



OpenL Tablets WebStudio User's Guide

OpenL Tablets 5.11

OpenL Tablets BRMS

Document number: TP_EPB_7.1.0_OpenLWS_User_2.3_LShSN
Revised: 09-10-2013



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
Signing In to OpenL Tablets WebStudio.....	10
Understanding the User Interface	11
Rules Editor	11
Repository Editor	17
Chapter 3: Using Rules Editor	19
Filtering Projects	19
Opening a Module.....	20
Managing Projects.....	20
Editing and Saving a Project.....	20
Exporting and Updating a Module	20
Reverting Module Changes	20
Viewing Tables.....	23
Modifying Tables	24
Editing Range Data Types.....	26
Copying Tables	30
Performing a Search	31
Simple Search	31
Advanced Search	32
Creating New Table.....	34
Datatype Table Wizard	34
Data Table Wizard.....	36
Simple Rules Wizard	38
Chapter 4: Editing and Testing Functionality	41
Editing Tables	41
Editing Table Properties	41
Unit Tests.....	48
Navigation.....	48
Run Tests.....	48
Creating New Test.....	54
Tracing.....	54
Benchmarking	57
Chapter 5: Using Repository Editor	59
Browsing Design Repository	59
Filtering the Project Tree	60

Creating Projects in Design Repository	61
Create a Project from Template	61
Create a Project from Excel File(s)	62
Create a Project from ZIP Archive	64
Import a Project from Workspace	65
Opening a Project	66
Closing a Project	67
Editing a Project	67
Saving a Project	67
Defining Project Dependencies	68
Modifying a Project	69
Modifying Project Properties	69
Modifying Project Contents	72
Copying a Project	74
Removing a Project	75
Deleting a Project	75
Erasing a Project	75
Deploying Projects	76
Creating a Deploy Configuration	76
Defining Projects to Deploy	76
Deploying a Deploy Configuration	77
Opening Deployed Configurations	77
Redeploying Projects	78
Comparing Project Revisions	79
Exporting a Project	80
Unlocking a Project	80
Using Administration Tools	81
Managing System Settings	81
Common Settings	82
Repository Settings	83
System Settings	84
User Management	84
Manage Groups	84
Manage Users	87
Index	90

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](#)

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 **Design repository**. Design repository is stored on the OpenL Tablets WebStudio server and can be accessed by all users. However, users cannot modify projects directly in Design repository. Instead, to make modifications to a project, users must execute the following procedure:

Procedure for modifying a project		
Step	Action	Description
1	Open a project for editing.	Opening a project for editing (the project gets the Editing status) creates its copy in user's workspace, a specific location on the OpenL Tablets WebStudio server. Working copies of projects made editable by the particular user are stored there. Users can only access their personal workspaces. A project in the Editing status is locked in Design repository to avoid loss of information. Other users cannot edit it until the project is saved. Other users can only open projects in the Editing status in read only mode.
2	Modify a project.	Modifications to a project in the Editing status 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	Save a project.	<p>Saving a project copies the modified copy of the project from user's workspace to Design repository. A new revision of the project is created in Design repository. A project can be restored to any of its previous revisions.</p> <p>From this point, changes are visible to other users and the project is available for editing.</p>

In addition to opening projects for editing (the project obtains the Editing status) and their saving, users can also open (the project obtains the Opened status) and close them. An open project is copied from Design 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 making it editable. A project in the Editing status is locked for editing by other users.

Closing a project deletes it from user's workspace without saving changes and does not affect the revision in Design repository. Closed projects can be browsed in repository editor but are not available in Rules Editor.

OpenL Tablets WebStudio Components

OpenL Tablets WebStudio consists of the following main components:

OpenL Tablets WebStudio components	
Component	Description
Rules Editor	<p>Graphic user interface running in a web browser allowing users to browse rule modules, modify table data, and run tests.</p> <p>Rules Editor is the default user interface displayed when user opens OpenL Tablets WebStudio.</p> <p>Rules 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>Rules 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 in the Editing status. For information on opening and editing projects, see Working with Projects in OpenL Tablets WebStudio.</p> <p>For detailed information on using Rules Editor, see Chapter 3: Using Rules Editor.</p>

OpenL Tablets WebStudio components	
Component	Description
Repository editor	<p>Graphic user interface running in a web browser allowing users to browse and manage projects in Design repository.</p> <p>Unlike Rules Editor, repository editor displays physical contents of rule projects.</p> <p>Users can easily switch between Rules 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 Design repository • editing, saving, opening, and closing projects • modifying project structure and properties • managing project revisions and dependencies • copying and deleting projects in Design repository • managing and tracing deploy configurations <p>For detailed information on using repository editor, see Chapter 5: Using Repository Editor.</p>
Design repository	<p>Centralized storage of rule projects accessible by all OpenL Tablets WebStudio users. Projects uploaded to Design repository are visible to other users.</p> <p>Design repository creates a separate project revision each time a project is saved. You can open any project revision if required.</p>
Deploy Configurations repository	<p>Centralized storage of final rule projects to be delivered to the production environment where solution applications use them.</p> <p>Projects can be deployed to production repository from Design repository using deploy configurations. A deploy configuration is a specific OpenL Tablets WebStudio project type. It stipulates which rule projects and project revisions must be deployed to production repository. Deploy configurations are saved and versioned so that developers can identify which specific rule project revisions are deployed.</p>
Production repositories	Production storages of deployed rule projects where solution applications use them.
User workspace	Project storage on the server containing projects edited by users. Each user has a personal workspace not accessible by others.

Security Overview

OpenL Tablets WebStudio supports the following user modes:

- **Single** – In this mode only one user who is currently logged in on the computer can work in OpenL Tablets WebStudio. You will want to select this mode if you are going to install WebStudio on your local machine. All user projects are located in the root of the `user-workspace` directory. Single user mode is set by default and does not require additional settings. Moreover, the system works faster in this mode but neither user management nor access control is provided. In the other words, you don't need to login/logout in the system.
- **Multi** - This mode enables multiple users to work in OpenL Tablets WebStudio and 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. When a user opens OpenL Tablets WebStudio in the web browser, a login screen appears and the user name and password must be specified. 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 some other users only have access rights to view or modify business rules. In this mode user's projects are located in the `user-workspace\<user name>` folder.

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:

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

Signing In to OpenL Tablets WebStudio

To sign 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`

In the Single mode (See [Security Overview](#) for details), by default the user is automatically logged in using the Windows account.

In the Multi-user mode the following form appears.

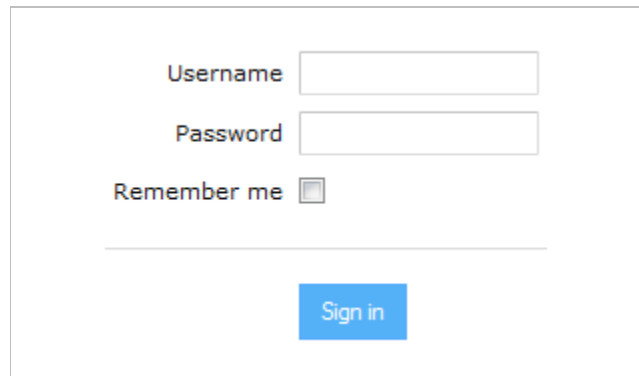
A screenshot of a web form for signing in. It contains three input fields: 'Username', 'Password', and 'Remember me' (which is a checkbox). Below these fields is a blue button labeled 'Sign in'.

Figure 1: Sign in window

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

To sign out, click an arrow next to the username on the top-right of the screen, then button **Sign out**.

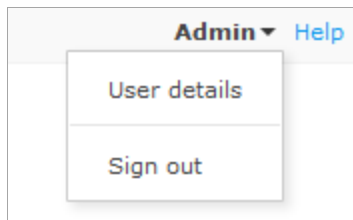


Figure 2: User's account window

To update user's profile information or change the password, click an arrow next to the username on the top-right of the screen, then on button **User details**. The system displays the dialog where user's profile information like surname, password can be specified / updated.

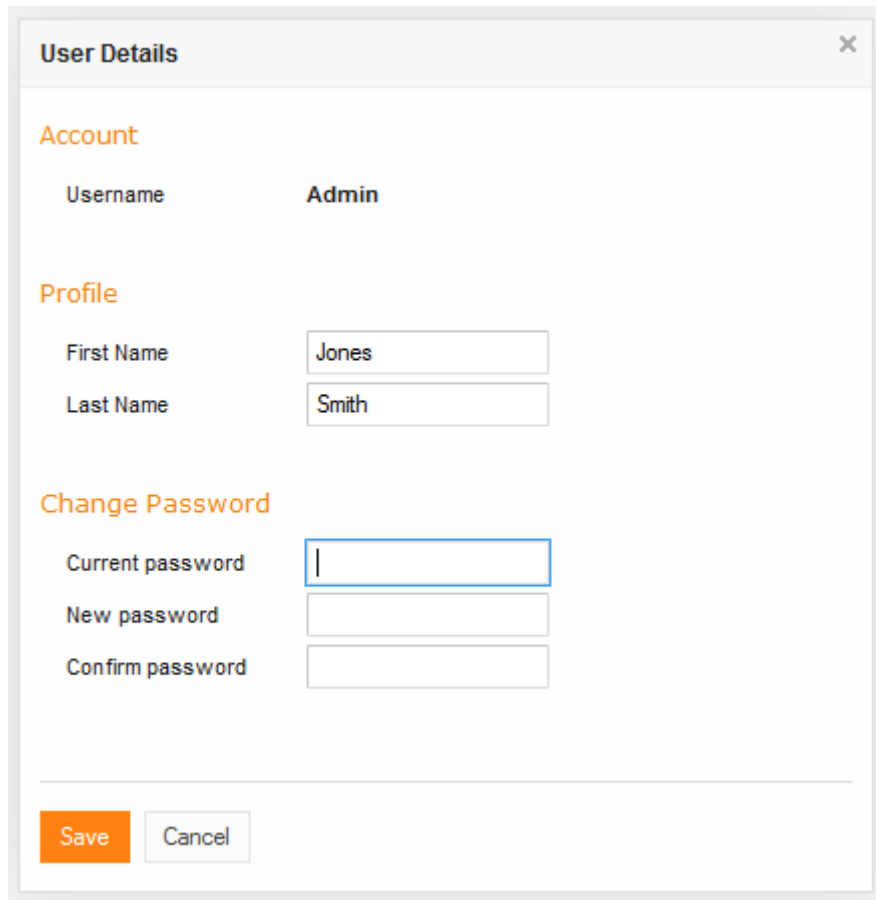
A screenshot of a 'User Details' dialog box. The dialog has a title bar with 'User Details' and a close button (X). It contains three sections: 'Account' with 'Username' set to 'Admin'; 'Profile' with 'First Name' set to 'Jones' and 'Last Name' set to 'Smith'; and 'Change Password' with three empty input fields for 'Current password', 'New password', and 'Confirm password'. At the bottom are 'Save' and 'Cancel' buttons.

Figure 3: User's details form

Understanding the User Interface

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

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

Rules Editor

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

The following topics are included in this section:

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

Rules Editor Overview

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

Rules Editor resembles the following:

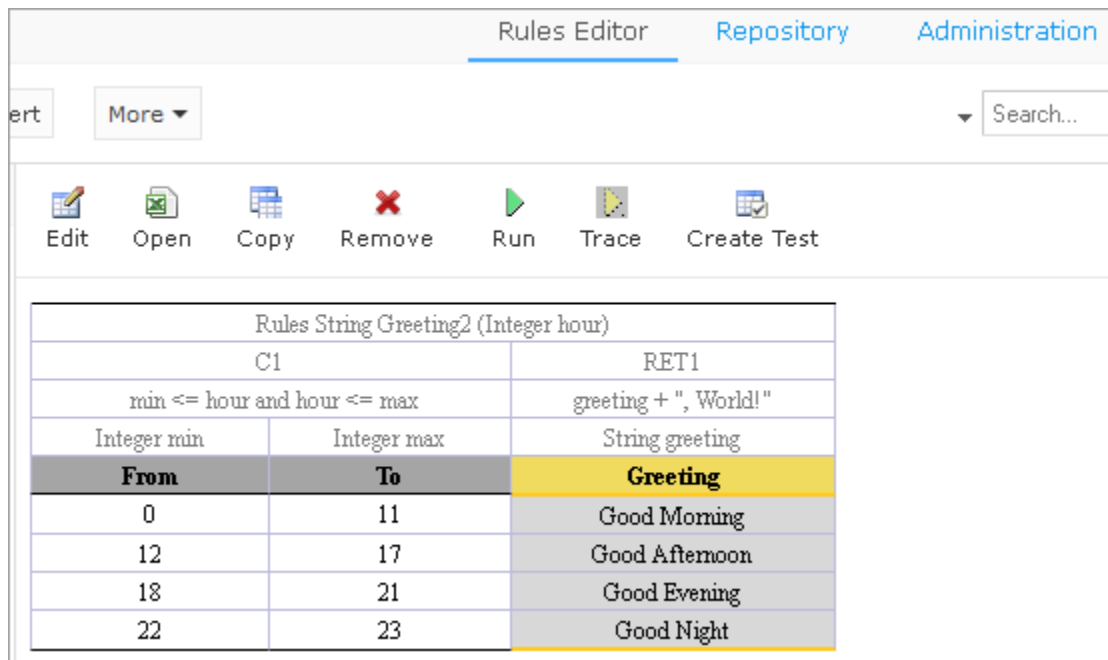


Figure 4: OpenL Tablets WebStudio Rules Editor





Rules 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:

Rules Editor toolbar buttons	
Button	Description
<div>More ▾</div> <hr/> <div>Table Dependency Graph</div> <hr/> <div>Table View Settings</div> <div><input checked="" type="checkbox"/> Show Header</div> <div><input type="checkbox"/> Show Formulas</div> <hr/> <div>Compare Excel files</div>	<p>Table Dependency Graph Opens Dependency graph.</p> <p>Table View Settings Switches to view with table headers and/or Excel formulas. For information on view modes, see View Modes.</p> <p>Compare Excel files Initiates a dialog for comparing Excel files.</p>
Revert	Opens a page for reverting module changes
	Runs a simple search. For information on performing searches, see Performing a Search .
	Refreshes OpenL Tablets WebStudio with latest changes in Excel files.
Create New	Initiates the table creation wizard.
	Displays recently visited tables instead of the module tree.
	Returns to the module tree view.
Repository	Switches user interface to repository editor. For general information on repository editor, see Repository Editor .
Help	Opens OpenL Tablets WebStudio help.
Sign out	Signs the user out of OpenL Tablets WebStudio.
User details	Opens a form with user's details info, allows changing the password or updating profile data.
Rules Editor	Switches user interface to rules editor For general information on rules editor, see Rules Editor
Administration	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 Rules 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:

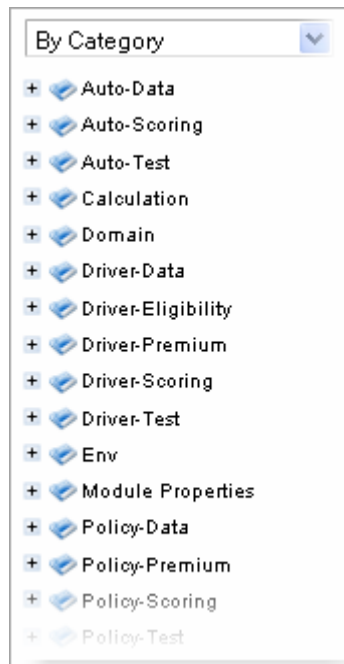


Figure 5: Module tree sorted by category

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 Rules Editor	
Mode	Description

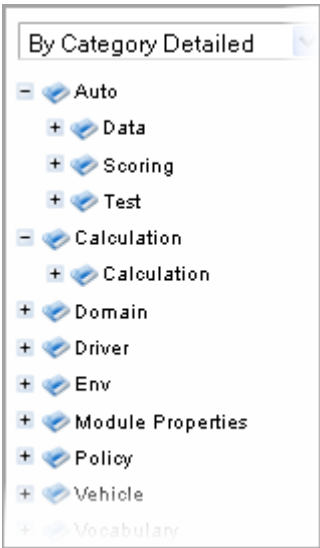


Figure 6:Module tree sorted by Category Detailed

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

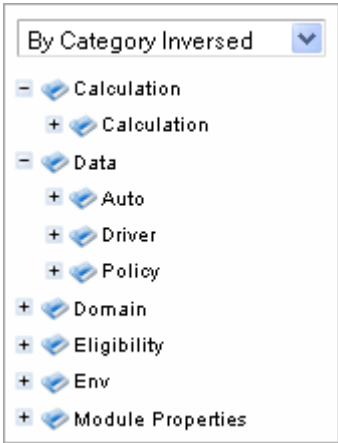


Figure 7:Module tree sorted by Category Inversed

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 Rules Editor	
Mode	Description

