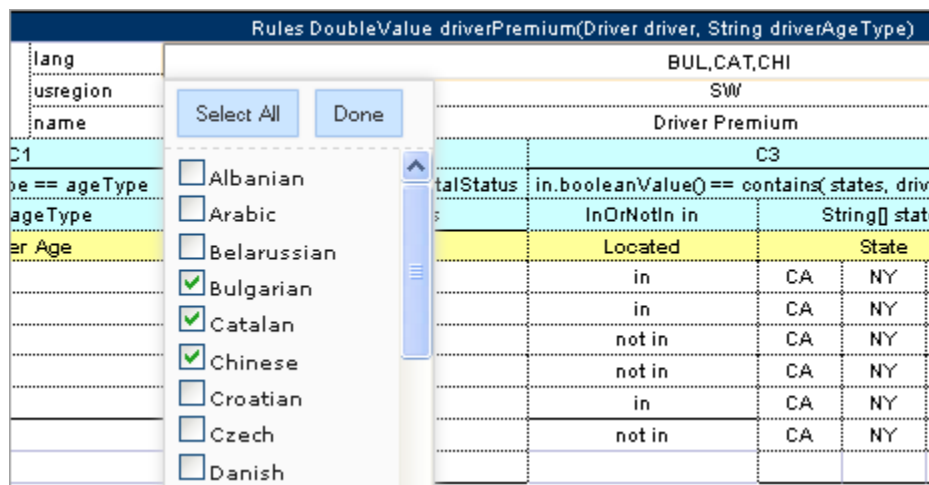


Figure 37: Multi editor in WebStudio for editing comma separated arrays

Default and Inherited Properties

If default property values are applied to the table they appear only in the **Properties** section, but not in the table source.

The screenshot shows the OpenL Tablets WebStudio interface. At the top, there is a toolbar with icons for Edit, Copy, Open in Excel, Remove, Run, Trace, Test, and Create Test. Below the toolbar, there is a section titled "Available Tests/Runs" with a link "Test Convert AM/PM to 24 hour (4 test cases)". The main area displays a table titled "Rules int ampmTo24(int ampmHr, String ampm)". The table has columns for properties, name, and result. The properties section is highlighted in yellow. The properties panel on the right shows the following details:

Name	Convert AM/PM to 24 hour
Start Request Date	02/01/2012
Active	true
Fail On Miss	false
Add	

Figure 38: Default table properties example

Module or category level properties appear in a different color in the **Properties** section and are accompanied with a link to the **Property** table where they are defined. The values are not stored in the table but are displayed in the **Properties** section, since they are inherited and applied to this table. Inherited properties can be overridden at a Table level.

The screenshot shows the OpenL Tablets WebStudio interface. At the top, there is a toolbar with icons for Edit, Copy, Open in Excel, Remove, Run, Trace, and Create Test. Below the toolbar, there is a section titled "Rules DoubleValue clientTierScore(Policy policy)". The table has columns for properties, name, and result. The properties section is highlighted in yellow. The properties panel on the right shows the following details:

Name	Client Tier Score Adjustment Table
Category	Policy-Scoring
LOB	category_Policy-Scoring_Lob
US Region	SW
Region	NCSA
Active	true
Fail On Miss	false
Add	

Figure 39: A category level properties example

System Properties

OpenL Tablets WebStudio applies system properties to each edited or created table. Their values are written in the table.

The “modifiedBy” property value is set using the currently logged in user. The “modifiedOn” property is set according to the current date. These properties are applied upon each save. These properties cannot be edited in the UI.

The “createdBy” property value is set using the currently logged in user. The “createdOn” property is set according to the current date. These properties are applied on the first save only while creating or copying a table in OpenL Tablets WebStudio.

Rules DoubleValue driverAccidentPremium(Driver driver, String driverRisk)		
properties	modifiedOn	2/22/12
	modifiedBy	LOCAL
	name	Driver Accident Premium
C1		RET1
driverRisk == risk		accidentPremium * driver.numAccidents
String risk		DoubleValue accidentPremium
Driver Risk		Per Accident Premium
		\$160

Figure 40: A system properties example

Properties for Particular Table Type

Some properties are only applicable to particular types of tables. When you open a table in OpenL Tablets WebStudio the properties section displays properties depending on the type of table. For example, such properties as 'Validate DT' or 'Fail On Miss' can be used for Decision Table as you can see on the picture below.

<a>Edit <a>Copy <a>Run <a>Trace <a>Test <a>Create Test <a>Available Tests /Runs <a>Open in Excel <a>Remove <a>Driver Eligibility Test (3 test cases)		
Rules String driverEligibility(Driver driver, String ageType)		
properties	validateDT	off
	usregion	SW
	name	Driver Eligibility Table
C1		RET1
ageType == d_ageType		hadTraining == driver.hadTraining
String d_ageType		boolean hadTraining
Driver		Eligibility
Young Driver		No Not Eligible
Senior Driver		No Not Eligible
		Eligible

Properties

Name	Driver Eligibility Table
Validate DT	true
LOB	moduleLob
US Region	SW
Region	NCSA
Active	true
Fail On Miss	false

Add

Figure 41: Properties for Decision table type

If you open a Data Table in this project, these properties will not be available in the **Properties** section.

<a>Edit <a>Copy <a>Run <a>Trace <a>Test <a>Create Test <a>Available Tests /Runs <a>Open in Excel <a>Remove <a>Driver Eligibility Test (3 test cases)		
Data Policy policyProfile2		
properties	name	Policies Set 2
	category	Policy-Data
name		Policy
drivers	>driverProfiles1	Drivers
vehicles	>autoProfiles2	Vehicles
clientTier		Client Tier
clientTerm		Client Term

Properties

Name	Policies Set 2
Category	Policy-Data
LOB	moduleLob
US Region	SW
Region	NCSA

Add

Figure 42: There are no properties on Data table that are defined for Decision table only

When doing action 'Copy', properties not suitable for current table type, don't appear in the wizard.

To add a new property for the selected table, click the **Add** link in the **Properties** pane, select the desired property, click the **Add** button, specify the property value, and then click the **Save** button to complete. All the steps are collected in the Figure below.

Figure 43: Adding a new property for the current table

Copy as New Version

The table versioning mechanism is based on copying the existing table and is initiated in OpenL Tablets WebStudio by clicking the **Copy** button. Then you should select **New Version** in the **Copy as** list, enter the data as needed and click **Copy** to save your work.

A new table version has the same identity, that is, signature and dimensional properties of the previous version. When a new table version is created, the previous version becomes inactive, as only one table version can be active at a time. By default, all tables are active. An example of an inactive table version follows:

Rules DoubleValue driverRiskScoreOverloadTest(String driverRisk)		
properties	name	Driver Risk Score Table
	category	Driver-Scoring
	version	0.0.1
	active	FALSE
C1		RET1
risk == driverRisk		score
String risk		DoubleValue score
Driver Risk		Score
High Risk Driver		100
		0

Figure 44: An inactive table version

Versions of the same table are grouped in the module tree under the table name. Clicking the table name displays the active version. If all tables are set to inactive, the latest created version is displayed.

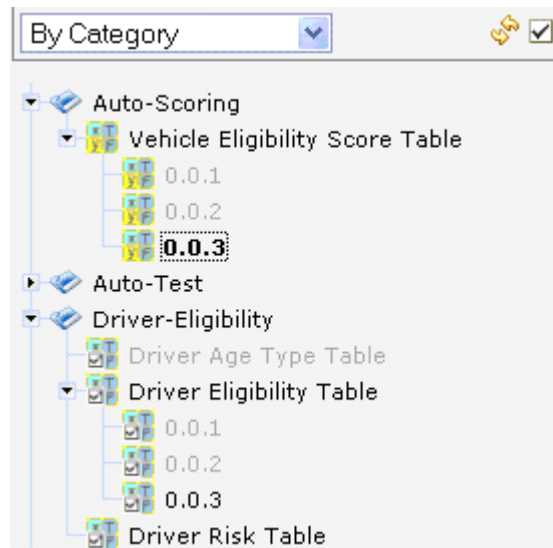


Figure 45: Displaying table versions in the module tree

The table version is defined in three digit format, such as 4.0.1. Table versions must be set in an increasing order.