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

Figure 8: Rule table in simple view

The user can switch to the simple view by clicking the **More** button and deselecting the **Show 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:
----------------------	--

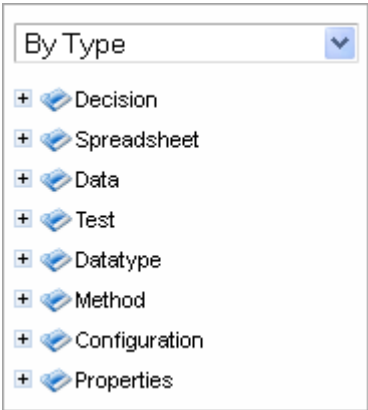


Figure 9: Module tree sorted by type

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

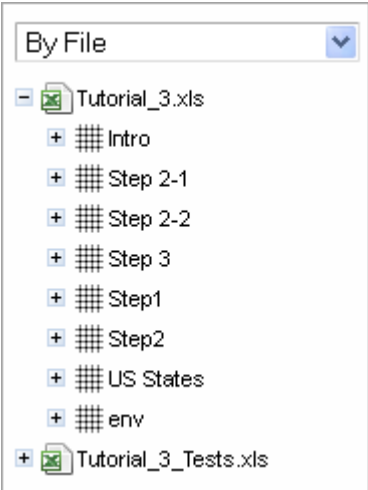


Figure 10: Module tree sorted by file

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 Rules 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 11: Rule table in extended view

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

Repository Editor

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

WebStudio Rules Editor Repository Administration

Create Project Create Deploy Configuration

Design - Production Projects

Filter by Name

- [-] Projects
 - + Empty Project
 - + Example 1 - Bank Rating
 - + Example 2 - Corporate Rating
 - + Example 3 - Auto Policy Calculat...
 - + Module1 [Local]
 - + MyTestProject_from templ [Local]
 - + New Test snm [Local]
 - + Tutorial 1 - Introduction to Decisi...
 - + Tutorial 2 - Introduction to Data ...
 - + Tutorial 3 - More Advanced Dec...
 - + Tutorial 4 - Introduction to Colum...
 - + Tutorial 7 - Introduction to Table ...
 - + org.openl.tablets.tutorial1 [Local]
 - + org.openl.tablets.tutorial10 [Local]
 - + org.openl.tablets.tutorial2 [Local]
 - + org.openl.tablets.tutorial3 [Local]
 - + org.openl.tablets.tutorial4 [Local]
 - + org.openl.tablets.tutorial5 [Local]
 - + org.openl.tablets.tutorial7 [Local]
 - + test revision number [Local]
 - + test snm [Local]
- Deploy Configurations

Name	Revision	Status	Modified By	Modified At	Actions
Empty Project	1	Editing	snm	04/18/2013	
Example 1 - Bank Rating	1	Editing	snm	04/18/2013	
Example 2 - Corporate Rating	1	Editing	snm	04/16/2013	
Example 3 - Auto Policy Calculation	1	Editing	snm	04/16/2013	
Module1	6	Local			
MyTestProject_from templ	0	Local			
New Test snm	1	Local			
Tutorial 1 - Introduction to Decision Tables	1	Editing	snm	04/16/2013	
Tutorial 2 - Introduction to Data Tables	1	Editing	snm	04/16/2013	
Tutorial 3 - More Advanced Decision and Data Tables	1	Editing	snm	04/16/2013	
Tutorial 4 - Introduction to Column Match Tables	1	Editing	snm	04/16/2013	
Tutorial 7 - Introduction to Table Properties	1	Editing	snm	04/16/2013	
org.openl.tablets.tutorial1	1	Local			
org.openl.tablets.tutorial10	0.0.1	Local			
org.openl.tablets.tutorial2	1	Local			
org.openl.tablets.tutorial3	1	Local			
org.openl.tablets.tutorial4	3	Local			

Figure 12: OpenL Tablets WebStudio repository editor

The left pane contains a tree of projects stored in Design Repository and user's workspace. Unlike Rules 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 Rules 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 Rules Editor

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

The following topics are included in this section:

- [Filtering Projects](#)
- [Opening a Module](#)
- [Managing Projects](#)
- [Viewing Tables](#)
- [Modifying Tables](#)
- [Copying 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.

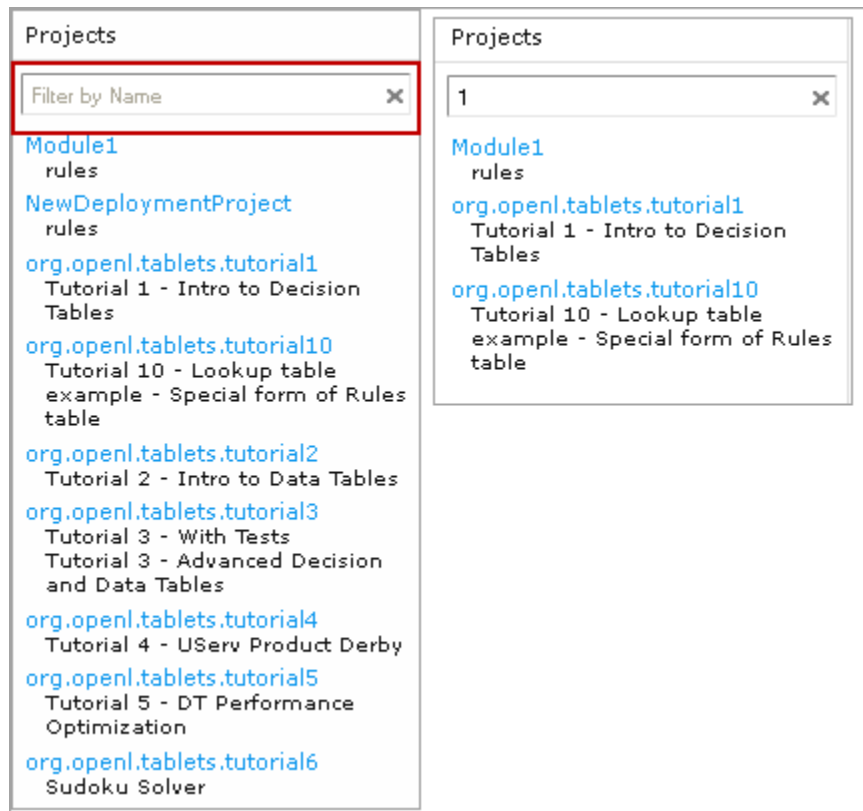


Figure 13: Filtering projects by Name

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

Opening a Module

Rules 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 Rules Editor:

- [Editing and Saving a Project](#)
- [Exporting and Updating a Module](#)
- [Reverting Module Changes](#)

Editing and Saving a Project

A project can be opened for editing and saved directly from the Rules Editor.

If you want to have an open project in the Editing status, above the module tree, clicks **Edit Project** .

If the project is edited, to save it, click **Save Project** .

NOTE: If the project is in the Local status these options are not available from the Rules Editor.

Exporting and Updating a Module

A user can export or update a module directly from Rules Editor. The following buttons in the top line menu



- import and update the module into the project



- export the module on user's local machine

perform this functionality for the project in Editing status.

Reverting Module Changes

OpenL Tablets WebStudio provides functionality allowing users to compare and/or roll back module changes against a specific date.

To compare module versions, select the desired module 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 module 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"/>
<div>Revert Compare</div>			

Figure 14: 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 15: Compare / Revert project changes – Step 2

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

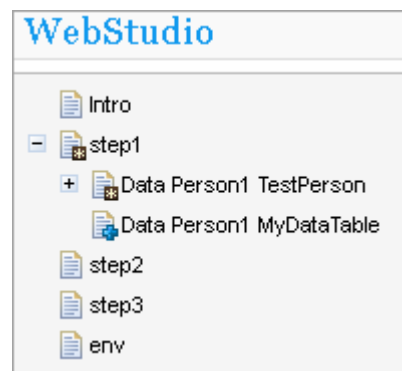


Figure 16: Table(s) with changes

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

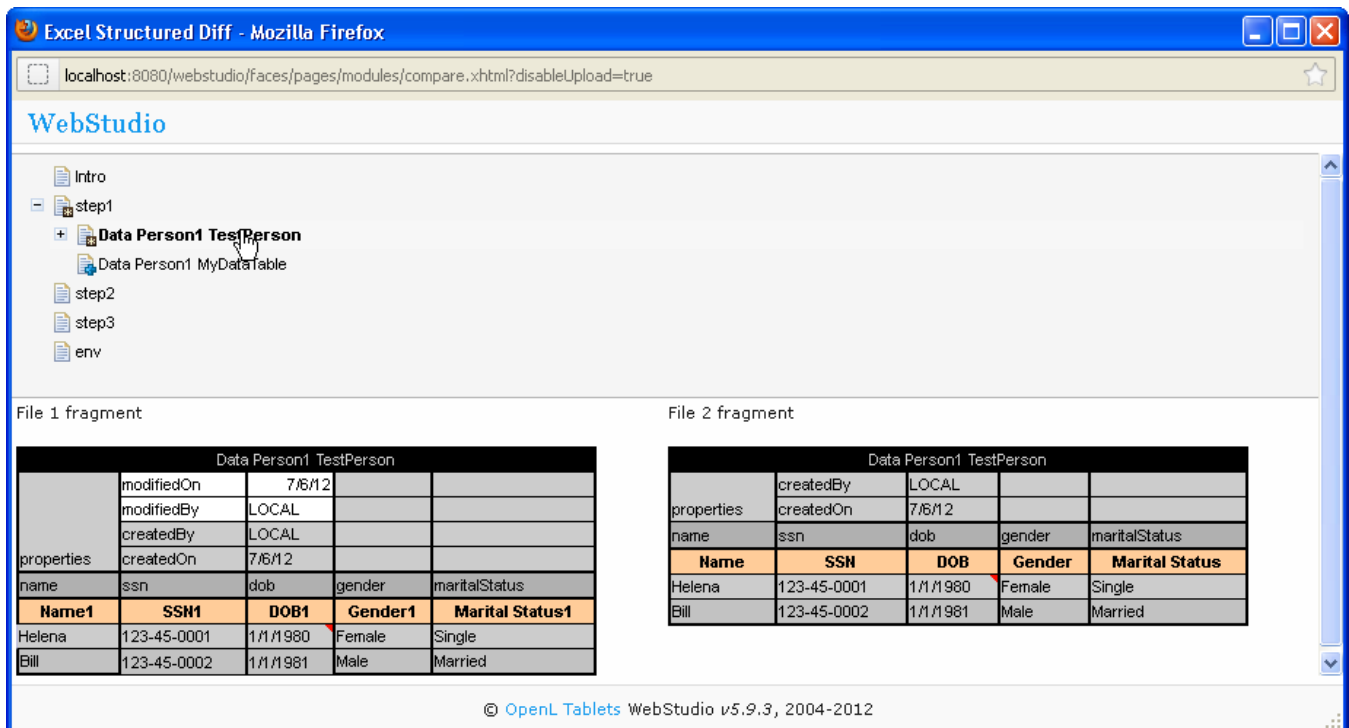


Figure 17: The result of the comparison

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"/>
<div> Revert Compare </div>				

Figure 18: Revert project changes

3. 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 Rules 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 in the Editing status, 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 in the Editing status, open it for editing as described in [Editing and Saving a Project](#)
2. In the module tree, select the required table.


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




Edit




Open




Copy




Remove




Run



Trace



Test



Create Test

Available Tests/Runs

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

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

Figure 19: Table opened in OpenL Tablets WebStudio

- If required to switch between simple / extended view, click the **More** button at the top right of the screen and select / deselect the **Show Header** and/or **Show Formula** options.
- 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 in the Editing status.

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 directly in Rules Editor (as described in [Exporting and Updating a Module](#)) or via the repository.

The following table is switched to edit mode.



















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

Figure 20: 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.

Table editing buttons	
Button	Description
	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** .

Editing Range Data Types

OpenL Tablets WebStudio provides a special tool — *range editor*, — which helps you to add and/or edit range data types: *IntRange* and *DoubleRange*, — in rule tables and Test tables.

We start from brief description of Range Editor concepts and then consider a very simple example of how the tool works.

The main Range Editor goal is to move to a single range format in OpenL rules, namely, the ‘..’ format. You can find detailed information about ranges on OpenL Tablets in the *Range types in OpenL* section of the [OpenL Tablets Reference Guide](#) document. While working with Range Editor, take into account the following.

- The default range format is set to ‘..’ in OpenL Tablets WebStudio.
- If you create a new range, the ‘..’ format will be used.
- If you edit a range format other than ‘..’, your format will remain the same in case you only edited values of the range. If you use any control in the editor, i.e. a checkbox, or the **Done** button and so on, the range format will be changed to ‘..’.

As an example, assume we have a Decision Table with some data represented as range like the following one.

Rules String Greeting3 (Integer hour)	
C1	RET1
hour	greeting + ", World!"
IntRange	String greeting
Hour	Greeting
0 - 11	Good Morning
12 - 17	Good Afternoon
18 - 21	Good Evening
	Good Night

Figure 21: Decision Table with a Range Data type

In this table, the **Hour** column contains hours with the IntRange Data type. All range sells are filled except for the last one. We will use this example to demonstrate how the Range Editor works.

Here is a list of controls available in the Range Editor:

- **From** — indicates the left border of the range.
- **To** — indicates the right border of the range.
- **Include** — indicates whether or not the border will be included to range.
- **'>'** — indicates values more than the specified border
- **'<'** — indicates values less than the specified border
- **'='** — indicates a constant
- **'-'** — indicates a range

Follow the steps below to enter new range.

1. Double click a cell you want to edit. For example, the cell containing 18-21. Your table extends by the following popup with a set of controls that allow editing the range.

Rules String Greeting3 (Integer hour)	
C1	RET1
hour	greeting + ", World!"
IntRange	String greeting
Hour	Greeting
0 - 11	Good Morning
12 - 17	Good Afternoon
19 - 21	Good Evening
	Good Night

Include

☒

From

>

<

.

=

To

Include

☒

Done

Figure 22: Create a range in Range Editor – step 1

- In the **From** field, enter the left border of the range, it is 22 in our example.
- In the **To** field, enter the right border of the range, it should be 24 in our example, but we will enter 23 to have a chance to edit this border later. And also let's uncheck right **Include**. Note how your range looks in the form.

Rules String Greeting3 (Integer hour)	
C1	RET1
hour	greeting + ", World!"
IntRange	String greeting
Hour	Greeting
0 - 11	Good Morning
12 - 17	Good Afternoon
19 - 21	Good Evening
	Good Night

Include

☒

From

22

>

<

.

=

To

23

Include

☐

[22 .. 23)

Done

Figure 23: Create a range in Range Editor – step 2

- Click **Done** to complete. The last cell in the **Hour** column is filled as follows.

Hour	Greeting
0 - 11	Good Morning
12 - 17	Good Afternoon
18 - 21	Good Evening
[22 .. 23)	Good Night

Figure 24: New range created in Range Editor

Now we will consider how to change a range with the Range Editor. We will start with the range created in the previous example and will change it to demonstrate how it works.

1. Double click the cell with the [22-23) range. Your table looks like this one:

Rules String Greeting3 (Integer hour)	
C1	RET1
hour	greeting + ", World!"
IntRange	String greeting
Hour	Greeting
0 - 11	Good Morning
12 - 17	Good Afternoon
18 - 21	Good Evening
[22 .. 23)	Good Night

Include

☒

From

22

>

<

=

To

23

Include

☐

[22 .. 23)

Done

Figure 25: Edit a range in Range Editor

2. Select the **To** field, set the right border to 24 and select **Include**. Click **Done** to save your work. Your range looks as follows.

Hour	Greeting
0 - 11	Good Morning
12 - 17	Good Afternoon
18 - 21	Good Evening
22 .. 24	Good Night

Figure 26: The range edited in Range Editor


You can also change your range by using '>', '<' and '=' controls described above in this topic.

Copying Tables

If you need to create a table based on an existing one you can copy that table in one of the following ways.