Figure 46: Entering a new version number

Unit Tests

Unit tests are used in OpenL Tablets to validate data accuracy. OpenL methods with predefined input data compare the test results with expected results. Every decision table can be accessed as an OpenL method. The method signature is included in the header of a decision table. Each unit test is stored in a separate table.

For example, in the following diagram, the table on the left is a decision table but the table on the right is a unit test table that tests data of the decision table:

Rules int ampmTo24(int ampmHr, String ampm)			Testmethod ampmTo24 ampmTo24Test		
C1	C2	RET1	ampmHr	ampm	_res_
range.contains	suffix.equals	result	Hour	AM/PM	24 Hr
IntRange range	String suffix	int result			
AM/PM hour	AM or PM	24 hour			
12	AM	0	3	AM	3
1-11	AM	=ampmHr	12	AM	0
12	PM	12	12	PM	12
1-11	PM	=ampmHr+12	3	PM	15

Figure 47: Decision table and its unit test table

OpenL Tablets WebStudio supports visual controls for creating and running project unit tests. Unit test tables can be modified like all other tables in OpenL Tablets WebStudio. For information on modifying a table, see [Modifying Tables](#). Test results are displayed in a simple format directly in the user interface.

Navigation

WebStudio adds to table view navigation link to appropriate test table and vice versa. See below.

Edit

Copy

Run

Trace

Benchmark

Open in Excel

Remove

Target Table

[Driver Age Type Table](#)

Properties

Name	Driver Age Type Test
LOB	moduleLob
US Region	SW
Region	NCSA
Add	

Testmethod driverAgeType driverAgeTypeTest		
properties	name	Driver Age Type Test
driver		_res_
>testDrivers1		
Driver		Expected Age Type
Sara		Standard Driver
Spencer, Sara's Son		Young Driver

Figure 48: Navigation link to target table

Edit

Copy

Open in Excel

Remove

Run

Trace

Test

Create Test

Available Tests/Runs

[Driver Age Type Test \(2 test cases\)](#)

Properties

Name	Driver Age Type Table
LOB	moduleLob
US Region	SW
Region	NCSA
Active	true
Fail On Miss	false
	Add

Gender	Age	Status
Male	<25	Young Driver
Female	<20	Young Driver
	71+	Senior Driver
		Standard Driver

Figure 49: Navigation links to available checks and tests

Run Tests

This section provides the methods used to run unit tests.

Execute all project tests at once

The system automatically executes all test runs in every unit test in project and displays a summary of results.

To run all project tests, click the **Run All Tests** ☒ icon above the module tree in rule editor.

Test results resemble the following:

Number of Tests: 5 (1 failed)			
<u>Driver Risk Test (3 test cases)</u>			
Driver	Expected	Result	
+ Driver	Standard Risk Driver	✓	Standard Risk Driver
+ Driver	High Risk Driver	✗	Standard Risk Driver
+ Driver	High Risk Driver	✓	High Risk Driver
<u>Driver Age Type Test (2 test cases)</u>			
Driver	Expected	Result	
+ Driver	Standard Driver	✓	Standard Driver
+ Driver	Young Driver	✓	Young Driver
<u>Driver Eligibility Test (3 test cases)</u>			
Driver	Age	Expected	Result
+ Driver	Standard Driver	Eligible	✓ Eligible
+ Driver	Young Driver	Eligible	✓ Eligible
+ Driver	Young Driver	Not Eligible	✓ Not Eligible

Figure 50: Results of running all project tests

Failed tests are represented by ✗ mark. Passed tests are represented by ☒ mark.

Execute tests for a single table

To execute all test runs for one particular rule table, in rule editor, in the module tree, select the rule table and, in the upper part of the middle pane, click **Test** ☒.

Test results resemble the following:


Results of running [ampmTo24](#)

ampmTo24Test 4 test cases

Hour	AM/PM	Expected	Result
3	AM	3	✓ 3
12	AM	0	✓ 0
12	PM	12	✓ 12
3	PM	15	✓ 15

Figure 51: Results of executing all test runs for one rule table

You can also test a rule table even if no tests have been created for the given table yet. For that, follow the steps below.

1. Select the desired rule table in the module tree and click the green **Run** arrow  above the table. The form appears where you shall enter required values:

ampmHr	ampm
 ampmHr = <input type="text" value="3"/>	 ampm = <input type="text" value="PM"/>
<input type="button" value="Run"/>	

Figure 52: Testing a rule table without tests

2. Click the **Run** button within the popup. The results of the testing are displayed:

Results of running [Virtual test suite](#)

ampmHr	ampm	Result
3	PM	15

Figure 53: Result of running virtual test

For Test tables, the system enables you to select test case(s) to be executed.

1. Navigate to the **Run** button above the Test table and click the small black arrow:

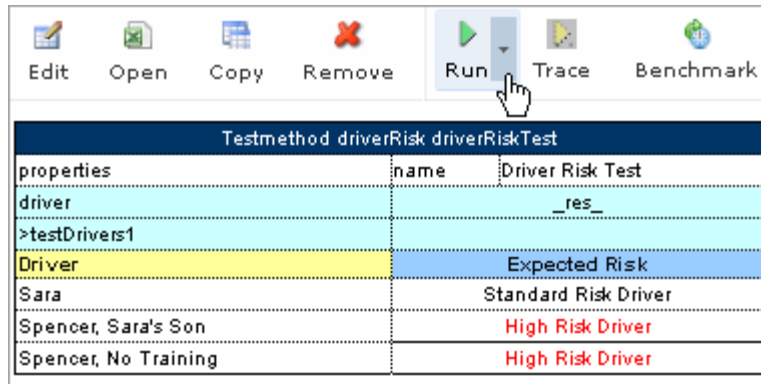


Figure 54: Running test cases with selection for a Test table

- The popup appears where you can choose from the test cases; select or clear the check-boxes as you need:

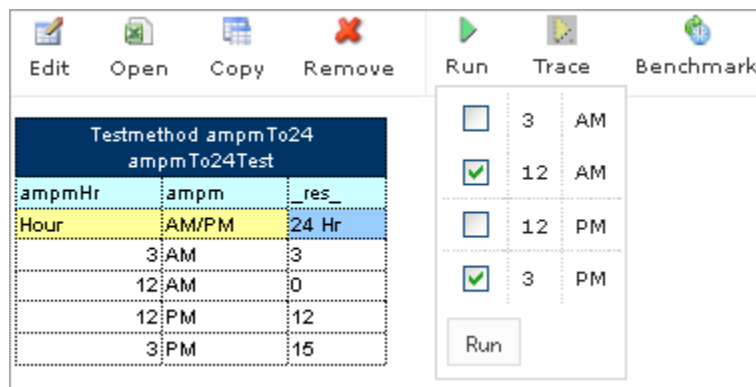


Figure 55: Select test cases to be executed

- Click the **Run** button within the popup. Only selected test cases will be executed:

ampmTo24Test 4 test cases			
Hour	AM/PM	Expected	Result
12	AM	0	✓ 0
3	PM	15	✓ 15

Figure 56: Result of selective testing

Creating New Test

WebStudio provides convenient way to create a new test table.

After you has created executable table (Decision, Method, Test, Run, Spreadsheet) the **Create Test** item will be available.