- As a New Table
- As a New Version
- As a New Business Dimension Version

To create a copy as a new table

1. Select a table you wish to copy in the module list.
2. Click the **Copy Table** icon: . The system displays the **Copy Table** form with **New Table** selected by default:

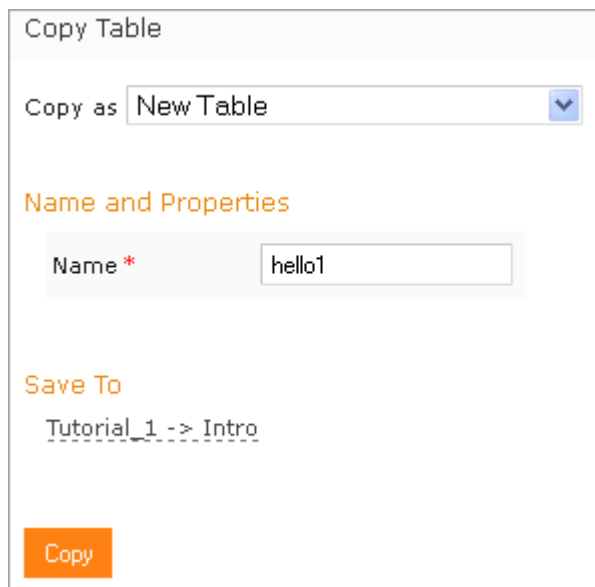


Figure 27: Creating a Copy as a New Table

3. Update the **Name** field (optional).
4. If you want to change the workbook and worksheet where the copy is to be saved click the link in the **Save To** area.
5. Select the desired module and category from the corresponding drop down lists. You can also choose to save your copy in a new category using the **New** option.
6. Click **Copy** to save your changes. The table appears in the module list.

Copying a table as a New Version is described in details in the [Copy as New Version](#) section. In that case dimensional properties of a new version are exactly the same as for the original one. OpenL Tablets allows you to create an overloaded table from an existing one.

To a copy as a New Business Dimension version:

1. Click the **Copy Table** icon for the selected table in the module list.
2. Select **New Business Dimension Version** in the **Copy as** list.
3. Enter business dimension properties as you need.

4. Change or leave the same workbook and worksheet in the **Save as** area.
5. Click **Copy** to save your work.

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 module.

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



Figure 28: 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.

11 tables found

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

Time for executing OpenL Rules heavily depends on the complexity of con
 Let's look at "Greeting" rules from Tutorial1 which only difference are cond

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

Rules String Greeting1 (Integer hour)		
C1		RET1
min <= hour and hour < max		greeting + ", World!"
Integer min	Integer max	String greeting
From	To	Greeting
0	12	Good Morning
12	18	Good Afternoon
18	22	Good Evening
22	24	Good Night

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

Rules String Greeting2 (Integer hour)		
C1		RET1
hour		greeting + ", World!"
Integer min	Integer max	String greeting
From	To	Greeting
0	12	Good Morning

Figure 29: 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 30: Advanced Search – Step 1

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.
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.
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.

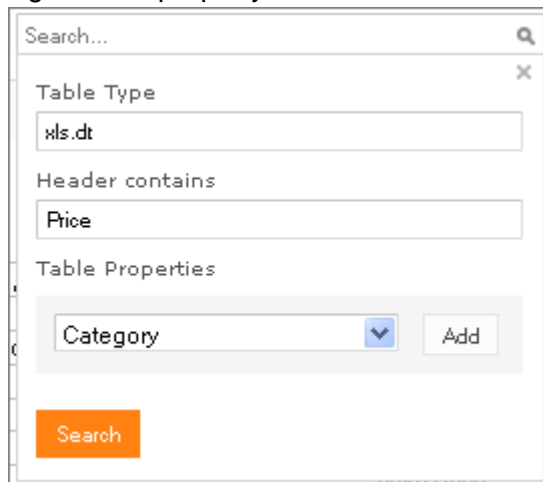
The image shows a web-based search form titled "Search...". It contains three main sections: "Table Type" with a text input field containing "xls.dt"; "Header contains" with a text input field containing "Price"; and "Table Properties" which is a list of properties. One property, "Category", is selected and shown in a text input field next to a blue downward arrow and an "Add" button. At the bottom of the form is a large orange "Search" button.

Figure 31: Advanced Search – the filled form

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.

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


Rules DoubleValue CarPrice (Car car, Address billingAddress)								
properties	effectiveDate	1/1/09						
	expirationDate	1/1/10						
Rule	C1	C2	HC1	HC2	RET1			
	country	region	brand	model				
	Country	String	CarBrand	String				
# Rule	Country	Region	BMW		Porsche			
			Z4 sDrive35i	Z4 sDrive30i	911 Carrera 4S	911 Targa 4	911 Carrera Cabriolet	2009 Audi R8 4.2 quattro Auto
R1	USA	Pacific West	\$51,650	\$45,750	\$93,200	\$90,400	\$87,000	\$121,500
R2		West	\$52,000	\$44,050	\$93,200	\$90,400	\$87,000	\$121,500
R3		Mid Atlantic	\$52,450	\$46,550	\$93,200	\$90,400	\$87,000	\$121,500
R4	Great Britain	England	\$53,650	\$47,750	\$94,200	\$91,400	\$88,000	\$121,500
R5		Wales	\$53,650	\$47,750	\$95,200	\$92,400	\$89,000	\$121,500
R6		Scotland	\$53,650	\$47,750	\$96,200	\$93,400	\$90,000	\$121,500
R7	Belarus	Minsk	\$56,650	\$49,750	\$93,200	\$90,400	\$87,000	\$121,500
R8		Vitebsk	\$56,650	\$49,750	\$93,200	\$90,400	\$87,000	\$121,500
R9		Grodna	\$56,650	\$49,750	\$93,200	\$90,400	\$87,000	\$121,500

Figure 32: Advanced Search result

Creating New Table

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

- Datatype table
- Datatype Alias table
- Data table
- Test table
- Properties table
- Simple Rules table

To open the table wizard page, click the **Create Table** button: . By using the wizards, you will normally create a table for a current module. If your project contains included modules, the table is available from all that modules, as well as for all the modules linked by dependencies. (See the *Module dependencies* section in [OpenL Tablets Reference Guide](#) for more information on dependencies.)

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.

Figure 33: Table wizards

- On the next screen, enter the name for your new data type in the **Name** text box. You can also select an existing data type in the **Parent type** drop down list if needed. In this case, your data type will have access to all fields defined in the parent data type. (See the *Inheritance in Data types* section in [OpenL Tablets Reference Guide](#) for details.) This option is unavailable if no custom data types are available in the module.

Figure 34: Datatype table wizard – Step 1

- Define the data type fields. To add more types, click the **Add parameter** link. Click **Next** to proceed.

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

Figure 35: Datatype table wizard – Step 2

- On the next page, indicate where your new Datatype table should be located. For that, define the module to which the new table belongs. If there are included modules in the project you will see all of them in the **Module** drop down list. In the **Category** area, select an existing sheet or specify a new one in the **New** text box.

Figure 36: Datatype table wizard – Step 3

- Click **Save** to complete. Your new Datatype table will be created and available in the OpenL Tablets WebStudio.

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.

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

Figure 37: Data Table wizard - Step 1

- From the **Type** list, select the table Type; enter the table name in the **Name** text box.

Figure 38: Data Table wizard – Step 2

3. Define the table columns configuration. In our example we create Data table for the custom Data Type "Loss1" so the screen will look similar to the next one.

Figure 39: Data Table wizard – Step 3

4. Indicate where your new Data table will be located. For that, define the module to which the new table belongs. If there are included modules in the project you will see all of them in the **Module** drop down list. In the **Category** area, select an existing sheet or specify a new one in the **New** text box.

Figure 40: Data Table wizard – Step 4

5. 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.

Simple Rules Wizard

This section describes how to create a new Simple Rules table in OpenL Tablets WebStudio.

1. Click the **Create Table** button and, select **Simple Rules Table** in the **Select table type** list and click **Next**:

Select table type

- ☐ Datatype Table
- ☐ Datatype Alias Table
- ☐ Data Table
- ☐ Test Table
- ☐ Properties Table
- ☒ Simple Rules Table

Next

Figure 41: Simple Rule Table wizard – Step 1

2. In the next screen, enter your table name in the corresponding field, select the desired Data type in the **Return Value Type** drop down list and click **Add Input Parameters**. The following form appears.

Enter the initial parameters

Table Name* MySimpleRule

Return Value Type* Integer

Type	Is Array	Name
BigDecimal	<input type="checkbox"/>	

+ Add Input Parameters

Prev Next Cancel

Figure 42: Simple Rule Table wizard – Step 2

3. Specify input parameters for your table and click next to proceed:

Enter the initial parameters

Table Name*

Return Value Type*

Type	Is Array	Name
<input type="text" value="String"/>	<input type="checkbox"/>	<input type="text" value="DriverAge"/> ✗
<input type="text" value="String"/>	<input type="checkbox"/>	<input type="text" value="MaritalStatus"/> ✗

[+ Add Input Parameters](#)

Figure 43: Simple Rule Table wizard – Step 3

4. In the next screen you get a blank Simple rule table with the header constructed based on your previous steps. Now you can fill in the table with your data as explained in the **Tips** box.

Construct a table

SimpleRules DoubleValue MySimpleRule (String DriverAge, String MaritalStatus)

DriverAge	MaritalStatus	RETURN

Tips

- For a cell value editing click left mouse button
- For an action with table click right mouse button on a cell

Figure 44: Simple Rule Table wizard – Step 4

5. In the next step you can do one of the followings by selecting appropriate option from the list:

Construct a table

SimpleRules DoubleValue MySimpleRule (String DriverAge, String MaritalStatus)

DriverAge	MaritalStatus	RETURN

Tips

- For a cell value editing click left
- For an action with table click right

[Add Rule](#)
[Insert Condition Before](#)
[Insert Condition After](#)
[Delete Condition](#)
[Add Property](#)

Figure 45: Simple Rule Table wizard – Step 6

Add Property — The properties row appears where you can select property from drop down list and indicate their values.

Add Rule — A new empty row appears allowing you to enter the data as you want, like in the example below:

SimpleRules DoubleValue MySimpleRule (String DriverAge, String MaritalStatus)		
DriverAge	MaritalStatus	RETURN
Young Driver	Married	200

Figure 46: Simple Rule Table wizard – Step 7

You should repeat until add as many rows as required.

Insert Condition Before / Insert Condition After — Condition column will be added in the specified position. In the following example the **DriverOccupation** condition column was added.

SimpleRules DoubleValue MySimpleRule (String occupation, String DriverAge, String MaritalStatus)			
DriverOccupation	DriverAge	MaritalStatus	RETURN
Teacher	YoungDriver	Married	200
IT	SeniorDriver	Single	220

Figure 47: Simple Rule Table wizard – Step 8

Delete Condition / Delete Rule — You can remove your condition /rule if required.

- Click **Next** after all information is added. The **Select destination** screen appears where you should indicate where your new Simple Rule table will be located. For that, define the module to which the new table belongs. If there are included modules in the project you will see all of them in the **Module** drop down list. In the **Category** area, select an existing sheet or specify a new one in the **New** text box.
- Click **Save** to save your work. Your new Simple Rules table is created and available in your project.

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.

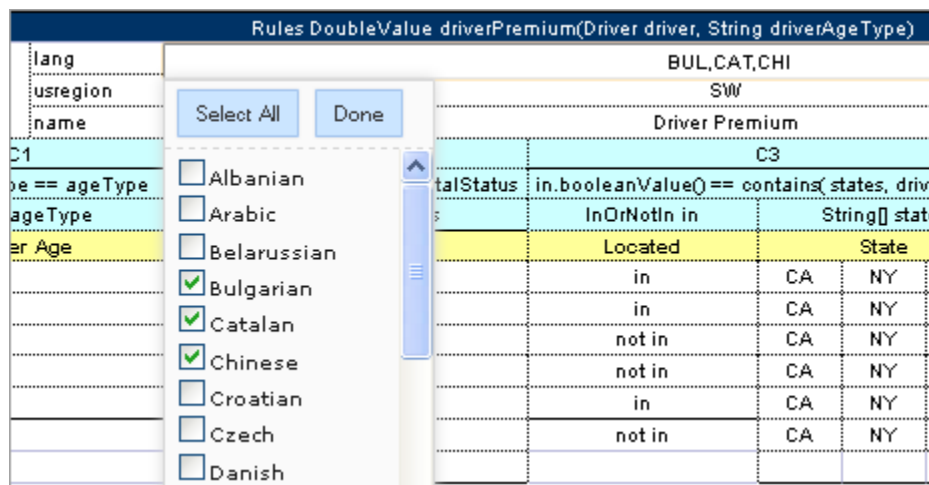


Figure 48: Multi editor in WebStudio for editing comma separated arrays

Editing Table Properties

This section describes table properties available in OpenL Tablets WebStudio. For detailed information on table properties refer to [OpenL Tablets Reference Guide](#), section *Table Properties*.

Default Properties

If default property values are defined for a table they appear only in the right hand **Properties** section, but not in the table. In the example below, there are **Active = true** and **Fail On Miss = false** default properties.

The screenshot shows the OpenL Tablets WebStudio interface. At the top is a toolbar with icons for Edit, Open, Copy, Remove, Run, Trace, and Create Test. Below the toolbar is a table titled "Rules String Greeting (Integer hour)". The table has three columns: "properties", "description", and "RET1". The "properties" column contains "C1", "min <= hour", "Integer min", "From", "0", "12", "18", and "22". The "description" column contains "C2", "hour <= max", "Integer max", "String greeting", "Good Morning", "Good Afternoon", "Good Evening", and "Good Night". The "RET1" column contains "greeting + ', World!'", "String greeting", and "Good Morning". The "From" and "To" columns are highlighted in yellow. To the right of the table is a sidebar titled "Table Details" with sections for Name, Info, Description, Version, and Dev. The Name is "Greeting". The Description is "The rule table determines appropriate greeting according to input hours.". The Version is "true". The Dev section has a "Fail On Miss" checkbox which is checked.

properties	description	RET1
C1	C2	greeting + ', World!'
min <= hour	hour <= max	String greeting
Integer min	Integer max	String greeting
From	To	Greeting
0	11	Good Morning
12	17	Good Afternoon
18	21	Good Evening
22	23	Good Night

Table Details

Name: **Greeting**

Info

Description: The rule table determines appropriate greeting according to input hours.

Version

Active: true

Dev

Fail On Miss: false

[Add Property](#)

Figure 49: Default table properties example

Default properties can be overridden at the table level; in the other words you can change them as follows.

1. In the **Properties** section, click the default property to be changed. Instead of the property value you will see a checkbox:

The screenshot shows a dialog box titled "Dev". It contains a checkbox labeled "Fail On Miss" which is currently unchecked. Below the checkbox are two buttons: "Save" and "Add".

Figure 50: Updating a default property

2. Select or deselect the checkbox as needed and click the **Save** button. The property appears in the table with its new value:

The screenshot shows the updated table titled "Rules String Greeting2 (Integer hour)". The table has three columns: "properties", "description", and "RET1". The "properties" column contains "C1", "min <= hour and hour <= max", "Integer min", "From", "0", "12", "18", and "22". The "description" column contains "C2", "hour <= max", "Integer max", "String greeting", "Good Morning", "Good Afternoon", "Good Evening", and "Good Night". The "RET1" column contains "greeting + ', World!'", "String greeting", and "Good Morning". The "From" and "To" columns are highlighted in yellow. The "failOnMiss" property is now visible in the "properties" column, and its value is "true".

properties	description	RET1
C1	C2	greeting + ', World!'
min <= hour and hour <= max	hour <= max	String greeting
Integer min	Integer max	String greeting
From	To	Greeting
0	11	Good Morning
12	17	Good Afternoon
18	21	Good Evening
22	23	Good Night

Figure 51: Default property was updated by a user

Inherited Properties

Module or category level properties are those inherited from a **Properties** table (See [OpenL Tablets Reference Guide](#), section *Properties Table* for details). In the **Properties** section of a given table, inherited properties appear in a different color and are accompanied with a link to the **Properties** table where they are defined. The values of inherited properties are not stored in the table, they are displayed in the **Properties** section, since they are inherited and applied to this table. Inherited properties can be overridden at a Table level, i.e. you can change them.