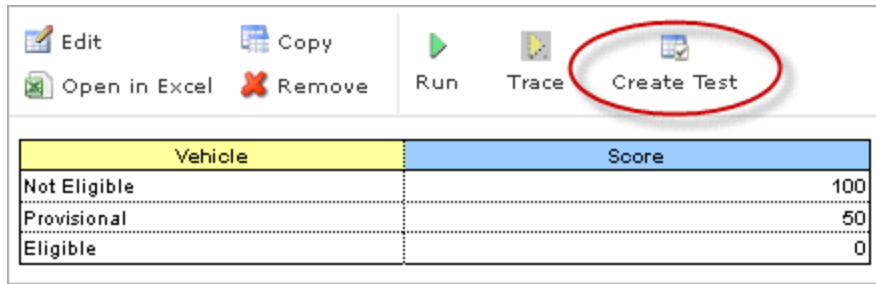


Figure 57: Create new test table

Click the **Create Test** link button to create Test table for current table. WebStudio proposes you the two step wizard which helps to create appropriate Test table.

Tracing


OpenL Tablets WebStudio provides a rule tracing view for all appropriate OpenL Tablets methods. These methods include the following:

- All unit tests
- Decision tables and method tables with attached `Runmethod` data

Rule tracing enables users to determine how results for complex rules are obtained.

IMPORTANT! Make sure your browser does not block pop-up windows. Otherwise, you will not see the trace results. See information on how to make this setting in your browser Help.

Follow the steps below to display the trace view for a **rule table**.

1. In Rules editor, open a rule table you want to trace and click **Trace**  in the middle pane.
2. Enter parameters to be traced in the popup window:

The screenshot shows a popup window titled 'driver'. It contains a tree view with 'driver = Driver' expanded. Below the tree are input fields for various parameters:

- name =
- gender =
- age =
- maritalStatus =
- state =
- numAccidents =
- numMovingViolations =
- numDUI =
- hadTraining =

At the bottom, there are two buttons: 'Trace' and 'Trace into File'. A mouse cursor is pointing at the 'Trace' button.

Figure 58: Tracing a rule for a rule table

3. Click the **Trace** button.

To trace a **Test table**:

1. For a Test table, navigate to the **Trace** button above the Test table and click the small right-hand black arrow to open a popup with test cases to be traced.

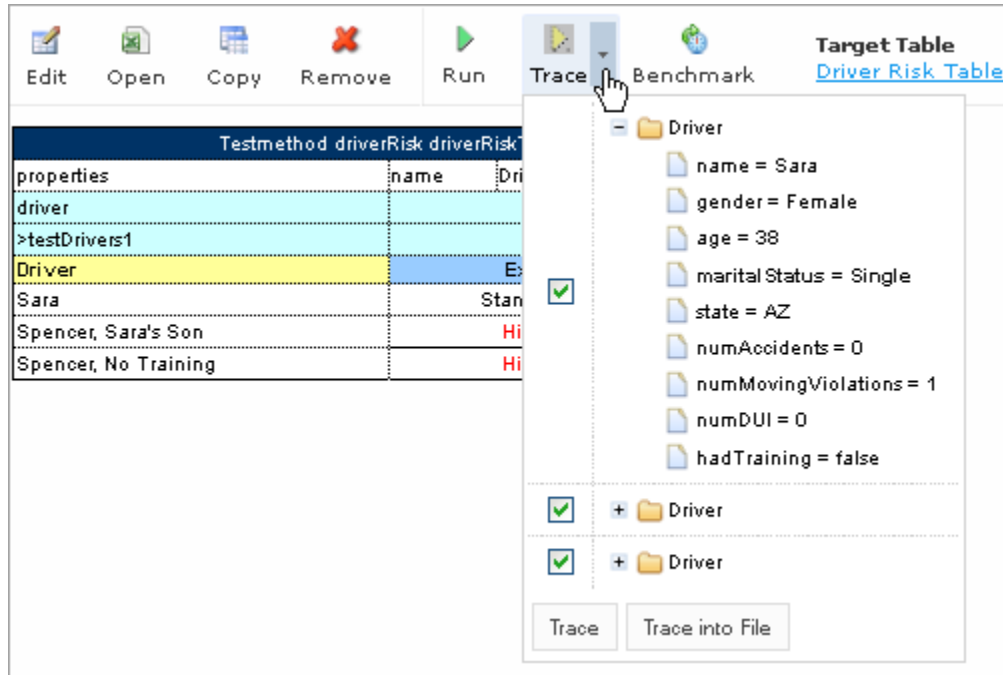


Figure 59: Run tracing for a Test table

2. In the popup, select or clear the test cases as you need. By default, all the cases are selected.
3. Click **Trace** to start the process. The system displays the tracing results in a separate browser window as illustrated below.

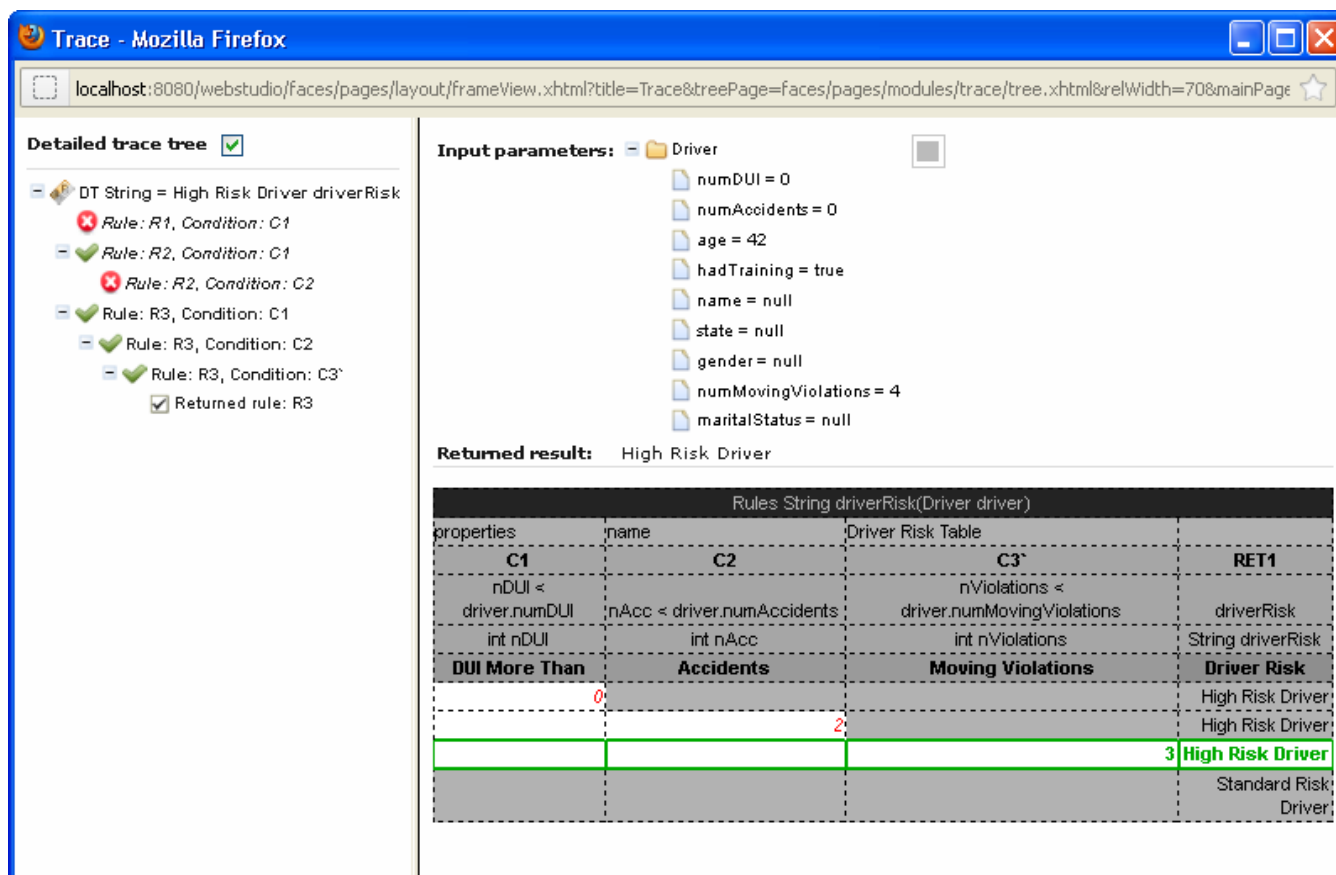


Figure 60: Tracing results

The left side displays a tree consisting of rule table(s) as tree nodes and fired rule rows as tree leaves. Selected **Detailed trace tree** option enables you to view all method calls. If you clear that option only successful calls will be displayed.