The screenshot displays the OpenL Tablets WebStudio interface. At the top is a toolbar with icons for Edit, Open, Copy, Remove, Run, Trace, and Create Test. Below the toolbar is a table titled "Rules DoubleValue clientTierScore(Policy policy)". The table has two columns: "properties" and "RET1". The "properties" column contains "usregion" with value "MW,NE,SE" and "category" with value "Policy-Scoring". The "RET1" column contains "score" with value "DoubleValue score". Below the table is a section titled "Client" with a "Score" column. The "Client" section has two rows: "Elite" with a score of -120 and "Preferred" with a score of -50. To the right of the table is a "Table Details" panel for the "clientTierScore" table. It shows the Name, Category (Policy-Scoring), Business Dimension (LOB: category_Policy-Scoring_Lob, US Region: MW,NE,SE, Region: NCSA), Version (Active: true, Dev: false), and a link to Add Property.

Rules DoubleValue clientTierScore(Policy policy)	
properties	usregion MW,NE,SE
	category Policy-Scoring
C1	RET1
clientTier == policy.clientTier	score
String clientTier	DoubleValue score
Client	Score
Elite	-120
Preferred	-50
	0

Table Details

Name: **clientTierScore**

Info

Category: Policy-Scoring

Business Dimension

LOB: category_Policy-Scoring_Lob

US Region: MW,NE,SE

Region: NCSA

Version

Active: true

Dev: false

Fail On Miss: false

[Add Property](#)

Figure 52: An example of inherited category-level properties

To change an inherited property, follow these steps:

1. In the **Properties** section, click the inherited property to be changed
2. Enter or select from drop down list the desired value(s) and click the **Save** button:

The screenshot shows a dialog box for updating an inherited property. The "Category" is "Policy-Scoring". The "Business Dimension" section has a dropdown menu for "US Region" with the value "MW, NE, SE" selected. A red arrow points to the "US Region" dropdown. The "Version" section has a "Midwest" checkbox checked. The "Dev" section has a "Southwest" checkbox checked. The "Fail On Miss" section has a "West" checkbox checked. There are "Save" and "Add" buttons at the bottom.

Category: Policy-Scoring

Business Dimension

LOB: category_Policy-Scoring_Lob

US Region: MW, NE, SE

Region: ☐ Select All

Version

Active: ☒ Midwest

Dev: ☒ Northeast

Fail On Miss: ☒ Southeast

☐ Southwest

☐ West

Save Add

Figure 53: Updating an inherited property

The system displays the property in the table:

The screenshot shows the OpenL Tablets WebStudio interface with the rule table updated. The "usregion" property is now highlighted with a red box, and its value is "MW,NE,SE". The "category" property is also highlighted with a red box, and its value is "Policy-Scoring".

Rules DoubleValue clientTierScore(Policy policy)	
properties	usregion MW,NE,SE
	category Policy-Scoring
C1	RET1
clientTier == policy.clientTier	score

Figure 54: Inherited category-level property was updated by a user

System Properties

By default, OpenL Tablets WebStudio applies system properties to each created or edited table. If you want to switch off this option, please refer to the [Common Settings](#) section for instructions. The values of System properties are provided in the table and in the **Properties** section.

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

The “createdBy” property value is set using the name of the currently logged in user. The “createdOn” property is set according to the current date. These properties are only applied in the Multi mode (See [Security Overview](#) for details) on the first save only while creating or copying a table in OpenL Tablets WebStudio. The WebStudio users can delete those properties if required.

Rules DoubleValue driverAccidentPremium(Driver driver, String driverRisk)		
properties	modifiedOn	10/26/12
	modifiedBy	snm
C1		RET1
driverRisk == risk		accidentPremium * driver.numAccidents
String risk		DoubleValue accidentPremium
Driver Risk		Per Accident Premium
		\$160

Figure 55: An example of system properties

Properties for a 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 the table.

For example, such properties as **Validate DT** or **Fail On Miss** are available for Decision Tables. That means you can select them from the drop down list after clicking the **Add** link in the bottom of the **Properties** section. The Figure below shows that properties applied to a Decision Table.

Edit Open Copy Remove Run Trace Test Create Test

Available Tests/Runs
[Driver Eligibility Test \(3 test cases\)](#)

Rules String driverEligibility(Driver driver, String ageType)		
properties	validatedDT	off
	name	Driver Eligibility Table
C1		C2
ageType == d_ageType		hadTraining == driver.hadTraining
String d_ageType		boolean hadTraining
Driver		Training
Young Driver		No
Senior Driver		No
		Not Eligible
		Not Eligible
		Eligible

Properties
Info
Name Driver Eligibility Table
Dev
Validate DT off
Fail On Miss false
Business Dimension
LOB moduleLob
US Region SW
Region NCSA
Version
Active true
Add

Figure 56: Properties for the Decision table type

If you open a Data Table in the same project, these properties will not be available for selecting from the drop down list in the **Properties** section.

The screenshot displays the OpenL Tablets WebStudio interface. On the left, a toolbar contains icons for Edit, Open, Copy, and Remove. Below the toolbar is a table titled "Data Policy policyProfile1". The table has columns for properties, category, and Policy-Data. The data is as follows:

properties	category	Policy-Data
name		Policy1
drivers	>driverProfiles1	Sara Spencer, Sara's Son
vehicles	>autoProfiles1	2005 Honda Odyssey 2002 Toyota Camry
clientTier		Preferred
clientTerm		

On the right, the "Table Details" pane shows the table name "policyProfile 1" and category "Policy-Data". Below this, the "Business Dimension" section lists properties: LOB (moduleLob), US Region (SW), and Region (NCSA). At the bottom, a "Select property to add" dropdown menu is open, showing a list of properties including Description, Tags, Effective Date, Expiration Date, Start Request Date, End Request Date, Build Phase, Countries, Currency, Language, US States, Transaction Type, Custom1, Custom2, Cacheable, and Recalculate.

Figure 57: The Decision table properties that are not available for a Data table

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, follow the steps below.

1. Click the **Add** link in the **Properties** pane:

Table Details	
Name	Driver Eligibility Table
Business Dimension	
Effective Date	10/23/2012
LOB	moduleLob
US Region	SW
Region	NCSA
Dev	
Validate DT	off
Fail On Miss	false
Version	
Active	true
<input type="button" value="Save"/> Add Property	

Figure 58: Add new property for the current table

- Enter the desired property or select it from drop down list and click the **Add** button.

Select property to add

Effective Date

[Cancel](#)

Figure 59: Selected table property to be added

- Specify the property value, and then click the **Save** button to complete. All the steps are collected in the Figure below.

Effective Date

02/02/2012

[Add](#)

Figure 60: Saving 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 since only one

table version can be active at a time. By default, all tables are active. Below is an example of an inactive table version.

Rules DoubleValue RiskFactor2 (Date MyDate, Integer hour)		
properties	version	0.0.1
	active	false
C1	C2	RET1
dayOfWeek(MyDate)	hourOfDay(setAllHours (MyDate, hour))	
IntRange	IntRange	
Day of Week	Hour of Day	Risk Factor (%)
[2 .. 6]	[10 .. 16]	75%
	[10 .. 16]	85%
		105%

Figure 61: 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.

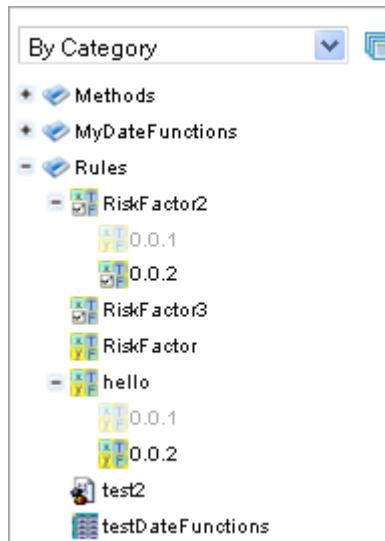


Figure 62: Displaying table versions in the module tree

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

0.0.3		Current: 0.0.2
Major	0	
Minor	0	
Variant	3	

Figure 63: 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 64: 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


 Open

 Copy

 Remove

 Run

 Trace

 Benchmark

Testmethod driverAgeType driverAgeTypeTest

driver	_res_
>testDrivers1	
Driver	Expected Age Type
Sara	Standard Driver
Spencer, Sara's Son	Young Driver

Target Table

[Driver Age Type Table \[lob = moduleLob, usregion = SW\]](#)

Figure 65: Navigation link to target table

Run Tests

This section provides the methods used to run unit tests.

Execute all module tests at once

The system automatically executes all test runs (test cases) in every unit test in a module and displays a summary of results.

To run all module tests, click the **Run All Tests**  icon above the module tree in Rules Editor.

Test results display resembles the following sample:

Number of Tests: 5 **1 tests failed**

Driver Risk Test 3 test cases

Driver	Expected	Result
+ Driver	Standard Risk Driver	✓ Standard Risk Driver
+ Driver	High Risk Driver	✗ Standard Risk Driver
+ Driver	High Risk Driver	✓ High Risk Driver

Driver Age Type Test 2 test cases

Driver	Expected	Result
+ Driver	Standard Driver	✓ Standard Driver
+ Driver	Young Driver	✓ Young Driver


Driver Eligibility Test 3 test cases

Driver	Age	Expected	Result
+ Driver	Standard Driver	Eligible	✓ Eligible
+ Driver	Young Driver	Eligible	✓ Eligible

Figure 66: Results of running all project tests

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

By default, test results are displayed with 5 test tables (unit tests) per page. If you want to change this

setting of 5 per page, click the arrow to the right of the **Run All Tests**  and choose a desired number of “Tests per page”. There is another mechanism to change the setting of 5 per page – the same setting options are displayed on the top of the window after executing all tests. The picture below provides an illustration

Tests: 13 **1 failed** 5 per page Failures Only ☒ All first

1
5
20
All

BankRatingGro... cases **2 failed**

Bank Rating	Expected	Actual	Result
0.7	R2	R3	✗
0.85	R3	R2	✗

BankLimitIndexTest 1 test cases

BankRatingTest 1 test cases

CapitalAdequacyScoreTest 4 test cases

DynamicIndexTest 1 test cases

« Prev 1 2 3 Next »

« Prev 1 2 3 Next »

Figure 67: Number of tests per page setting

The setting is stored for each user individually.

Execute tests for a single table

To execute all test runs for one particular rule table, in Rules 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 68: 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 69: 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 70: 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:

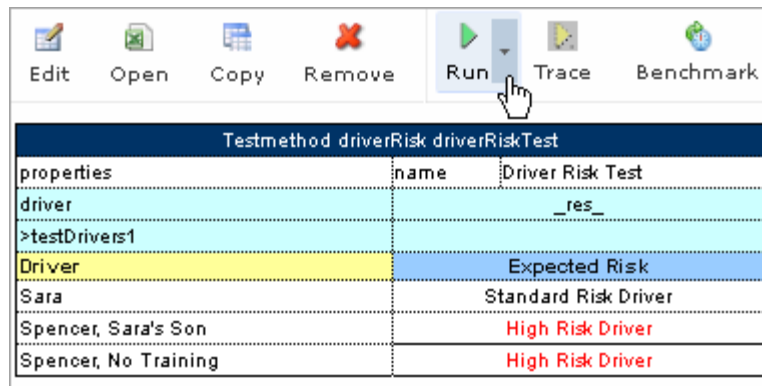


Figure 71: Running test cases with selection for a Test table

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

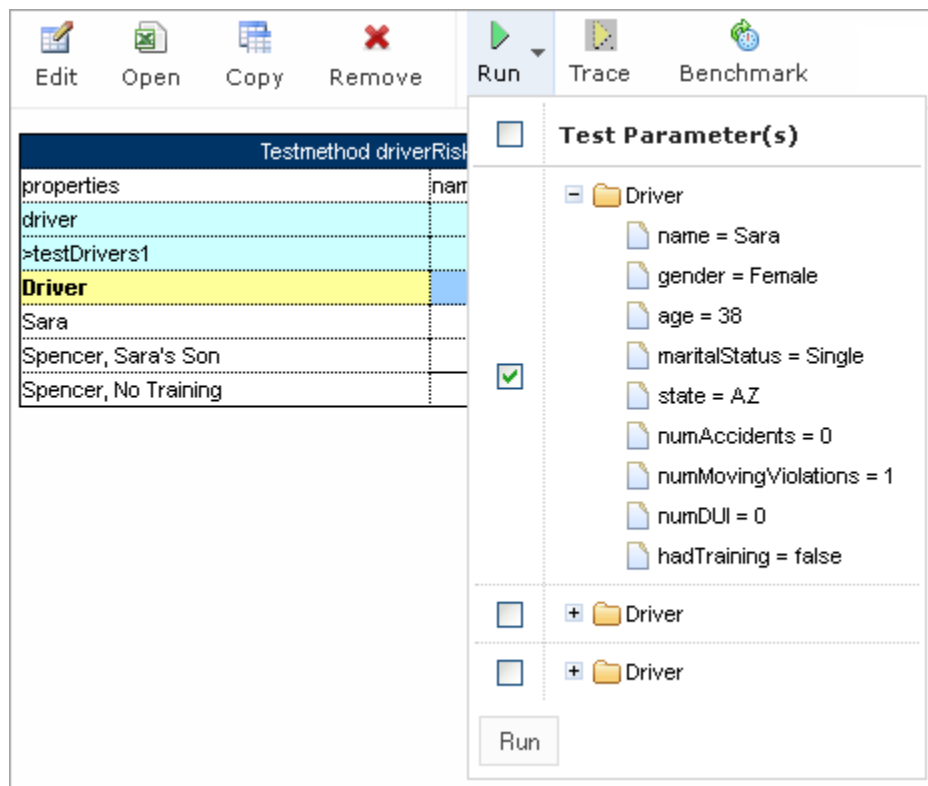


Figure 72: Select test cases to be executed

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


Driver Risk Test 3 test cases		
Driver	Expected	Result
+ Driver	Standard Risk Driver	✓ Standard Risk Driver
+ Driver	High Risk Driver	✗ Standard Risk Driver
+ Driver	High Risk Driver	✓ High Risk Driver

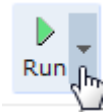
Figure 73: Result of selective testing

Displaying failures only

There are cases when a user wants to examine results of failed test cases only. For example, the project contains a test with more than 50 test cases and the user needs just to know whether project rules are operating correctly, it means all test cases are passed. If a user runs the test they will get a huge table of results which will be difficult to review and find failures to correct the rule or case. For such situations WebStudio provides option to display failed test cases only.

By default, all test results are displayed. To change this setting, there are multiple options:

- 1) Click the arrow to the right of the **Run All Tests**  and check “Failures only” in an displayed popup form.
- 2) Select the Test table. Then navigate to the **Run** button above the table and click the arrow



- 3) Check “Failures only” setting which is appeared on the top of the window after executing all tests at once (see Figure 74: Number of tests per page setting).

Additionally, the user can limit the number of failed test cases to be shown for one unit test. For example, the user is testing rules iteratively – he/she is interested just in first several failures in order to analyze the first ones, correct and execute tests again (sequentially correcting errors). To do this, change “All” on an appropriate value next to “Failures per test” label (for ways 1 and 2) or “first” label (for way 3). The setting is available only if “Failures only” is checked:

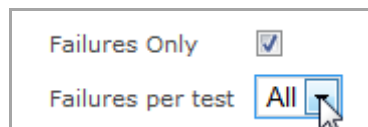


Figure 75: Settings for displaying failed test cases only

The settings are stored for each user individually.

Creating New Test

WebStudio provides convenient way to create a new test table.

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

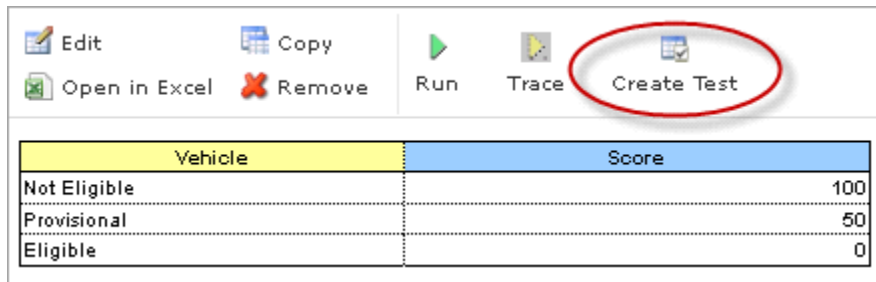


Figure 76: Create new test table

Click the **Create Test** 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
- All Rule tables with the possibility of specifying input parameters and tracing any method
- Run Method tables with preset parameters

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.

Using the tracing functionality you can check the result of each step of the rule and how the result was obtained without creating test cases. For that, follow the steps below.

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:

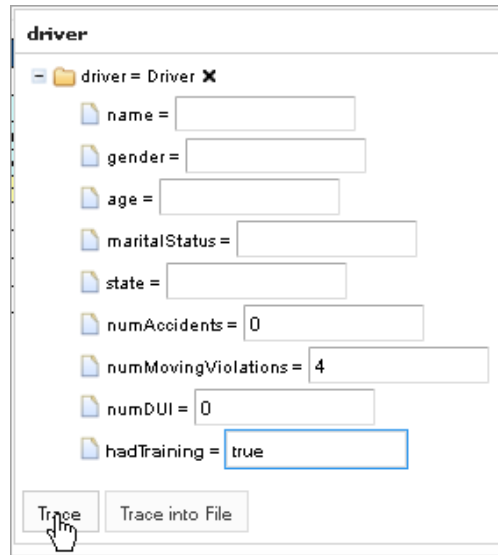


Figure 77: Tracing a rule for a rule table

3. Click the **Trace** button.

If you have a set of test cases and want to check the result of each step of the rule and how the result was obtained, you can trace the Test table as follows.

1. Open the desired Test table and hover the mouse pointer over the **Trace** button.
2. Click the small right-hand black arrow to open a popup with test cases to be traced.

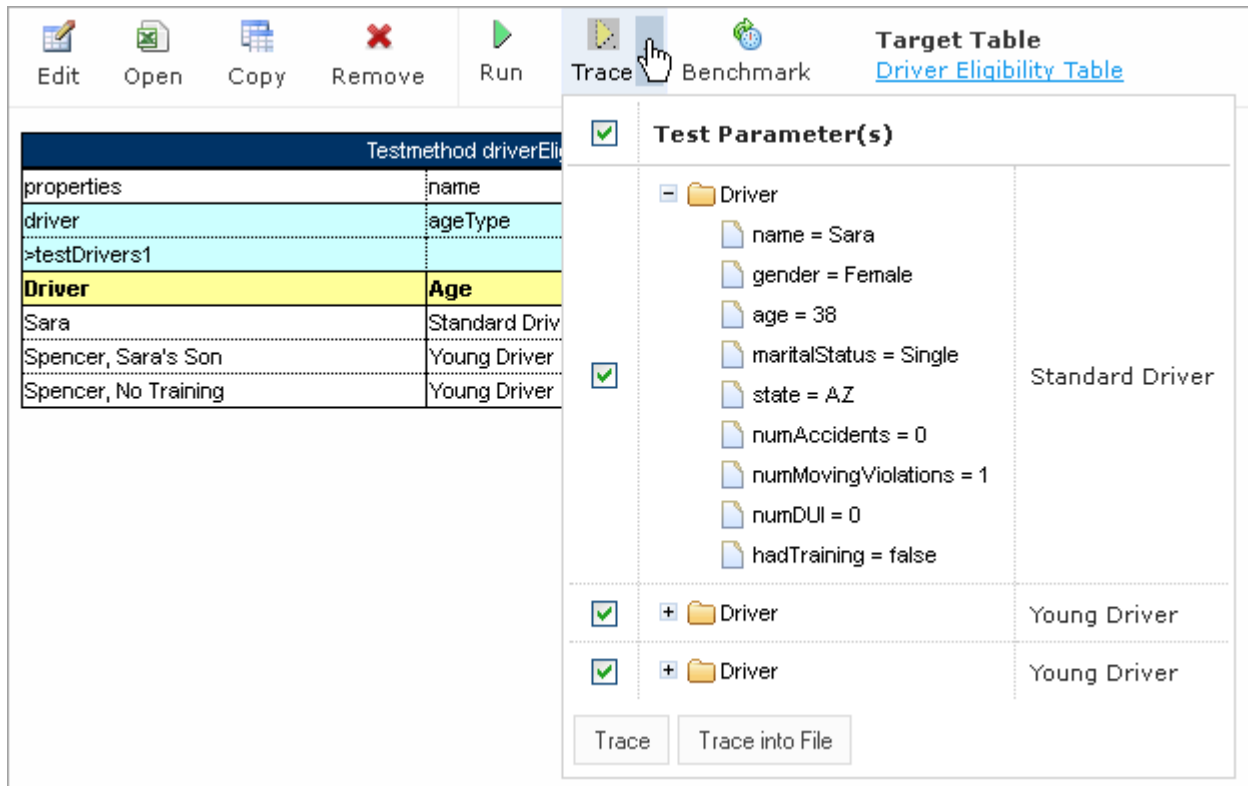


Figure 78: Run tracing for a Test table

- In the popup, select or deselect the test cases as you need. By default, all the cases are selected. You can also check or uncheck all test cases by using the checkbox on the left of **Test Parameter(s)**.
- Click **Trace** to start the process. The system displays the tracing results in a separate browser window as illustrated below.

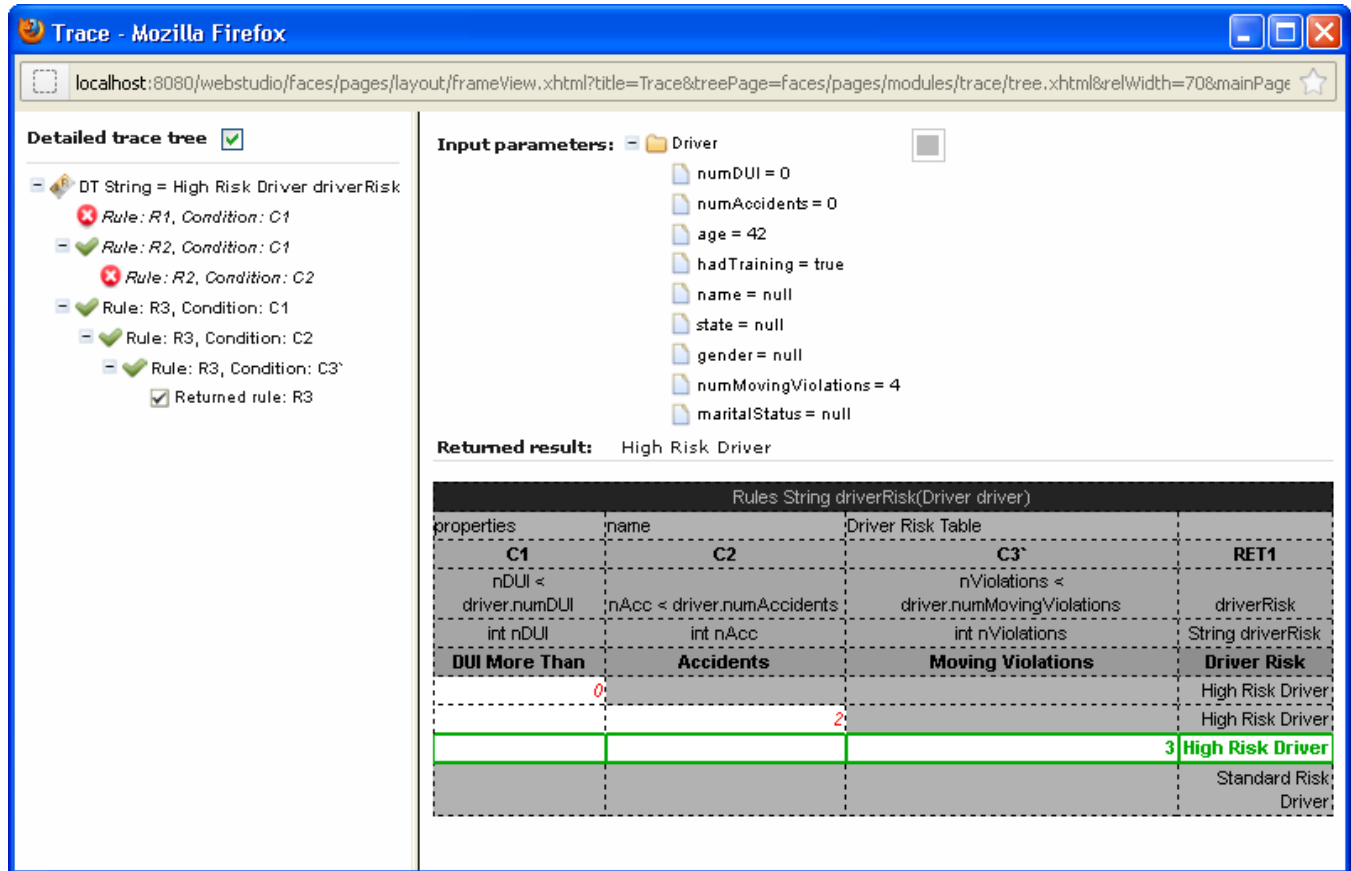


Figure 79: Tracing results

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. This option can only be used for a Decision table or if a Decision table is used in a complex rules.

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.

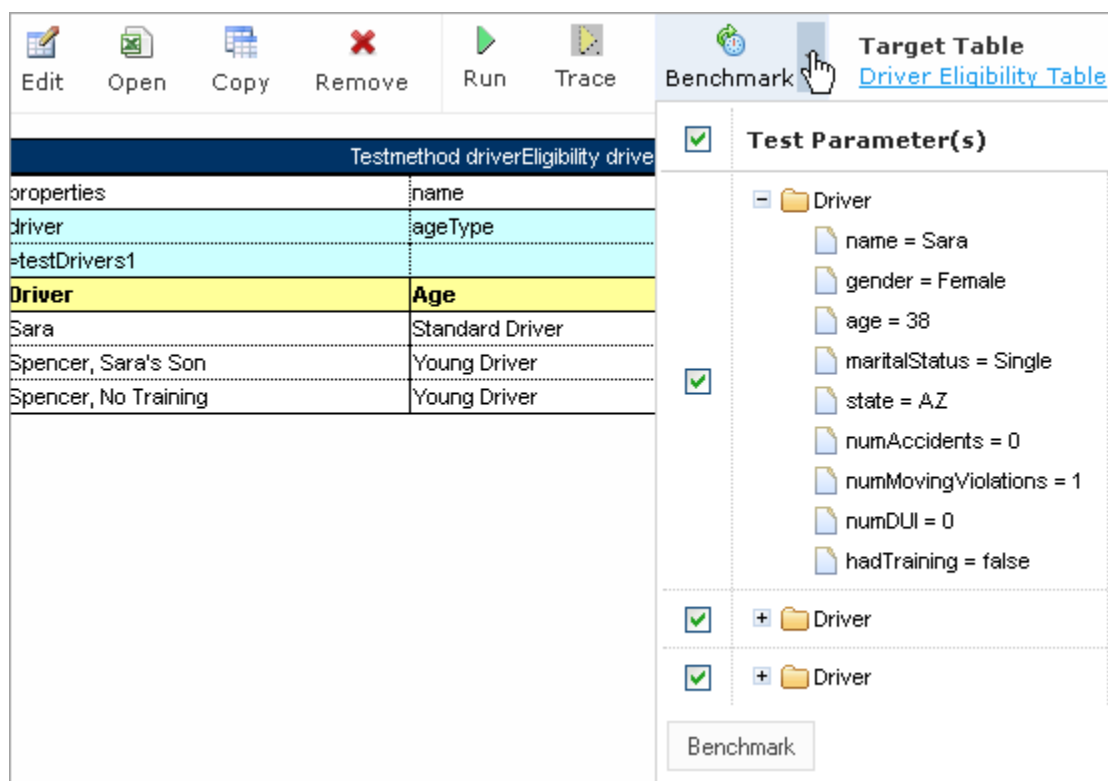


Figure 80: Controls for measuring performance

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. Select or deselect the test cases as you need. By default, all the cases are selected. You can also check or uncheck all test cases by using the checkbox on the left of **Test Parameter(s)**.
4. 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									
	Name	Parameters	Runs(ms)	Runs/sec	Unit Type	Units	Unit(ms)	Units/sec	<input type="checkbox"/>
1.	Driver Risk Test 3 test cases		2,14	467,20	Test Units	3	0,713	1 401,59	<input type="checkbox"/>
2.	Driver Risk Test 3 test cases		1,82	548,58	Test Units	3	0,608	1 645,74	<input type="checkbox"/>
3.	Driver Eligibility Test 3 test cases		1,39	718,51	Test Units	3	0,464	2 155,53	<input type="checkbox"/>
<div>Compare</div> <div>Delete</div>									

Figure 81: 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.	Driver Risk Test	1 401,59	3	1,54
2.	Driver Risk Test	1 645,74	2	1,31
3.	Driver Risk Test	2 155,53	1	1,00

Figure 82: 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 Repository](#)
- [Filtering the Project Tree](#)
- [Creating Project in Design Repository](#)
- [Opening a Project](#)
- [Closing a Project](#)
- [Editing a Project](#)
- [Saving a Project](#)
- [Defining Project Dependencies](#)
- [Modifying a Project](#)
- [Copying a Project](#)
- [Removing a Project](#)
- [Deploying Projects](#)
- [Comparing Project Versions](#)
- [Exporting a Project](#)
- [Unlocking a Project](#)






Browsing Design Repository

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

Categories in the project tree	
Category	Description
Projects	Contains OpenL Tablets rule projects.
Deploy Configurations	Contains deploy configurations for deploying rule projects to Production Repository. For information on using deploy configurations, 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 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 moved to the Editing status for modification.

Project icons in repository editor							
Icon	Description						
	Project is edited by the current user. It is copied to user's workspace and can be modified. Other users cannot edit the project. To save changes, the project must be saved.						
	Project is closed by the current user but edited by another user (Closed - Locked). Current user cannot edit the project.						
	Project is opened by the current user but edited by another user (Opened - Locked). Current user cannot edit the project but can browse the project in the Rules Editor.						
	Project exists only in user's workspace but not in Design Repository (Local). Other users do not see this project. User can delete the project or import it into Design Repository as described in the Creating Projects in Design Repository section.						
	Project is marked for deletion. In OpenL Tablets WebStudio, deletion of a project takes place in the following phases:						
<table> <tr> <th>Phase</th><th>Description</th></tr> <tr> <td>Deleting a project</td><td>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.</td></tr> <tr> <td>Erasing a project</td><td>Deleted project is permanently removed from Design Repository. After this phase, the project cannot be restored. For information on erasing a project, see Erasing a Project.</td></tr> </table>		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 Repository. After this phase, the project cannot be restored. For information on erasing a project, see Erasing a Project .
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 Repository. After this phase, the project cannot be restored. For information on erasing a project, see Erasing a Project .						

Filtering the Project Tree

You can filter projects in the Repository editor the same way as in the Rules Editor. To filter projects by name, just enter the name in filter text box. All projects matching the name will be displayed in the **Projects** list.

An advanced filter can be also applied to the project tree so that only files of particular types are displayed.

1. Click the **Advanced Filter** icon to the right of the filter text box.
2. In the **Advanced 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**.

Creating Projects in Design Repository

OpenL Tablets WebStudio allows users to create new rule projects in the Design Repository in one of the following ways:

- [Create an new rule project from template](#)
- [Create a rule project from Excel file\(s\)](#)
- [Create a rule project from zip archive](#)
- [Import a rule project from workspace](#)

Whatever the way used, new projects will be created in the **Editing** status that means they are open and can be modified.

Create a Project from Template

This is the easiest way to create a rule project in the Design Repository that should preferably be used for demonstration or introductory purposes.

While creating a project from template, you can use the following template types.

Simple Templates that include:

- Sample Project — a very simple project which consists of one rule table (and hence, one Excel file)
- Empty Project — allows you to create a project with empty Excel file. You can then open the project and create tables as you need.

Examples — provide several quite simple projects demonstrating how OpenL Tablets can be used in various business domains.

Tutorials — a number of projects designed to familiarize you with OpenL Tablets step-by-step, taking you from simple features and concepts to more complex ones.

You can use projects represented as Examples and Tutorials not only to learn how they are organized and work but also to create your own projects from them.

To create a new project from template, proceed as follows:

1. In the top line menu, click **Create Project**.
The **Create Project from** window appears.
2. Clicks the **Template** tab (this tab is normally selected by default). All project templates are organized into three areas: Simple Templates, Examples and Tutorials described above in this topic.
3. Navigate to the desired template and click its name. The name appears in the **Project Name** field. An example below demonstrates creating a Simple project.