If an element in the tree is selected, the corresponding rule table is displayed in the right pane. The fired rule rows are highlighted using the specified color. The highlight color and transparency level can be configured by clicking the ☐ button above the rule table (the button is gray by default).

In addition the right pane displays the actual parameters used in the particular method call and the returned result.

The example above demonstrates the results of tracing Decision table. For other rule tables, the picture differs a bit but the meaning is essentially the same.

For a Decision table the tracing results are shown as follows:

- The rules that were traced are not highlighted (white rows)
- Successfully completed (returned) rules are boxed with green lines
- The failed rules are displayed in red

Benchmarking

OpenL Tablets WebStudio provides benchmarking tools for measuring execution time for all appropriate OpenL Tablets elements. In OpenL Tablets, everything that can be run can be benchmarked too. Benchmarking is useful for optimizing the rule structure and identifying critical paths in rule calculation.

The benchmarking icon is displayed above a table to be traced.

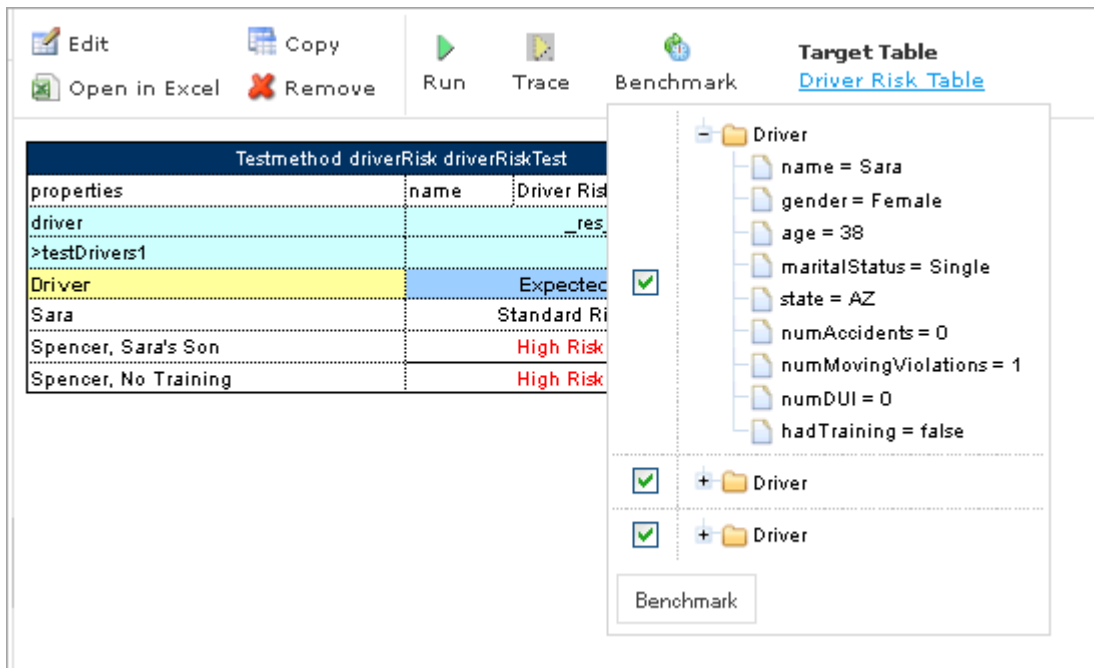


Figure 61: Controls for measuring performance

For a Test table, you can select the test cases as follows:

1. Open the desired Test table.
2. Navigate to the **Benchmark** button above the Test table and click the small right-hand black arrow to open a popup with test cases as you need.
3. Click the **Benchmark** button within the popup.

Clicking the benchmarking icon runs the corresponding method or set of methods and displays the results in a table.

Results of benchmarking Test Driver Premium

	Name	Run(ms)	Runs/sec	Unit Type	Units	Unit(ms)	Units/sec	<input type="checkbox"/>
1.	Test driverPremium	0.133289	7,502.49	Test Units	8	0.016661	60,019.96	<input checked="" type="checkbox"/>
2.	Test regionInd	0.001141	876,413.25	Test Units	4	0.000285	3,505,652.98	<input checked="" type="checkbox"/>
3.	Test driverPremium	0.125449	7,971.35	Test Units	8	0.015681	63,770.79	<input checked="" type="checkbox"/>
4.	Test regionInd	0.001089	918,615.98	Test Units	4	0.000272	3,674,463.92	<input checked="" type="checkbox"/>
5.	Test region	0.018622	53,699.22	Test Units	4	0.004656	214,796.90	<input checked="" type="checkbox"/>
Compare		Delete						

Figure 62: Benchmarking results

OpenL Tablets WebStudio remembers all benchmarking runs executed within one session. Every time a new benchmark is run, a new row is added to the results table.

Benchmarking results can be compared to identify the most time consuming methods. Select the required check boxes and click **Compare** to compare results in the results table.

Comparison results are displayed below the benchmarking table.

1.	Test driverPremium	60,019.96	5	61.22
2.	Test regionInd	3,505,652.98	2	1.05
3.	Test driverPremium	63,770.79	4	57.62
4.	Test regionInd	3,674,463.92	1	1.00
5.	Test region	214,796.90	3	17.11

Figure 63: Comparing benchmarking results

Chapter 5: Using Repository Editor

This section describes tasks that can be performed in repository editor. For general information on repository editor, see [Repository Editor](#).

The following topics are included in this section:




- [Browsing Design Time Repository](#)
- [Filtering the Project Tree](#)
- [Uploading a Project](#)
- [Creating a Project](#)
- [Opening a Project](#)
- [Closing a Project](#)
- [Checking Out a Project](#)
- [Checking In a Project](#)
- [Defining Project Dependencies](#)
- [Modifying a Project](#)
- [Copying a Project](#)
- [Removing a Project](#)
- [Deploying Projects](#)
- [Comparing Project Versions](#)





Browsing Design Time Repository

Repository editor displays all projects in user's workspace and design time repository. The project tree is organized into the following categories:

Categories in the project tree	
Category	Description
Projects	Contains OpenL Tablets rule projects.
Deployments	Contains deployment projects for deploying rule projects to production time repository. For information on using deployment projects, see Deploying Projects .

The status of each project in the tree is identified by a specific icon. The following table describes the icons in the project tree:

Project icons in repository editor	
Icon	Description
	Project is closed. It is available only in design time repository and must be opened to copy it to user's workspace.
	Project is opened. It is copied to user's workspace in read only mode and must be checked out for modification.
	Project is checked out by current user. It is copied to user's workspace and can be modified. Other users cannot check out the project. To save changes, the project must be checked in.

Project icons in repository editor	
Icon	Description
	Project is closed by current user but checked out by another user. Current user cannot check out the project.
	Project is opened by current user but checked out by another user. Current user cannot check out the project.
	Project exists only in user's workspace but not in design time repository. Other users do not see this project. User can delete the project or upload it to design time repository as described in Uploading Projects to Design Time Repository .
	Project is marked for deletion. In OpenL Tablets WebStudio, deletion of a project takes place in the following phases:
Phase	Description
Deleting a project	Project is removed from user's workspace and marked for deletion. In this phase, the project can be restored using the undelete function. For information on deleting a project, see Deleting a Project .
Erasing a project	Deleted project is permanently removed from design time repository. After this phase, the project cannot be restored. For information on erasing a project, see Erasing a Project .

Filtering the Project Tree

A file filter can be applied to the project tree so that only files of particular types are displayed.