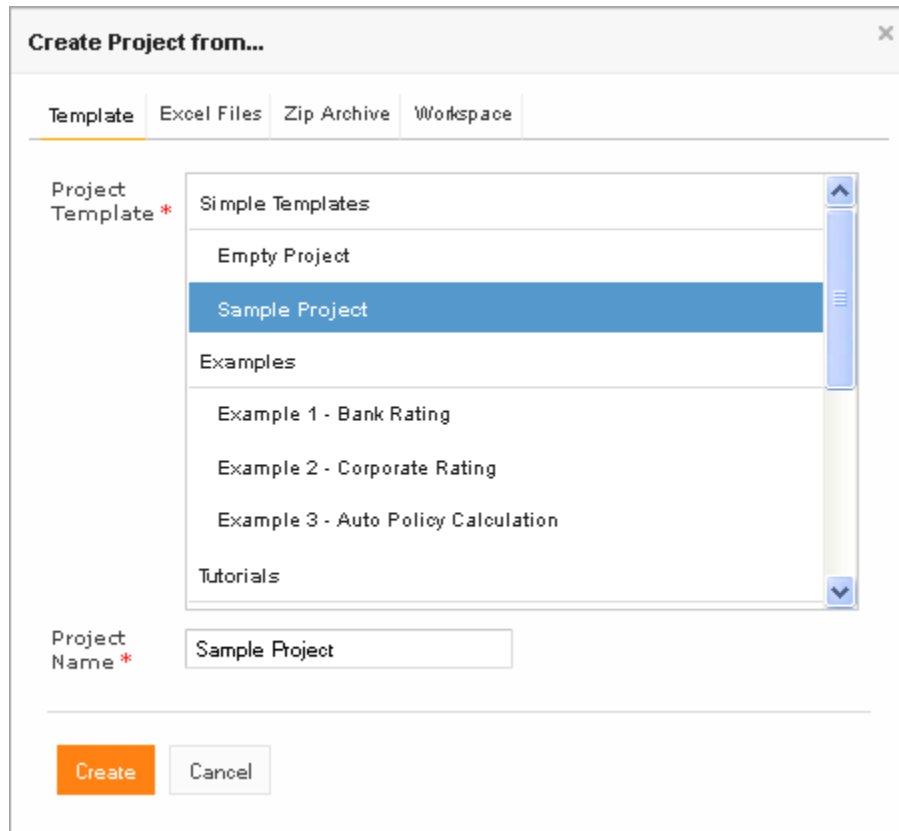


Figure 83: Creating Simple project **from** template

You can use the name provided or enter your own if you want.

4. Click **Create**. A new project is created in Design Repository. Initially, project structure corresponds to selected project template but can be constructed manually.
5. To construct the project structure, add folders and upload files as described in [Modifying Project Contents](#).

Create a Project from Excel File(s)

You can create a rule project in the Design Repository by loading one or more Excel files that contain OpenL rule tables or entire rule projects. For that follow the steps below.

1. Click **Create Project** in the top line menu.
2. In the **Create Project from** dialog, click the **Excel Files** tab. The **Create Project** dialog will look as follows:

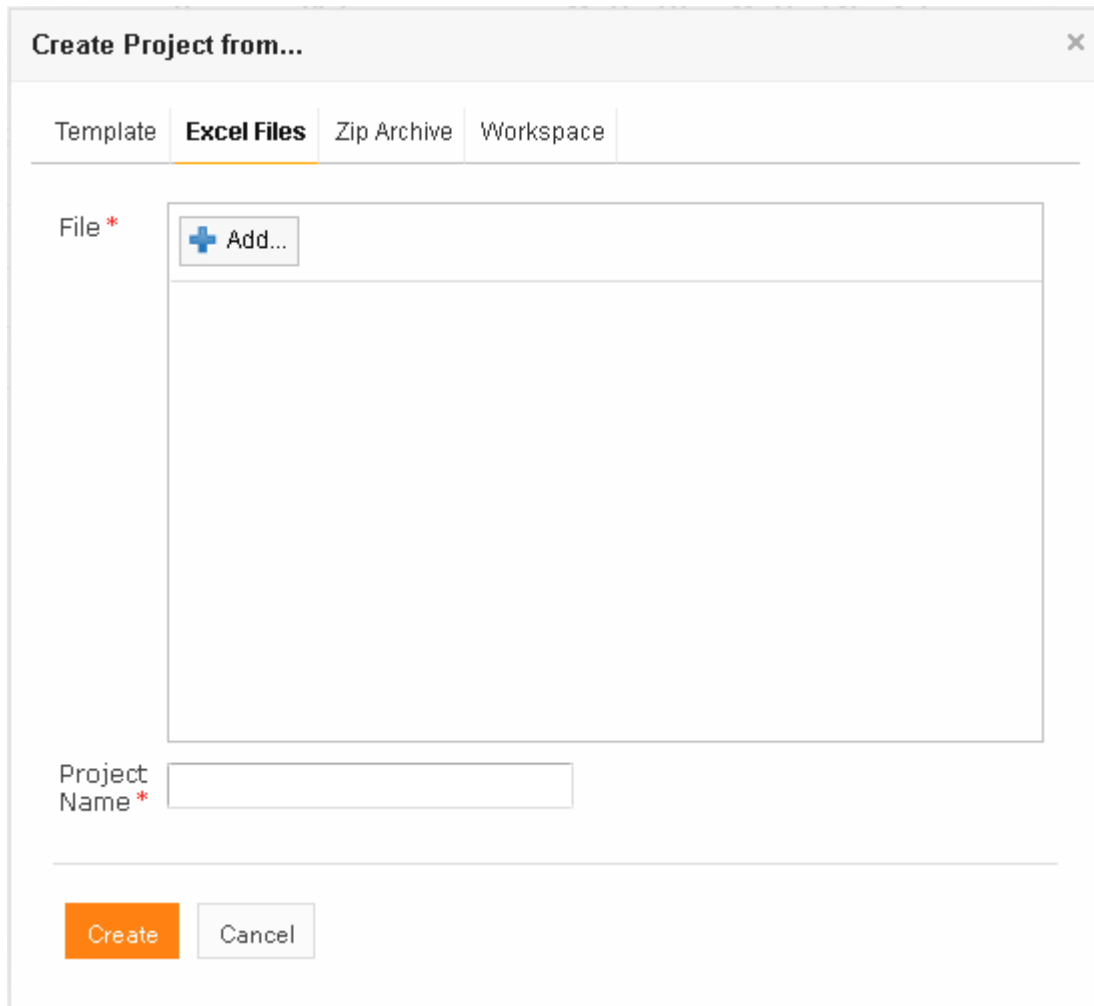


Figure 84: Creating a project from Excel file

3. Click the **Add** button, locate the desired Excel file in a file system and click **Open**.
4. If required, repeat the previous step to add more file(s) for your project. All the files will be listed in the **File** area.

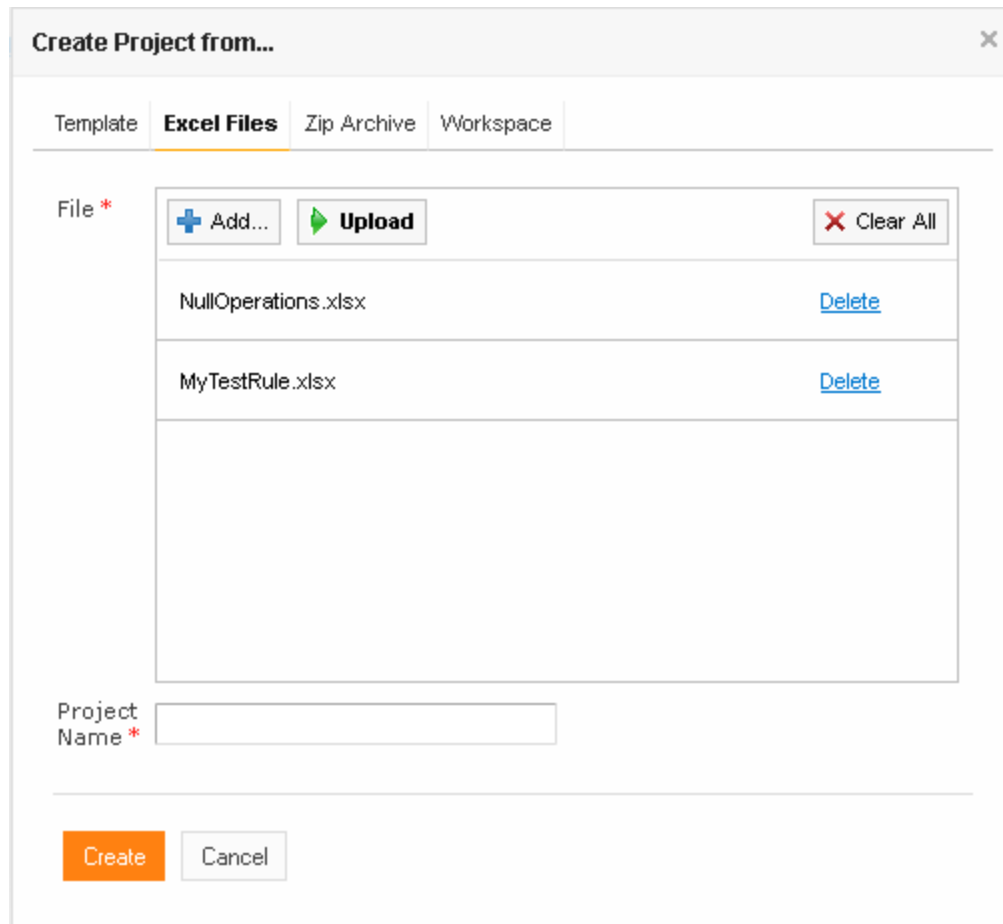


Figure 85: Excel files ready to be uploaded into Design Repository

You can remove any file from the list by clicking the corresponding **Delete** link or delete all of them by clicking **Clear All**.

5. After adding all the files you want, click **Upload** to load the files into the repository. You can also upload each file separately but we do not recommend you to follow that way.
6. In the **Project Name** field, enter the name by which the project must be represented in the Design Repository.
7. Click **Create** to complete.

Create a Project from ZIP Archive

OpenL Tablets WebStudio provides a control for loading rule project(s) archived in a ZIP file into Design Repository. The procedure is similar to creating a project from Excel files described above although there are a few differences.

1. Click **Create Project** in the top line menu.
2. In the **Create Project from** dialog, click the **Zip Archive** tab.
3. Click the **Add** button, locate the desired zip archive and click **Open**. You will see the file in the **File** area.

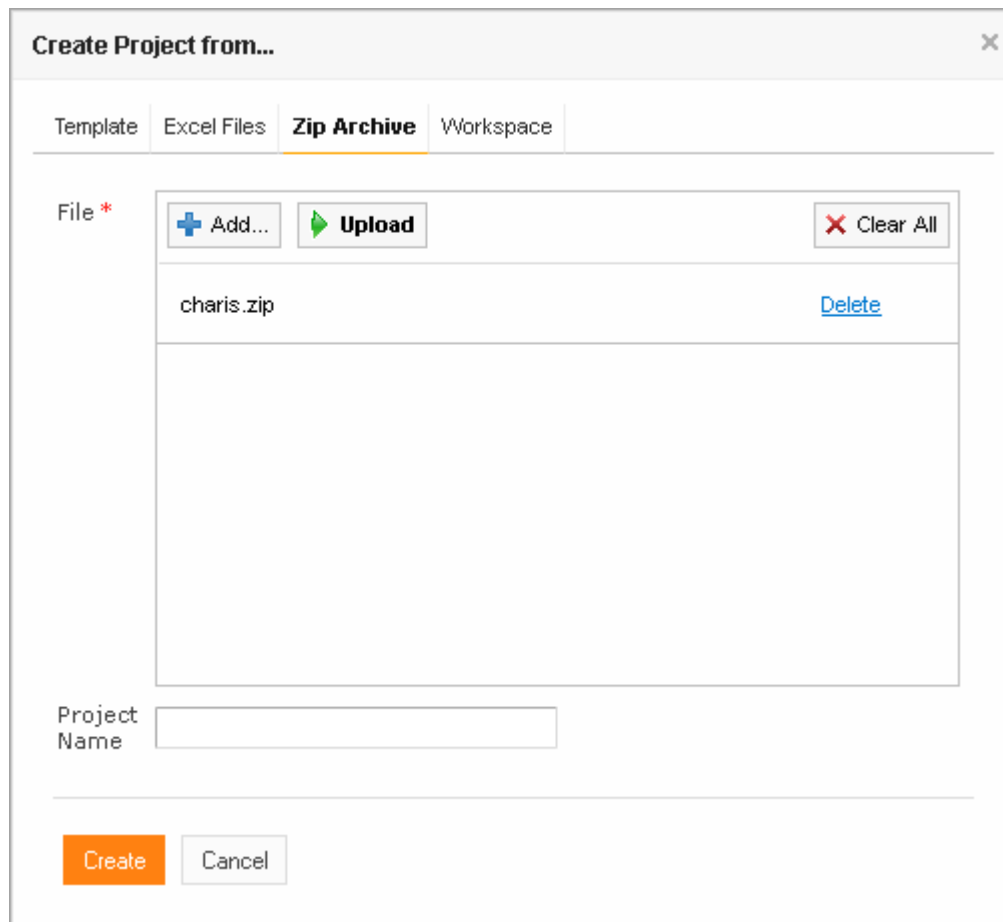


Figure 86: Creating a project from ZIP file

4. Click **Upload** to proceed. The file name appears in the **Project Name** text box.
5. Click **Create** to complete.

Import a Project from Workspace

You can create a new project in the Design Repository by loading a project with the **Local** status from user workspace.

1. Click **Create Project** in the top line menu.
2. In the **Create Project from** dialog, click the **Workspace** tab. The system displays rule projects available in your workspace:

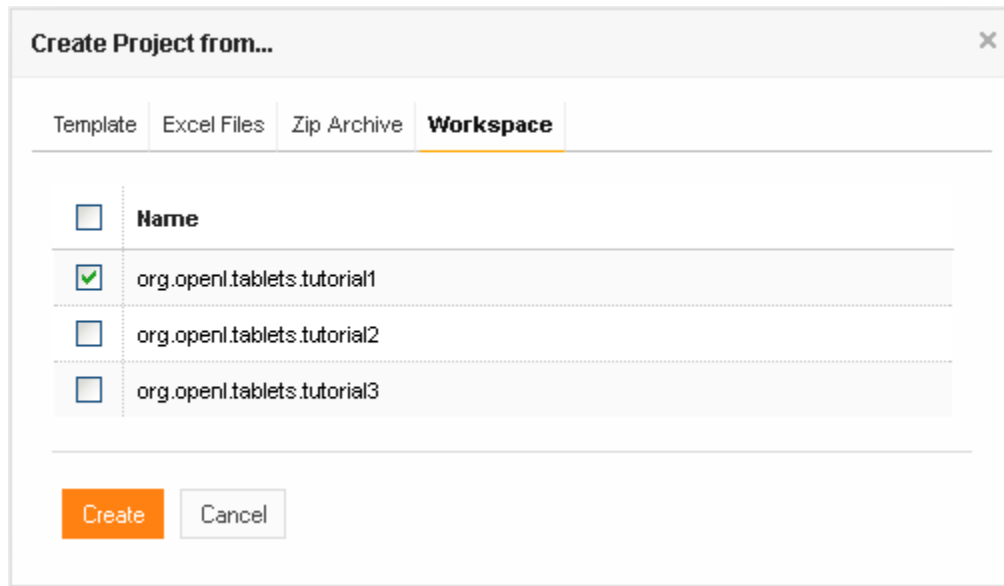


Figure 87: Creating a project from Workspace

3. Select checkboxes for projects to be uploaded and click **Create** to complete.

In case project(s) are loaded successfully appropriate notification will be shown.

Opening a Project

An opened project is copied to user's workspace and becomes available for selection in Rules Editor. An opened project **cannot be modified**, it must become Editable as described in [Editing 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 the latest revision of project.
Open Revision	Displays window where user can specify which project revision must be opened.

You can also open any project revision from the **Revisions** tab. For that, select the project in the **Projects** tree and proceed as follows:

1. Click the **Revisions** tab. A list of revisions appears.

Revision	Modified By	Modified At	Comment	Action
2	snm	10/18/2012	Effective/Expiration dates added.	
1	snm	10/17/2012		
0	system	10/17/2012		

Figure 88: List of project revisions

- Navigate to the revision you want to open and click the corresponding magnifier icon in the **Action** column.
- Click **OK** in the information message.

Closing a Project

Closing a project deletes it from user's workspace. No changes you have made to the project will be applied and stored. From that point, the project is not available for selection in the Rules Editor. Users can still browse closed projects in the Repository Editor.

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

Editing a Project

An editable project (the project status is **Editing**) is copied to user's workspace, becomes available for selection in Rules Editor and can be modified. Only editable project can be modified in the Rules Editor. To apply changes made to a project, the project must be saved as described in Saving a Project.

To make a project editable, in the project tree, select the project and, in the right pane, click **Edit**.

The latest project revision is becomes editable even if the user previously opened an older project revision.

Alternatively, an opened project can be made editable directly from Rules Editor as described in [Editing and Saving a Project](#).

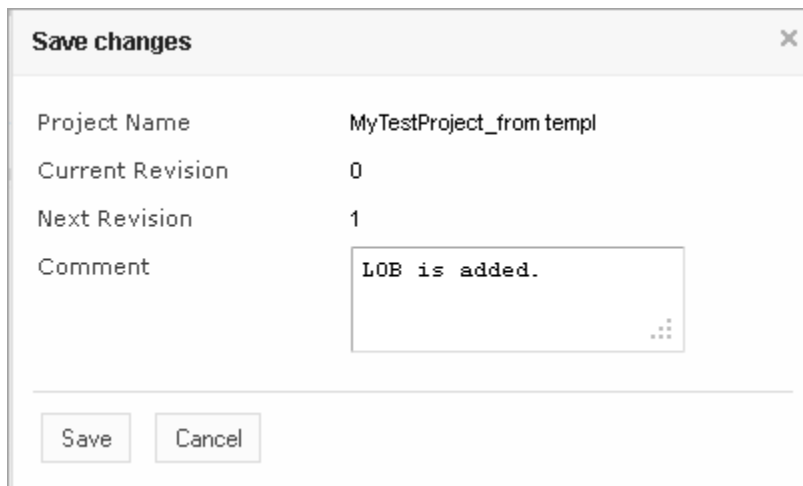
Saving a Project

A modified project is saved and copied from the user's workspace to Design Repository as a new revision.

To save a project, proceed as follows:

- In the project tree, select the project, and, in the right pane, click **Save**.

The **Save changes** window appears.



A dialog box titled "Save changes" with a close button (X) in the top right corner. It contains four fields: "Project Name" with the value "MyTestProject_from templ", "Current Revision" with the value "0", "Next Revision" with the value "1", and "Comment" with the text "LOB is added." in a text area. At the bottom are "Save" and "Cancel" buttons.

Figure 89: Save changes in a project

The number of a revision is updated automatically and specified in the **Next Revision** field.

2. Enter comments if you need and click **Save**.

An editable project can be saved and closed directly from Rules Editor as described in [Editing and Saving 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 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 editable, make it editable as described in [Editing a Project](#).
2. In the project tree, select the project, and, in the right pane, select the **Dependencies** tab.


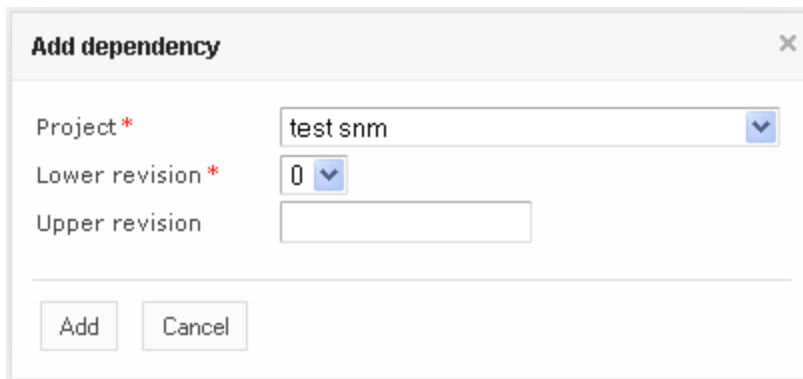
Properties	Revisions	Elements	Dependencies
Project	Revisions	Actions	
test snm	0 - ...		
<div>Add</div>			

Figure 90: Project 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.

A dialog box titled "Add dependency" with a close button (X) in the top right corner. It contains three input fields: "Project *" with a dropdown menu showing "test snm", "Lower revision *" with a dropdown menu showing "0", and "Upper revision" with an empty text box. At the bottom, there are two buttons: "Add" and "Cancel".

Add dependency	
Project *	test snm
Lower revision *	0
Upper revision	
<div>Add Cancel</div>	

Figure 91: Defining a new dependency

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

Modifying a Project

For an editable project, users can modify the project's properties and content.

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 displayed in the **Properties** tab when a project is selected.

Properties	Revisions	Elements	Dependencies
Name	org.openl.tablets.tutorial1		
Revision	1		
Status	Editing		
Created At	10/18/2012		
Created By	snm		
Add Property			

Figure 92: Project properties

Some properties such as Name, Crated At / Created By, etc are updated automatically by the system, and users cannot edit them in the WebStudio UI. The others can be specified for editable projects using the **Add Property** link as described in the next section.

Adding Business Dimension Properties

From OpenL Tablets WebStudio 5.9.4, users can add *Business Dimension* properties predefined in a drop down list.

Follow the steps below to add a Business Dimension property to a project.

1. Select the project in the **Projects** tree.
2. In the **Properties** tab, click the **Add Property** link.
3. Select the desired property from the drop down list and click **Add**:

Figure 93: Add a project property

4. Specify the value as you need and click the green tick on the right to save your changes:

Figure 94: Save a new project property

Adding Custom Properties

The list of project properties available in the drop down list can be extended by adding *Custom Properties*. To add Custom properties, 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, for example:
`props.use = attribute1, attribute6, attribute7, attribute13`

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

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

3. Uncomment the attributes to be added as Custom properties (remove appropriate '#') and define the properties' names, for example, as shown in this snippet:

```
props.attribute1 = Additional String Property
props.attribute7 = Additional End Date
props.attribute13 = Additional Number
```

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

For example, if properties 1, 6, 7 and 13 are added, they appear in the drop down list as follows:

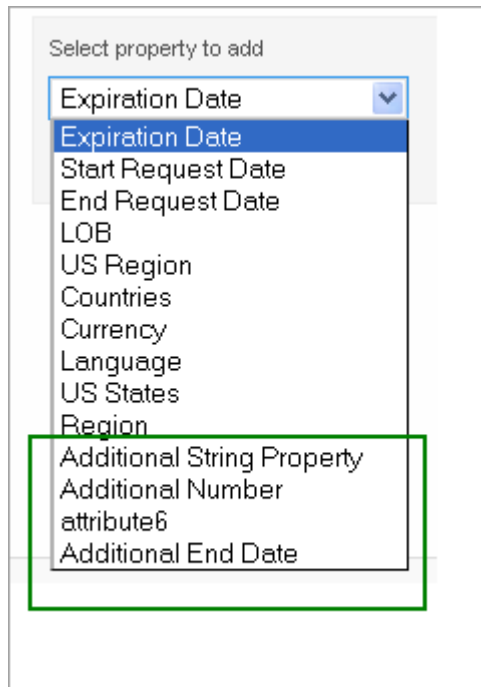


Figure 95: Customer properties 1, 6, 7 and 13 are shown in the green box

You will add Customer properties in the same way as described in the [Adding Business Dimension Properties](#) section above. The next Figure illustrates properties 1, 6, 7 and 13 as they appear in the project UI.

Properties	Revisions	Elements	Dependencies
Name	Module1		
Revision	2		
Status	Editing		
Created At	10/17/2012		
Created By	snm		
Modified At	10/18/2012		
Modified By	snm		
Business Dimension			
Effective Date	10/01/2012		
Custom Properties			
Additional String Property	New Headway		
Additional Number	45.567		
attribute6	10/23/2012		
Additional End Date	10/23/2012		
Add Property			

Figure 96: Project UI with properties 1, 6, 7 and 13 added

Modifying Project Contents

This section describes modifying the physical structure of the 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 editable, make it editable as described in [Editing 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 editable, make it editable as described in [Editing a Project](#).
2. In the project tree, select the folder where the file should 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.

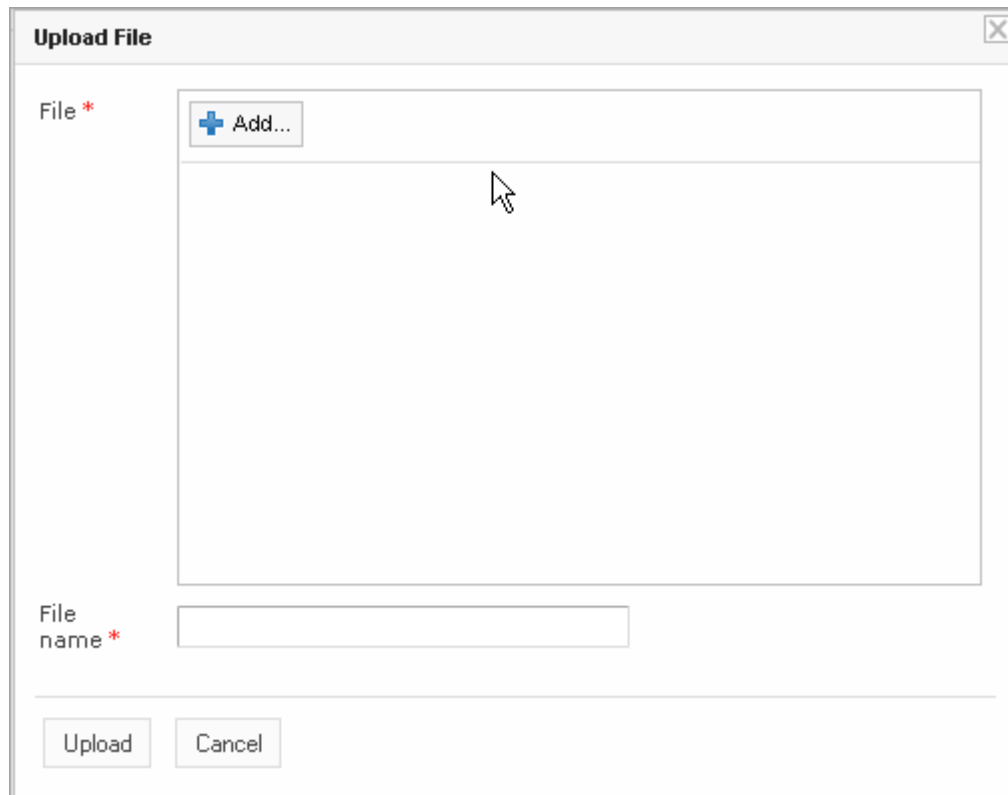


Figure 97: Uploading a file

4. Click **Add** in the **File** area and select the file to be uploaded.
5. Click the upper **Upload** button (with a green arrow).
6. In the **File name** field, enter or modify the name of the file to be used in the Design Repository.
7. Click the **Upload** button at the bottom.

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 editable, make it editable as described in [Editing a Project](#).
2. Perform one of the following steps as required:
 - Expand the project tree, select the folder or file to be deleted and, in the right pane, click **Delete**.

Add Folder
Upload File
Delete


Properties
Revisions
Elements

Folder Name
New Root folder

Business Dimension Properties

Region
EU

Figure 98: Deleting a project element

- If you want to delete an element inside the parent folder, select that folder, click **Elements** to expand the folder and then click **Delete**  at the right of the item to be deleted.

























Properties	Revisions	Elements	Dependencies
Type	Name	Actions	
	.settings		
	META-INF		
	bin		
	build		
	gen		
	rules		
	src		
	.classpath		
	.project		
	Generate org.openl.tablets.tutorial1 Interface.launch		
	build.properties		
	rules.xml		


Figure 99: Deleting project elements from the **Elements** tab

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

Copying a Project

Copying a project creates a new project with identical contents and a different name in the Design Repository. This function can be used for copying local projects to the Design Repository with a different name.

To copy a project, proceed as follows:

1. Perform one of the following steps as required:
 - In the **Projects** tree, select the desired project and, in the right pane, click the **Copy** button.
 - Click **Projects** in Navigator to get a list of projects, navigate to the project you want to copy and click the corresponding **Copy** item  at the right.
2. In the **Copy Project** window, enter the new project name and click **Copy**. Your new project appears in the list of projects in the **Closed** status.

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 Repository. All users can see that a project is deleted. Physically, it still remains in Design Repository.

NOTE: Projects that were not uploaded to the Design Repository (they have the **Local** status) will be removed physically and cannot be restored.

To delete a project, proceed as follows:

1. Perform one of the following steps as required:
 - In the **Projects** tree, select the project and, in the right pane, click **Delete**.
 - Click **Projects** in Navigator to get a list of projects, navigate to the project you want to remove and click the corresponding **Delete** item  at the right.
2. In the confirmation window, click **Delete** or **OK**.

Deleted projects (except for those in the **Local** status) can be restored by using the **Undelete** button.

To make deleted projects visible uncheck the **Hide deleted projects** checkbox in the filter pop up window which appears after clicking the **Filter** button above the **Projects** tree and click **Apply**.

If you want to restore a deleted project:

1. Navigate to the deleted project in the **Projects** tree.
2. Click the **Undelete** button in the right pane.
3. Click **Undelete** in the confirmation window.

Erasing a Project

Erasing a project permanently removes it from the Design 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 **Projects** 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 Repository.

The following topics are included in this section:

- [Creating a Deploy Configuration](#)
- [Defining Projects to Deploy](#)
- [Deploying a Deploy Configuration](#)
- [Opening Deployed Configurations](#)
- [Redeploying Projects](#)

Creating a Deploy Configuration

Deployment to Production Repository is performed by using deploy configurations. A Deploy configuration is a list of rule projects and specific project revisions to be deployed together to Production Repository. Deploy configurations are useful for recording the history of project deployments.

Deploy configurations are listed in the **Deploy Configurations** tree. Just like rule projects, deploy configurations are stored in Design Repository and can be versioned.

To create a deploy configuration, proceed as follows:

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

Defining Projects to Deploy

A Project to Deploy is a reference to one specific revision of a rule project to be included in the deploy configuration. Project to Deploy must be added to the deploy configuration specifying which rule projects and project revisions are deployed.

To add a new project to deploy to the deploy configuration, proceed as follows:

1. If the deploy configuration is not editable, make it editable as described in [Editing a Project](#).
2. In the **Deploy Configuration** tree, select the deploy configuration and, in the right pane, select the **Projects to Deploy** tab.



Properties	Revisions	Projects to Deploy		
Selected	Name	Revision	Message	Actions
<input type="checkbox"/>	Module1	2		
<input checked="" type="checkbox"/>	test snm	1		
Add		Open		

Figure 100: Deploy Configuration with projects to deploy

The **Projects to Deploy** tab displays existing projects to deploy of the selected deploy configuration.

- To add a new project to deploy, click **Add** and specify the project and revision to be included in the deploy configuration.
- Repeat this procedure to add as many projects as required.

Deploying a Deploy Configuration

To deploy a deploy configuration, save it and click **Deploy**.

Specified projects are deployed to Production Repository and a deployment message is displayed.

Project 'Test Deploy Conf' successfully deployed with id 'Test Deploy Conf#0' to repository 'Production'

Figure 101: Deployment message

Opening Deployed Configurations

Deploy configurations provide the means for tracking the deployment history of project revisions. OpenL Tablets WebStudio provides functionality for quickly opening the deployed configuration revisions. 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 revisions included in a deploy configuration, proceed as follows:

- In the **Deploy Configuration** tree, select the deploy configuration.
- In the right pane, select the **Projects to Deploy** tab.
- In the **Selected** column, select the check boxes for projects to be opened.
- Click **Open**.

The selected project revisions are opened in repository editor.

Redeploying Projects

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

To update related deploy configurations and redeploy a rule project, proceed as follows:

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

Note: The **Deploy** button is disabled if the selected project has the Local status or if it is edited.

The **Auto Deploy** window appears listing all existing deploy configurations whose latest revision contains a reference to the selected rule project. Deploy configurations marked for deletion are not displayed.