To filter the project tree, proceed as follows:

1. Above the project tree, click **Filter**.
2. In the **Repository filter** pop up window, enter a list of file extensions, separated by semicolon as follows:
xls;properties;txt
3. Select the **Hide deleted projects** option if required.
4. Click **Apply**.

The project tree is filtered so that only files of the specified extensions are displayed. Project folders are always displayed.

Note: To reset the filter, the user must clear the previously entered file extensions and click **Apply**.

Uploading a Project

OpenL Tablets WebStudio provides controls for uploading rule projects archived in a ZIP file to design time repository.

To upload an archived rule project to design time repository:

1. Click the **Upload** button on the top left of the screen. The **Upload Project** window appears.

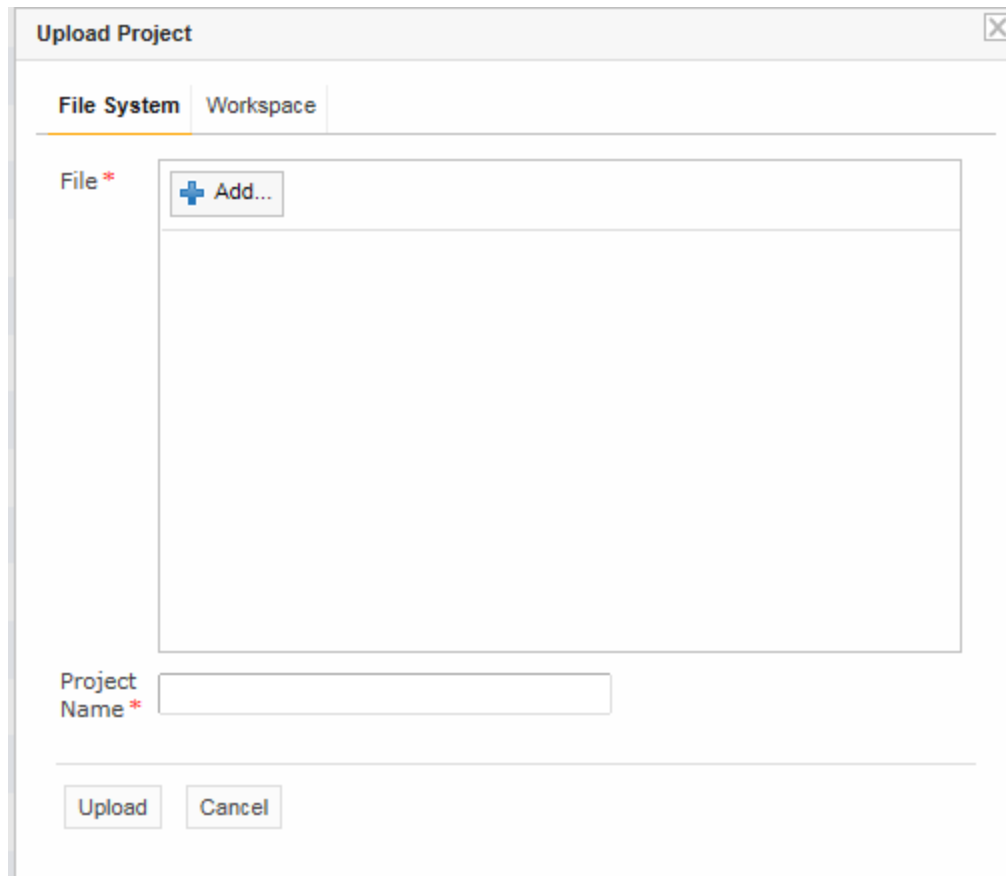


Figure 64: Uploading a project

2. Click **Add** in the **File** field and select the ZIP file containing the rule project or excel file containing OpenL tables.
3. In the **Project Name** field, enter the name by which the project must be represented in design time repository.
4. Click **Upload**.
5. To upload projects from workspace, click the **Workspace** tab, select checkboxes for projects to be uploaded and click **Upload** to complete.

Creating a Project

OpenL Tablets WebStudio allows users to create new projects in design time repository by creating folders and uploading files. A rule project is created when the user manually produces a correct rule project folder structure and uploads project files into the folders.

To create a new project, proceed as follows:

1. In the top line menu, click **Create New**.
The **New Project** window appears.

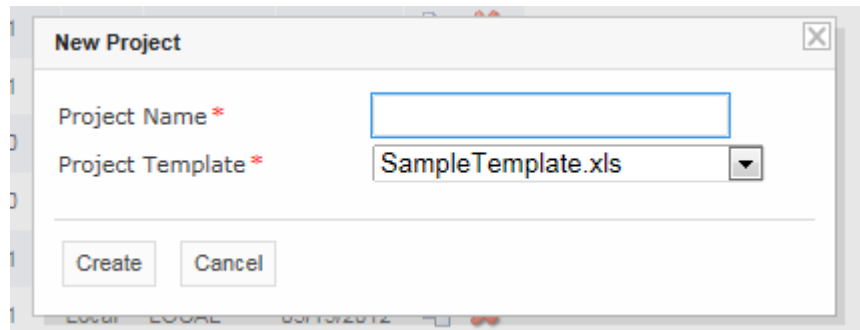


Figure 65: Creating a new project

2. In the **Project Name** field, enter the project name and click **Create**.
New project is created in design time repository. Initially, project structure corresponds to selected project template but can be constructed manually.
3. To construct the project structure, add folders and upload files as described in [Modifying Project Contents](#).

Opening a Project

An opened project is copied to user's workspace and becomes available for selection in rule editor. An opened project cannot be modified, it must be checked out as described in [Checking Out a Project](#) for modification.

To open a project, in the project tree, select the project and, in the right pane, click one of the following buttons as required:

Buttons for opening a project	
Button	Description
Open	Opens latest version of project.
Open Version	Displays window where user can specify which project version must be opened.

Closing a Project

Closing a project deletes it from user's workspace. As a result, the project is not available for selection in rule editor. Users can still browse closed projects in repository editor.

To close a project, in the project tree, select the project and, in the right pane, click **Close**.

Checking Out a Project

A checked out project is copied to user's workspace and becomes available for selection in rule editor. Only checked out projects can be modified. To apply changes made to a project, the project must be checked in as described in [Checking In a Project](#).

To check out a project, in the project tree, select the project and, in the right pane, click **Check Out**.

The latest project version is checked out even if the user previously opened an older project version.

Alternatively, an opened project can be checked out directly from rule editor as described in [Checking Out and Checking In a Project](#).

Checking In a Project

A modified project is checked in and copied from the user's workspace to design time repository as a new version.

To check in a project, proceed as follows:

1. In the project tree, select the project, and, in the right pane, click **Check In**.

The **Check In** window appears.



Figure 66: Checking in a project

The **Check In** window allows the user to specify the new version number. The **Major Version** field specifies the first of the three version numbers separated by a period. The **Minor Version** field specifies the second of the three version numbers. The third number of a version is updated automatically.

2. Specify the version numbers and click **Check In**.

A checked out project can be checked in directly from rule editor as described in [Checking Out and Checking In a Project](#).

Defining Project Dependencies

A project dependency can be defined when a particular rule project depends on contents in another project. Project dependencies are checked when projects are deployed to production time repository. OpenL Tablets WebStudio displays warning messages when a user deploys projects with conflicting dependencies.

To define a dependency on another project, proceed as follows:

1. If the project is not checked out, check it out as described in [Checking Out a Project](#).
2. In the project tree, select the project, and, in the right pane, select the **Dependencies** tab.



Figure 67: Defining dependencies

The **Dependencies** tab lists all projects required by the selected project.

3. To define a new dependency, click **Add**.

The **Add dependency** window appears.



Figure 68: Defining a new dependency

4. In the **Project** list box, select the required project.
5. In the **Lower version** list box, select the oldest allowed version of the referenced project.
6. Optionally, in the **Upper Version** field, enter the latest allowed project version.
If the **Upper Version** field is empty, any project version above the one specified in the **Lower version** field is allowed.
7. Click **Add**.
8. Repeat this procedure to add as many dependencies as required.