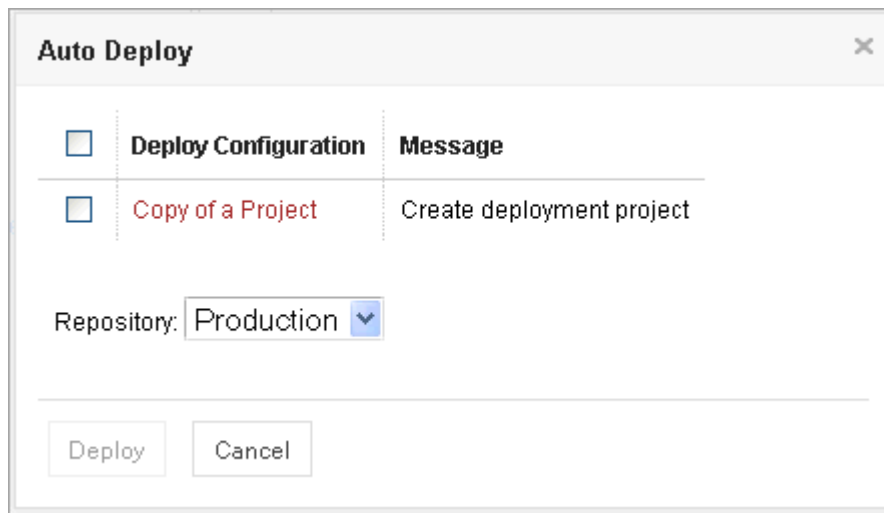


Figure 102: Deploying a project

The **Message** column displays the current status of displayed deploy configurations. If a particular deploy configuration cannot be deployed, the check box is gray. Following are possible reasons for a deploy configuration to be disabled:

- The deploy configuration is saved.
- The deploy configuration is locked by another user and cannot be updated.
- The deploy configuration is up to date and references the selected revision of the rule project.
- The deploy configuration references a revision of the rule project that is higher than the one currently opened.

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

3. Select check boxes for the deploy configurations that must be updated and deployed.
4. Click **Deploy**.

Update and deployment results are displayed in the user interface.

```
Deployment project 'org.openl.tablets.tutorial4' successfully updated
Project 'org.openl.tablets.tutorial4' successfully deployed with id: org.openl.tablets.tutorial4#0.0.4
```

Figure 103: Redeployment results

Comparing Project Revisions

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

To compare contents of the currently opened project revision with any other revision, 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.

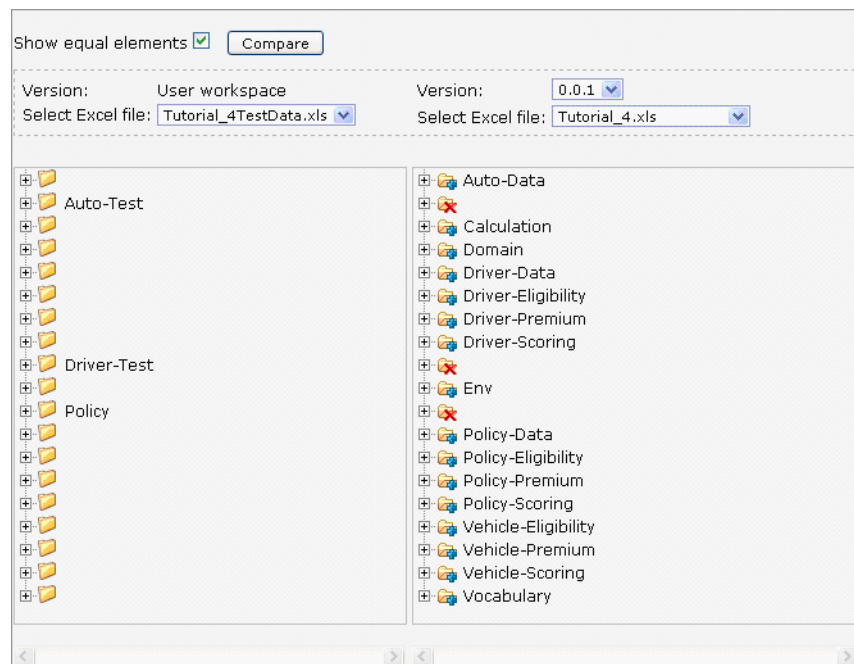


Figure 104: Comparing project versions

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.

Exporting a Project

To export any revision of a project from Repository editor:

1. In the project tree, select the project.
2. In the right pane, click **Export**.
3. In displayed window select a required project revision and click **Export**.

To export any revision of a file, proceed as follows:


1. In the project tree, select the project. If the project is not editable, make it editable as described in [Editing a Project](#).
2. Expand the project tree and select the file to be exported.
3. In the right pane, click **Export file**.
4. In displayed window select a required file revision and click **Export**.

NOTE: If the project is in the Local status these options are not available.

Unlocking a Project

WebStudio provides a function for a user to unlock a project which is edited and, therefore, locked by another user. Be aware that after unlocking, all unsaved changes made by another user will be lost and the project will be closed.

To unlock a project, proceed as follows:

1. Perform one of the following steps as required:
 - In the **Projects** tree, select the project and, in the right pane, click **Unlock**.
 - Click **Projects** in Navigator to get a list of projects, navigate to the project you want to remove and click the corresponding **Unlock** item  at the right.
2. In the confirmation window, click **OK**.

We recommend to grant permission to “Unlock” functionality only for administrators.

Using Administration Tools

This section explains how to view and control OpenL Tablets WebStudio system settings and also describes the user management procedure in the system. To perform administration tasks, click the Administration item in the top line menu. By default, the System Settings tab is displayed.

Managing System Settings

All System Settings are organized into the following groups: **Common**, **Repository** and **System**. To open the desired group, you should click the corresponding icon on the left.

The screenshot displays the 'WebStudio' administration interface. At the top, there are two tabs: 'System Settings' (active) and 'User Management'. Below the tabs, a left sidebar contains three icons: a gear for 'Common', a database cylinder for 'Repository', and a tablet for 'System'. The main content area is divided into three sections: 'User Workspace' with a 'Workspace Directory' field containing '/openl/user-workspace'; 'History' with a 'History Directory' field containing '/openl/project-history'; and 'Other' which includes a checked checkbox for 'Update table properties ('createdOn', 'modifiedBy' etc.) due editing:' and a 'Date Format' field containing 'MM/dd/yyyy'. At the bottom of the 'Other' section are two buttons: 'Restore Defaults' and 'Apply'.

Figure 105: 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** and restart the WebStudio. If you wish to restore the original settings, click the **Restore Defaults** button.

The following topics are included in this section:

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

Common Settings

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

User Workspace

Defines **Workspaces Directory** where the user's projects should be located.

History

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

Unlimited numbers of copies – If unchecked makes it possible to define the number of history files to be saved for a project.

To specify a number of history files:

1. Clean the **Unlimited numbers of copies** checkbox (it is selected by default).
2. In the **The maximum count of saved changes for each project** field enter the desired number (by default, it's set as 30).
3. Click **Apply** to save your changes.
4. Click **OK** in the confirmation dialog.

You can also clean all the history files for your project. For that:

1. Click the **Clean history** link.
2. In the **Clean projects history** form, select checkbox for a particular project, or select the "**Project name**" to clear history for all projects.
3. Click **Clean** to complete.

Other

Update table properties... - Indicates whether table properties controlled by the system should be updated and can be viewed in the WebStudio UI. If the option is unchecked, the information about who and when created or modified tables such as **Created By/On**, **Modified By/On** would not be added to tables' properties.

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

Repository Settings

The **Repository** section contains connection settings of Design and Production repositories:


- connection **name** which is displayed as a repository name in repository editor;
- connection **type** and repository **directory** path, which define a repository to be used as a datasource:

Type	Description
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.

- **login** and **password** if the repository is remote and secured. To set up a secure connection, select the “**Secure connection**” checkbox and fill in the login and password fields.

For more information on repository security refer to the *Configuring Private Key for Repository Security* section in the [OpenL Tablets Installation Guide](#).

The settings can be changed by editing on the tab or deleted by clicking the red cross .

Connection to a local production repository is configured by default. You can connect more Production repositories by clicking the **Connect To Production Repository** button, enter the repository parameters as needed and click **Connect**.

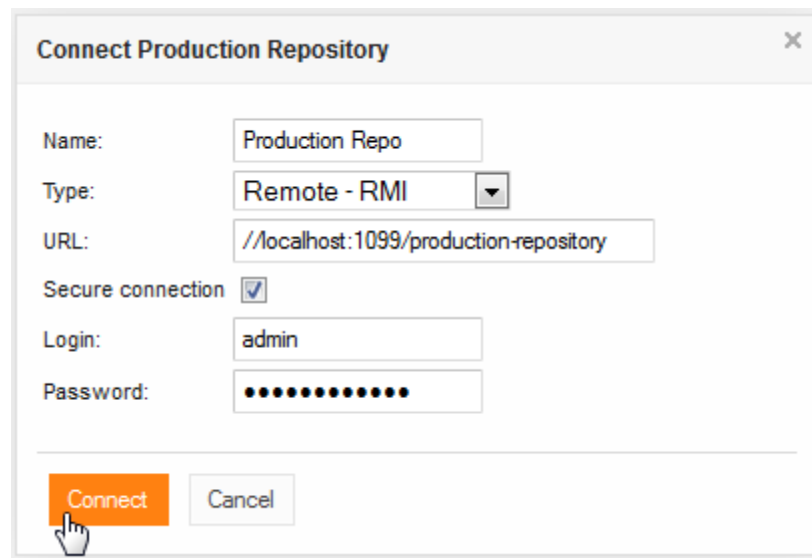


Figure 106: Connecting Production Repository

The **Repository** tab provides an option of creating a local production repository by clicking the **Create Production Repository** button, in the form enter new repository parameters and click **Create**.

Do not forget to click **Apply** to save your changes and refresh the page.

System Settings

The **System** tab provides the following options:

Custom Spreadsheet Type – indicates whether the `Custom Spreadsheet Type` Java system feature is turned on or off. By default the feature is turned off. To activate the feature, select the **Custom Spreadsheet Type** checkbox.

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

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

User Management

This section describes how to control user access in the OpenL Tablets WebStudio application based on users and user groups. All privileges in the system are assigned at a group level and will be granted to a particular user after he or she is included in a particular group.

You will manage users and groups from the **User Management** tab which, in turn, is divided into **Users** and **Groups & Privileges** tabs. Only members of the **Administrators** group have rights to manage users and groups in the WebStudio.

Manage Groups

In this section you will learn how to create, modify, and delete a user group with a certain set of privileges. The **Administrators** group cannot be deleted from the system.

1. From the **Administration** tab, click **User Management** on the top-left of the screen.
2. Click **Groups & Privileges** at the left. The system displays a list of groups similar to the following one:

User Management			
Name	Description	Privileges	
Viewers		View Projects	
Developers		Viewers Create Projects Edit Projects Erase Projects Delete Projects Create Tables Edit Tables Remove Tables	
Deployers		Viewers Deploy Projects Edit Deploy Configuration Create Deploy Configuration Delete Deploy Configuration Erase Deploy Configuration	
Testers		Viewers Run Tables Trace Tables Benchmark Tables	
Analysts		Viewers Developers Testers	
Administrators		NO RESTRICTIONS	

[Add New Group](#)

Figure 107: User groups in the **Groups & Privileges** tab

From here, you can create a new group, and edit or delete an existing one.

To add a new group:

1. Click the **Add New Group** link. The **Add New Group** form appears.
2. Enter the group name in the **Name** field.
3. Enter any useful information in the **Description** text box (optional).
4. In the **Privilege** area select the privileges as you need. You can also assign a set of privileges for the group by clicking the name of the group above the list of privileges (Developers, Deployers, etc). The **Viewers** group is selected for a new user group by default.

Add New Group

Name *

Super User

Description

For VIP users

Privilege *	Viewers	Developers	Deployers	Testers	Analysts	Administrators
<input checked="" type="checkbox"/> View Projects	✓	✓	✓	✓	✓	✓
<input checked="" type="checkbox"/> Create Projects		✓			✓	✓
<input checked="" type="checkbox"/> Edit Projects		✓			✓	✓
<input type="checkbox"/> Erase Projects		✓			✓	✓
<input checked="" type="checkbox"/> Delete Projects		✓			✓	✓
<input type="checkbox"/> Unlock Projects						✓
<input checked="" type="checkbox"/> Deploy Projects			✓			✓
<input checked="" type="checkbox"/> Create Deploy Configuration			✓			✓
<input checked="" type="checkbox"/> Edit Deploy Configuration			✓			✓
<input checked="" type="checkbox"/> Delete Deploy Configuration			✓			✓
<input type="checkbox"/> Erase Deploy Configuration			✓			✓
<input type="checkbox"/> Unlock Deploy Configuration						✓
<input checked="" type="checkbox"/> Create Tables		✓			✓	✓


Save

Cancel


Figure 108: Add a new user group with required set of privileges

5. Click **Save** to save your work.

To modify a User Group:

1. Locate the group you wish to change in the list of groups and click the **Edit** icon .
2. In the **Edit Group** form, you can change the group name, add or modify its description and change privileges as needed.
3. Click **Save** to complete.

To delete a User Group:

1. Locate the group to be deleted and click the red cross at the right: .

2. Click **OK** in the confirmation dialog.

Manage Users

Users get access to WebStudio functions by including them in particular group(s).

By default, there are the following users in OpenL Tablets WebStudio.

User name	User password	Groups
user	user	Viewers
u0	u0	Testers
u1	u1	Developers, Analysts
u2	u2	Viewers
u3	u3	Viewers
u4	u4	Deployers
a1	a1	Administrators

On the first start of the WebStudio you are provided with a1/a1 login/password pair that gives you Administrator's permissions. You can then set up your users in the WebStudio as you need. For information about the permissions of the groups, refer to the [Manage Groups](#) section above.

While creating a new user, you should include the user in at least one group.

1. From the **Administration** tab, click **User Management** at the top-left of the screen.
2. Click **Users** at the left. The system displays a list of WebStudio users.























System Settings		User Management			
 Users  Groups & Privileges	Username	First Name	Last Name	Groups	
	user			Viewers	 
	u0			Testers	 
	u1			Developers Analysts	 
	u2			Viewers	 
	u3			Viewers	 
	u4			Deployers	 
	a1			Administrators	 
	snm	N	S	Administrators	 
	MyTest	My	Test	Developers	 
	SuperUser	Tom	Sawyer	Super Group	 
Add New User					

Figure 109: List of OpenL Tablets WebStudio Users

From the **Users** tab you can create, modify or delete user from the system.

To create a new User:

1. Click the **Add New User** link. The system displays the **Add New User** form.

Add New User

Username * SuperUser

Password * ••••••

First Name Tom

Last Name Sawyer

Groups *

- ☐ Viewers
- ☐ Developers
- ☐ Deployers
- ☐ Testers
- ☐ Analysts
- ☐ Administrators
- ☒ Super Group


Save Cancel

Figure 110: Creating a new user

2. Specify the user's login in the **Username** field and enter the password in the **Password** field.
3. Enter the user's first and last name (optional).
4. Select one or more groups to assign the user to.
5. Click **Save** to complete.


The system displays the new user in the **Users** list.

To modify a User:

1. Locate the user you wish to modify in the **Users** list and click the **Edit** icon: .
2. In the **Edit User** form, you can change the user First Name, Last Name and select or deselect the groups to include the user in.
3. Click **Save** to save your changes.

To delete a user:

Warning: You must have at least one Administrator (a member of the **Administrators** group) in the WebStudio, i.e. you cannot delete the only WebStudio Administrator.

1. Locate the user you want to delete in the **Users** list and click the **Delete** icon: .
2. Click **OK** in the confirmation dialog.

Index

A

advanced functionality, 41

B

benchmarking, 57

D

deploy configuration
 creating, 76
 defining descriptors, 76
 deploying, 77
deployed project
 opening, 77
Design Repository
 browsing, 59

F

file
 deleting, 73
 uploading, 73
folder
 creating, 72
 deleting, 73

G

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

M

module
 opening, 20

O

OpenL Tablets WebStudio
 components, 8
 definition, 7
 logging in, 10
 security, 9
 user interface, 11
 working with projects, 7
OpenL Tablets WebStudio, 7

P

project
 closing, 67
 compare versions, 21
 copying, 74
 creating, 61, 64
 deleting, 75
 deploying, 76
 deployment, 76
 editing, 20, 67
 erasing, 75
 exporting, 80
 managing, 20
 modifying, 69
 modifying contents, 72
 modifying properties, 69
 opening, 66
 redeploying, 78
 removing, 75
 revert changes, 20
 saving, 20, 67
 unlocking, 80
project dependency
 defining, 68
project tree
 filtering, 60
project versions
 comparing, 79

R

repository editor, 17
 using, 59
rules editor, 11
 managing projects, 20
 opening a module, 20
 overview, 12
 using, 19
 view modes, 13

S

search
 advanced, 32
 modes, 31
 performing, 31
 simple, 31
security, 9

T

tables
 modifying, 24
 viewing, 23
tracing, 54

U

unit tests, 48

V

view modes, 13