Modifying a Project

A project's properties and contents can be modified when it is checked out.

The following topics are included in this section:

- [Modifying Project Properties](#)
- [Modifying Project Contents](#)

Modifying Project Properties

Each rule project has a set of properties, which are displayed in the **Properties** tab when a project is selected.

Property	Value
Project Name	Module1
Version	0.0.2
Status	Checked-out
Created At	08/25/2011
Created By	LOCAL
Effective date	01/31/2011
Expiration date	12/31/2012
Line of business	Home

Figure 69: Project properties

By default, the following editable properties are displayed:

Default manually editable project properties	
Property	Description
Effective date	Starting date from which project or file is valid.
Expiration date	Expiration date after which project or file is no longer valid.
Line of business	Company branch or territory in which project or file is valid.

Some properties are updated automatically by the system, but for others, values must be entered by an administrator.

To add more properties to UI, proceed as follows:

1. Open the <tomcat directory>\webapps\webstudio\WEB-INF\conf\repository-artefact-props.properties file for editing.
2. Enter the list of attributes to add to UI, delimited by comma, as follows:
 props.use = attribute1, attribute6, attribute8, attribute11, attribute13

The following table describes types of attributes that can be added to UI:

Project properties that can be added to UI	
Property	Description
attribute 1 - attribute 5	String properties.
attribute 6 - attribute 10	Date properties.
attribute 11 - attribute 15	Number properties.

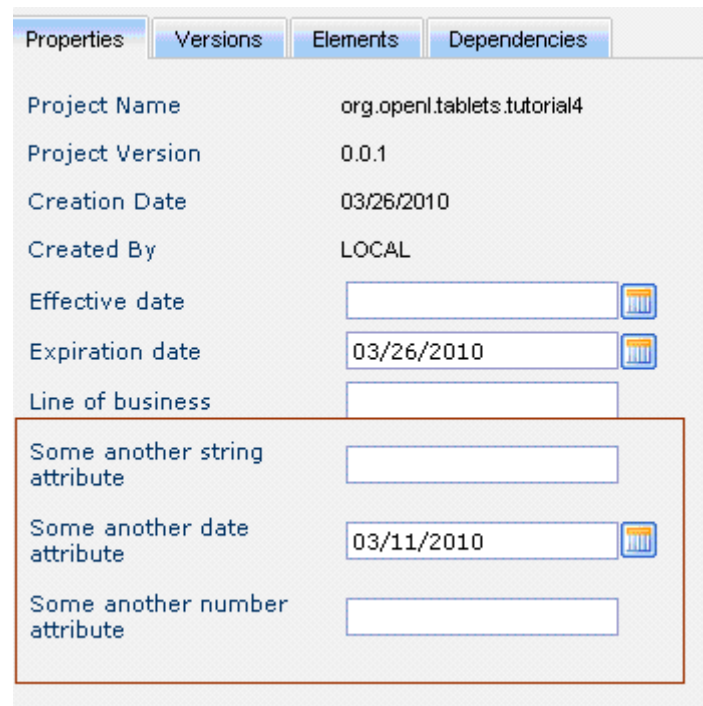
3. Define property names.

An example is as follows:

```
props.attribute1 = Attribute_Label_Name 1
props.attribute2 = Attribute_Label_Name 2
```

If a property name is not defined, the property appears in UI with its sequential number, such as **attribute 6**.

For example, if properties 1, 6, and 10 are added, they are displayed in UI as follows:



The screenshot shows a web application interface with four tabs: Properties, Versions, Elements, and Dependencies. The 'Properties' tab is active. It displays a list of properties with their values. The properties are: Project Name (org.openl.tablets.tutorial4), Project Version (0.0.1), Creation Date (03/26/2010), Created By (LOCAL), Effective date (empty text box with a calendar icon), Expiration date (03/26/2010 with a calendar icon), Line of business (empty text box), and three additional properties: 'Some another string attribute' (empty text box), 'Some another date attribute' (03/11/2010 with a calendar icon), and 'Some another number attribute' (empty text box). A red rectangular box highlights the last three properties.

Property Name	Value
Project Name	org.openl.tablets.tutorial4
Project Version	0.0.1
Creation Date	03/26/2010
Created By	LOCAL
Effective date	<input type="text"/>
Expiration date	03/26/2010
Line of business	<input type="text"/>
Some another string attribute	<input type="text"/>
Some another date attribute	03/11/2010
Some another number attribute	<input type="text"/>

Figure 70: Project UI with properties 1, 6, and 10 added

Modifying Project Contents

This section describes modifying the physical structure of a project.

The following topics are included in this section:

- [Creating a Folder](#)
- [Uploading a File](#)
- [Deleting a Folder or a File](#)

Creating a Folder

To create a new folder in the project structure, proceed as follows:

1. If the project is not checked out, check it out as described in [Checking Out a Project](#).
2. In the project tree, select the parent folder in which the new folder must be created.
To create a root level folder, the project name must be selected in the project tree.
3. In the right pane, click **Add Folder**.
4. In the **Add Folder** window, enter the folder name and click **Add**.

Uploading a File

To upload a file to a project folder proceed as follows:

1. If the project is not checked out, check it out as described in [Checking Out a Project](#).
2. In the project tree, select the folder in which the file must be uploaded.
To upload a file to the root level, the project name must be selected in the project tree.
3. In the right pane, click **Upload File**.
The **Upload File** window appears.

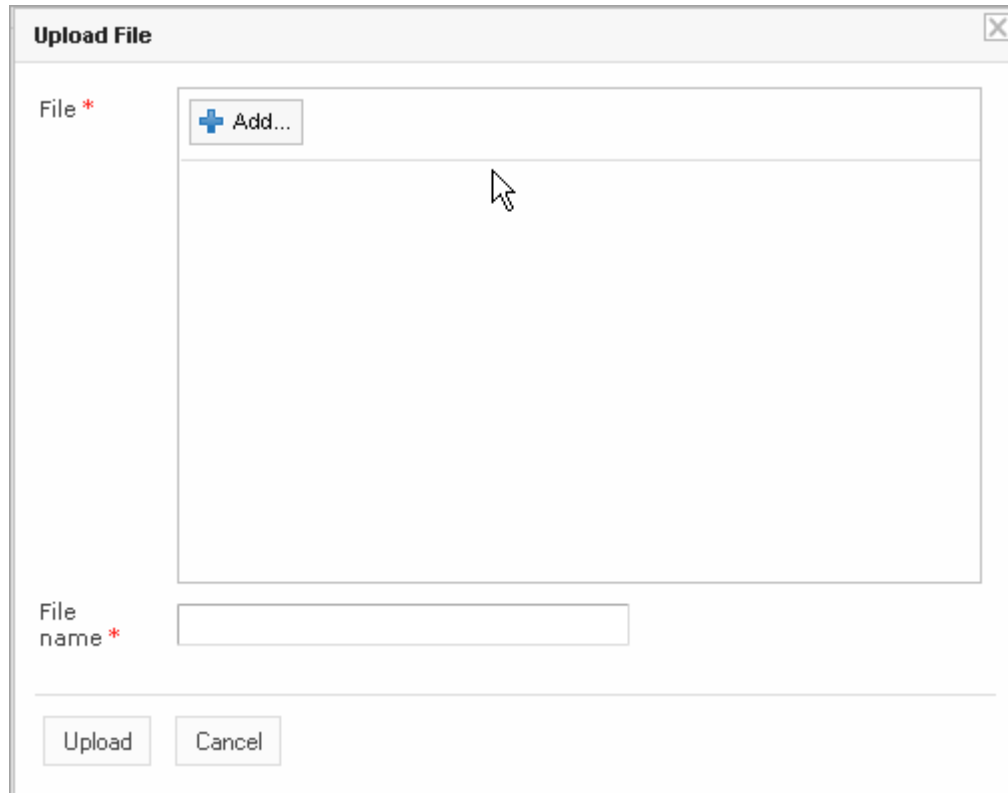


Figure 71: Uploading a file

4. Click **Add** in the **File** field and select the file to be uploaded.
5. Click the upper **Upload** button (with green arrow).
6. In the **File name** field, enter the name of the file to be used in design time repository.
7. Click Upload.

Deleting a Folder or a File


To delete a folder or a file in the project structure, proceed as follows:


1. If the project is not checked out, check it out as described in [Checking Out a Project](#).
2. Perform one of the following steps as required:
 - In the project tree, select the folder or file to be deleted and, in the right pane, click **Delete**.

Add Folder
Upload File
Delete

Properties
Versions
Elements

Folder Name bin


Effective date


Expiration date


Line of business

Save

Figure 72: Deleting a project element

- In the project tree, select the parent folder and, in the right pane, in the **Elements** tab, click **Delete** .























Properties	Versions	Elements	Dependencies
Type	Name	Actions	
	META-INF		
	bin		
	build		
	doc		
	gen		
	rules		
	src		
	.classpath		
	.project		
	Generate org.openl.tablets.tutorial4 Wrapper.launch		
	build.properties		

Figure 73: Deleting project elements in the Elements tab


A confirmation window appears.

- In the confirmation window, click **OK**.

Copying a Project

Copying a project creates a new project with identical contents and a different name in design time repository. This function can be used for copying local projects to design time repository with different name.

To copy a project, proceed as follows:

1. Perform one of the following steps as required:
 - In the project tree, select the project and, in the right pane, click **Copy**.
 - In the project tree, select **Projects** and, in the right pane, next to the project name, click **Copy** .
2. In the **Copy Project** window, enter the new project name and click **Copy**.

Removing a Project


Removing a project is executed in the following phases:

- [Deleting a Project](#)
- [Erasing a Project](#)

Deleting a Project

A deleted project is removed from user's workspace and marked as deleted in design time repository. All users can see that a project is deleted. Physically, it still remains in design time repository.

To delete a project, proceed as follows:

1. Perform one of the following steps as required:
 - In the project tree, select the project and, in the right pane, click **Delete**.
 - In the project tree, select **Rules Projects** and, in the right pane, next to the project name, click **Delete** .
2. In the confirmation window, click **Delete**.

Deleted projects can be restored by using the **Undelete** button.

To make deleted projects visible uncheck 'Hide deleted project' checkbox on the filter pop up window and click 'Apply'.

Erasing a Project

Erasing a project permanently removes it from design time repository.

Warning: Erased projects cannot be restored.

To erase a project, proceed as follows:

1. Delete the project as described in [Deleting a Project](#).

2. In the project tree, select the project and, in the right pane, click **Erase**.
3. In the confirmation window, click **Erase**.

Deploying Projects

This section describes tasks related to deploying rule projects to production time repository.

The following topics are included in this section:

- [Creating a Deployment Project](#)
- [Defining Deployment Project Descriptors](#)
- [Deploying a Deployment Project](#)
- [Opening Deployed Projects](#)
- [Redeploying Projects](#)

Creating a Deployment Project

Deployment to production time repository is performed by using deployment projects. A deployment project is a list of rule projects and specific project versions to be deployed together to production time repository. Deployment projects are useful for recording the history of project deployments.

Deployment projects are listed in the project tree, in the **Deployments** category. Just like rule projects, deployment projects are stored in design time repository and can be versioned.

To create a deployment project, proceed as follows:

1. Click **Create Deployment** in the top line menu.
2. In the **New Deployment** window, enter the deployment project name and click **Create**.
The new deployment project appears in the project tree.
3. Define deployment project descriptors as described in [Defining Deployment Project Descriptors](#).

Defining Deployment Project Descriptors

A descriptor is a reference to one specific version of a rule project to be included in the deployment. Descriptors must be added to the deployment project specifying which rule projects and project versions are deployed.

To add a new descriptor to the deployment project, proceed as follows:

1. If the deployment project is not checked out, check it out as described in [Checking Out a Project](#).
2. In the project tree, select the deployment project and, in the right pane, select the **Descriptors** tab.

Properties	Versions	Descriptors			
Selected	Name	Version	Message	Actions	
<input checked="" type="checkbox"/>	org.openl.tablets.tutorial4	0.0.1			
Add		Open			

Figure 74: Deployment descriptors

The **Descriptors** tab displays existing descriptors of the selected deployment project.

3. To add a new descriptor, click **Add** and specify the project and version to be included in the deployment.
4. Repeat this procedure to add as many descriptors as required.

Deploying a Deployment Project

To deploy a deployment project, check it in and click **Deploy**.

Specified projects are deployed to production time repository and a deployment message is displayed.

Project 'NewDepTest' successfully deployed with id: NewDepTest#0.0.1

Figure 75: Deployment message

Opening Deployed Projects

Deployment projects provide the means for tracking the deployment history of project versions. OpenL Tablets WebStudio provides functionality for quickly opening the deployed project versions. This is especially useful when some time has passed since deployment and a review of files during specific deployments is desired.

To open the specific project versions included in a deployment, proceed as follows:

1. In the project tree, select the deployment project.
2. In the right pane, select the **Descriptors** tab.
3. In the **Selected** column, select the check boxes for projects to be opened.
4. Click **Open**.

The selected project versions are opened in repository editor.

Redeploying Projects

OpenL Tablets WebStudio provides a function that allows a simple update and redeployment of many related deployment projects when a particular rule project is modified. This function takes into account the version of the opened rule project and works correctly, even with older project versions.

To update related deployment projects and redeploy a rule project, proceed as follows:

1. In the project tree, select the modified rule project.
2. In the right pane, click **Redeploy**.

Note: The **Redeploy** button is disabled if the selected project is a local project or if it is checked out.

The **Auto Redeploy** window appears listing all existing deployment projects whose latest version contains a reference to the selected rule project. Deployment projects marked for deletion are not displayed.

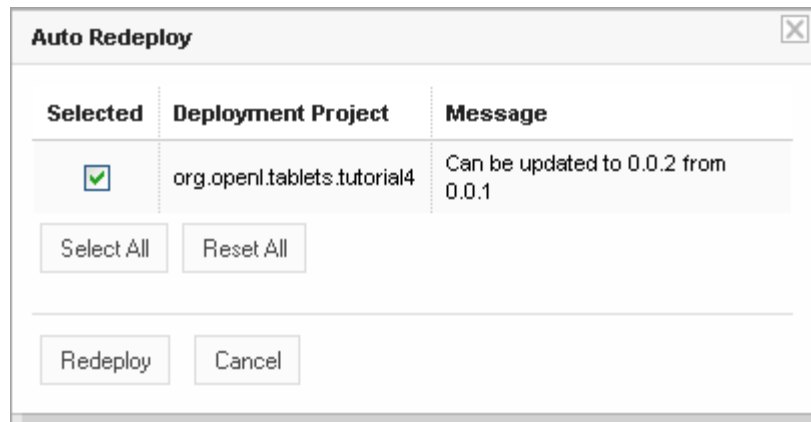


Figure 76: Redeploying a project

The **Message** column displays the current status of displayed deployment projects. If a particular deployment project cannot be redeployed, the check box in the **Selected** column is gray. Following are possible reasons for a deployment project to be disabled:

- The deployment project is checked out.
- The deployment project is locked by another user and cannot be updated.
- The deployment project is up to date and references the selected version of the rule project.
- The deployment project references a version of the rule project that is higher than the one currently opened.

If the selected rule project is not referenced by any existing deployment project, the system offers to create a new deployment project containing only the rule project with an identical name.

3. Select check boxes for the deployment projects that must be updated and redeployed.
4. Click **Redeploy**.

Update and redeployment results are displayed in the user interface.



Figure 77: Redeployment results

Comparing Project Versions

OpenL Tablets WebStudio provides a function for comparing files and sheets in Excel files between two project versions.

To compare contents of the currently opened project version with any other version, proceed as follows:

1. In the project tree, select the project.
2. In the right pane, click **Compare**.

A window appears listing contents of the currently opened project version on the left side and contents of another project version on the right side.

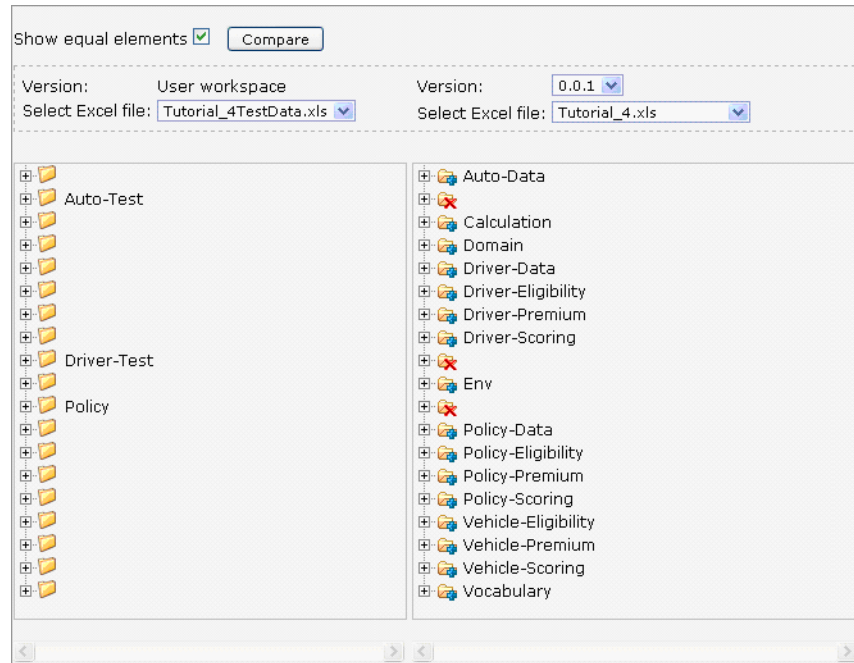


Figure 78: Comparing project versions

Green entries indicate new elements and red crosses indicate deleted or nonexistent elements.

3. To compare the current project version with a different version, in the **Version** list box, select the version number.

Using Administration Tools

This section explains how to view and control OpenL Tablets WebStudio system settings. All the settings are organized into three groups displayed in the **Administration** tab: Common, Security, and Repository. To open the desired group, you should click the corresponding icon on the left.

The screenshot shows the 'Administration' tab of the OpenL Tablets WebStudio interface. The left sidebar contains four icons: a gear for 'Common', a lock for 'Security', a cylinder for 'Repository', and a tablet for 'System'. The 'Common' icon is selected. The main content area is titled 'System Settings' and contains three sections: 'User Workspace' with 'Workspaces Directory' set to '/openl/user-workspaces' and 'Local Directory' set to 'D:\openl\local-workspace'; 'History' with 'History Directory' set to '/openl/project-history'; and 'Other' with 'Date Format' set to 'MM/dd/yyyy'. At the bottom of the main area are two buttons: 'Restore Defaults' and 'Apply'.

Figure 79: OpenL Tablets WebStudio Administration

Normally, the default settings are recommended but users with appropriate permissions can change them as needed. After making changes, you should click **Apply**. If you wish to restore the original settings, click the **Restore Default** button.

The following topics are included in this section:

- [Common settings](#)
- [Security settings](#)
- [Repository settings](#)
- [System settings](#)

Common Settings

The **Common** section defines general WebStudio settings described below.

User Workspace

Defines where the user's projects should be located:

- **Workspaces Directory** – Provides the path to the workspaces for users logged in to WebStudio by means of WebStudio login dialog. See [Logging In to OpenL Tablets WebStudio](#) for details.
- **Local Directory** – Represents the workspace for the LOCAL user.

History

History Directory – Shows where your projects history files are saved. These files are used to track, compare and revert the changes made in projects.

Other

Date Format – Enables changing the date format in the WebStudio Interface.

Security Settings

Auto Login – allows users to log in automatically with their Windows accounts. If unchecked, the system will request for credentials (Login/Password) to register to the WebStudio.

Repository Settings

Defines which type of repository should be used as a datasource:

- **Local** – The Repository is located on the local machine as a folder. This folder should be specified in the **Repository Directory** field.
- **Remote - RMI** – The Repository is located on a remote server and can be accessed by the RMI protocol. The **Repository URL** field displays URL for remote access to the repository.
- **Remote - WebDav** – The Repository is located on a remote server and can be accessed via WebDav protocol. The **Repository URL** field displays URL for remote access to the repository.

For more information on repository settings refer to the *Data Source Configuration* section in the [OpenL Tablets Web Services Usage and Customizations](#) documentation.

System Settings

Custom Spreadsheet Type – indicates whether the `Custom Spreadsheet Type` Java system property is turned on or off. By default the feature is turned off but it takes just a few simple steps to turn it on described below.

In Eclipse:

1. Open `Start WebStudio.launch` – right click **Start WebStudio.launch** and select **Open**.
2. Locate the string containing `VM_ARGUMENTS` and add `Dcustom.spreadsheet.type=true` as follows:

```
<stringAttribute key="org.eclipse.jdt.launching.VM_ARGUMENTS" value="-Xms256M -Xmx1024M -
XX:+UseParallelOldGC -XX:MaxPermSize=256M -XX:PermSize=128M -
Dworkspace.local.home=${workspace_loc} -Dcustom.spreadsheet.type=true"/>
</launchConfiguration>
```

3. Save your changes by clicking **File > Save** or the **Save** icon in the top line menu.
4. Start WebStudio.

Under Tomcat

If you work under Tomcat, edit JVM options:

1. For Windows, open TOMCAT_HOME/bin/setenv.bat file and update it as follows:

```
set JAVA_OPTS=%JAVA_OPTS% -Xms256m -Xmx1024m -XX:+UseParallelOldGC -XX:PermSize=128m -
XX:MaxPermSize=384m -Dcustom.spreadsheet.type=true
```

2. For UNIX, open TOMCAT_HOME/bin/setenv.sh file and update it similarly:

```
export JAVA_OPTS="$JAVA_OPTS -Xms256m -Xmx1024m -XX:+UseParallelOldGC -XX:PermSize=128m -
XX:MaxPermSize=384m -Dcustom.spreadsheet.type=true"
```

Save your changes and restart Tomcat.

Rules Dispatching Mode – Indicates what rule tables dispatching mode is set:

- **Java** – is set by default
- **Decision Table dispatching**

Index

A

advanced functionality, 31

B

benchmarking, 43

D

deployed project
 opening, 57
deployment project
 creating, 56
 defining descriptors, 56
 deploying, 57
design time repository
 browsing, 45

F

file
 deleting, 53
 uploading, 53
folder
 creating, 52
 deleting, 53

G

guide
 audience, 5
 related information, 5
 typographic conventions, 5

M

module
 opening, 18

O

OpenL Tablets WebStudio
 components, 8
 definition, 7
 logging in, 10
 security, 9
 user interface, 10
 working with projects, 7
OpenL Tablets WebStudio, 7

P

project
 checking in, 18, 49
 checking out, 18, 48
 closing, 48
 compare versions, 18
 copying, 55
 creating, 47
 deleting, 55
 deploying, 56
 deployment, 56
 erasing, 55
 managing, 18
 modifying, 50
 modifying contents, 52
 modifying properties, 51
 opening, 48
 redeploying, 57
 removing, 55
 revert changes, 18
 uploading, 46
project dependency
 defining, 49
project tree
 filtering, 46
project versions
 comparing, 58

R

repository editor, 16
 using, 45
rule editor, 10
 managing projects, 18
 opening a module, 18
 overview, 11
 using, 17
 view modes, 12

S

search
 advanced, 24
 modes, 23
 performing, 23
 simple, 24
security, 9

T

tables
 modifying, 22
 viewing, 21
tracing, 40

U

unit tests, 35

V

view modes, 12