exigen®

OpenL Tablets WebStudio User's Guide

OpenL Tablets 5.9.1

OpenL Tablets BRMS

Document number: TP_EPB_7.1.0_OpenLWSt_User_3.0_SN
Revised: 04-06-2012

Table of Contents

Preface

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

The following topics are included in this preface:

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	• Manage technical details of rule tables. • Organize and deploy rule projects.	Knowledge of OpenL Tablets technology is required.

Related Information

OpenL Tablets Web Studio 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	• 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.
`Courier`	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**
Italic	• 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 Web Studio

This section introduces the main OpenL Tablets Web Studio concepts.

The following topics are included in this section:

What Is OpenL Tablets Web Studio?

OpenL Tablets Web Studio (further also referred as **Web Studio**) 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.*

Users can modify rules directly in a web browser without installing additional tools by using OpenL Tablets Web Studio. OpenL Tablets Web Studio 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 Web Studio

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

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

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

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

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

OpenL Tablets Web Studio Components

OpenL Tablets Web Studio consists of the following main components:

OpenL Tablets Web Studio components	
Component	**Description**
Rule editor	Graphic user interface running in a web browser allowing users to browse rule modules, modify table data, and run tests. Rule editor is the default user interface displayed when user opens OpenL Tablets Web Studio. Rule editor does not display all rule module files but provides a logical view of rules stored in a module. This view is convenient for users who modify business rules. Rule editor displays only modules available in projects stored in user's workspace. To retrieve a project to user's workspace, the project must be opened or checked out. For information on opening and checking out projects, see Working with Projects in OpenL Tablets Web Studio. For detailed information on using rule editor, see Chapter 3: Using Rule Editor.
Repository editor	Graphic user interface running in a web browser allowing users to browse and manage projects in rules repository. Unlike rule editor, repository editor displays physical contents of rule projects. Users can easily switch between rule editor and repository editor in user interface. Repository editor provides the following main functions: • uploading projects from the file system to rules repository • checking out, checking in, opening, and closing projects • modifying project structure and properties • managing project versions and dependencies • copying and deleting projects in rules repository • managing and tracing project deployments For detailed information on using repository editor, see Chapter 4: Using Repository Editor.

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

Security Overview

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

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

Chapter 2: Getting Started

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

The following topics are included in this section:

Logging In to OpenL Tablets Web Studio

To log in to OpenL Tablets Web Studio, proceed as follows:

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

   The OpenL Tablets Web Studio URL has the following pattern:

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

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

Figure 1: Login window

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

Understanding the User Interface

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

Rule Editor

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

The following topics are included in this section:

Rule Editor Overview

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

Rule editor resembles the following:

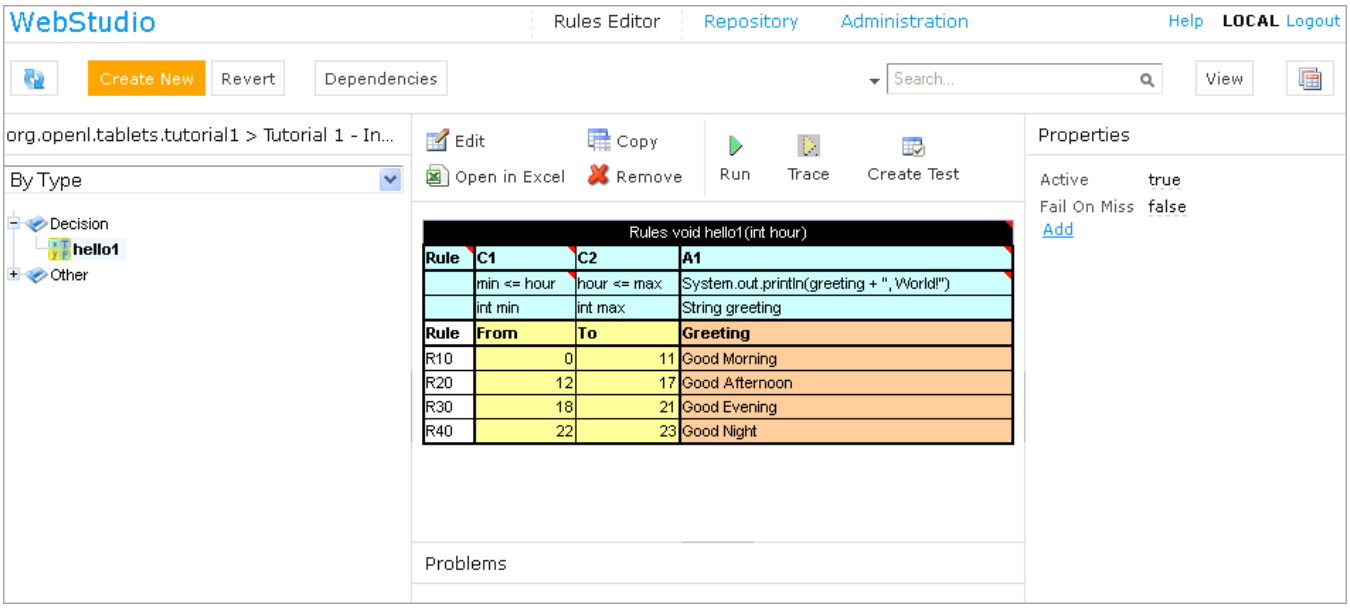


Figure 2: OpenL Tablets Web Studio rule editor

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

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

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

The right pane displays properties of the currently displayed table.

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

Rule editor toolbar buttons	
Button	**Description**
View	Switches to view with table headers and/or Excel formulas. For information on view modes, see View Modes.
Revert	Opens a page for reverting project changes
	Opens the search window. For information on performing searches, see Performing a Search.

Rule editor toolbar buttons	
Button	**Description**
	Refreshes OpenL Tablets Web Studio with latest changes in Excel files.
	Initiates the table creation wizard.
	Initiates a dialog for comparing Excel files.
	Opens Dependency graph
	Switches user interface to repository editor. For general information on repository editor, see Repository Editor.
	Opens OpenL Tablets Web Studio help.
	Logs the user out of OpenL Tablets Web Studio.
	Switches user interface to rules editor For general information on rules editor, see Rule Editor
	Switches user interface to the Administration mode

View Modes

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

Project display modes in rule editor	
Mode	**Description**
Simple view	Project view is more business oriented displaying only those project elements relevant to a business user. Structure of the tree is logical rather than physical. Rule tables are organized into categories based on Excel table sheets and the **category** table property. An example of a module tree displayed in simple view sorted by the **Category** parameter is as follows:

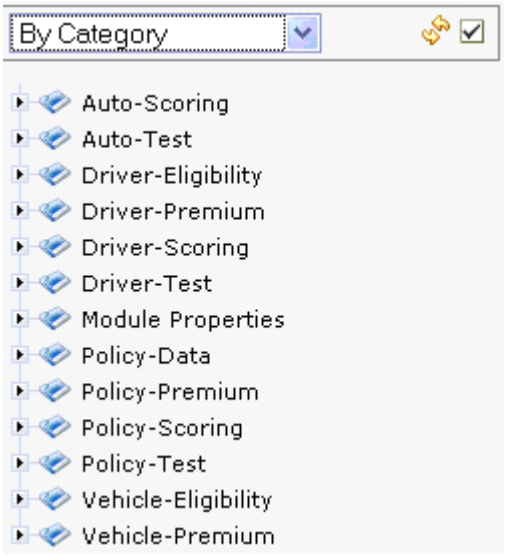


Figure 3: Module tree in simple view sorted by a category

An example of the module tree sorted by **Category Detailed** is as follows:

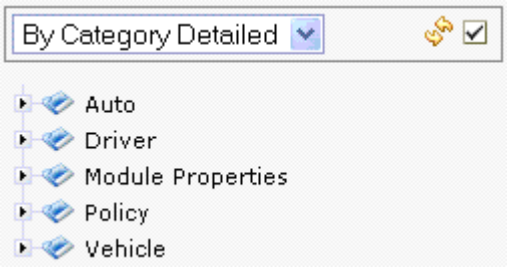


Figure 4:Module tree in simple view sorted by Category Detailed

An example of the module tree sorted by **Category Inversed** is as follows:

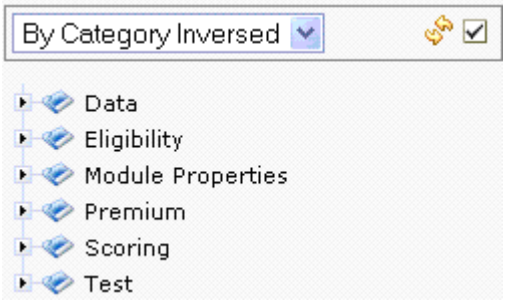


Figure 5:Module tree in simple view sorted by Category Inversed

Project display modes in rule editor	
Mode	**Description**
	OpenL Tablets Web Studio 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:

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

Figure 6: Rule table in simple view

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

Extended view

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

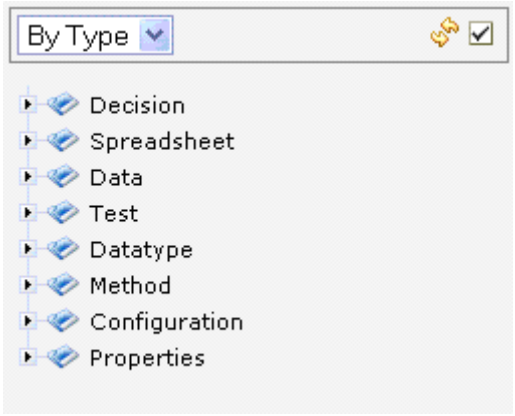


Figure 7: Module tree in extended view sorted by type

The same module tree displayed with sorting by file is as follows:

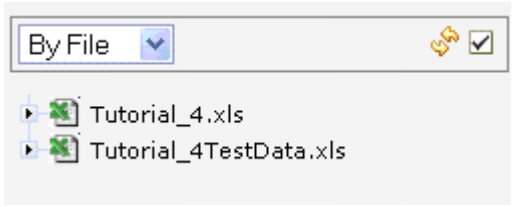


Figure 8: Module tree in extended view sorted by file

OpenL Tablets Web Studio 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 extended view:

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

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

Figure 9: Rule table in extended view

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

Repository Editor

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

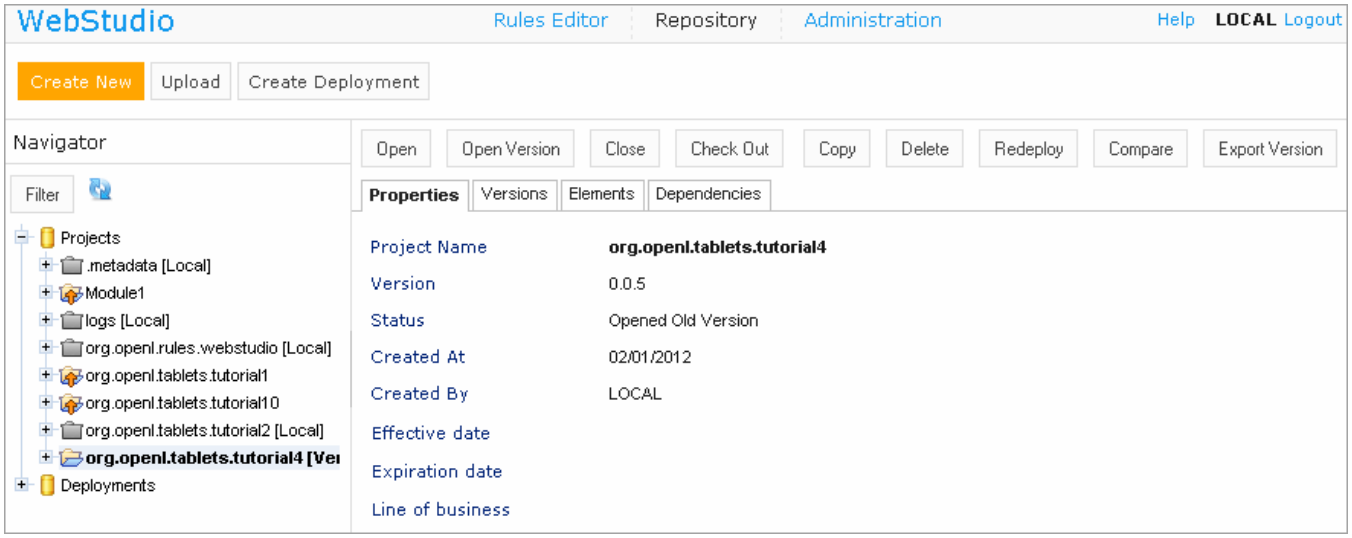


Figure 10: OpenL Tablets Web Studio repository editor

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

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

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

For detailed information on tasks that can be performed in repository editor, see Chapter 4: Using Repository Editor.

Chapter 3: Using Rule Editor

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

The following topics are included in this section:

Filtering Projects

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

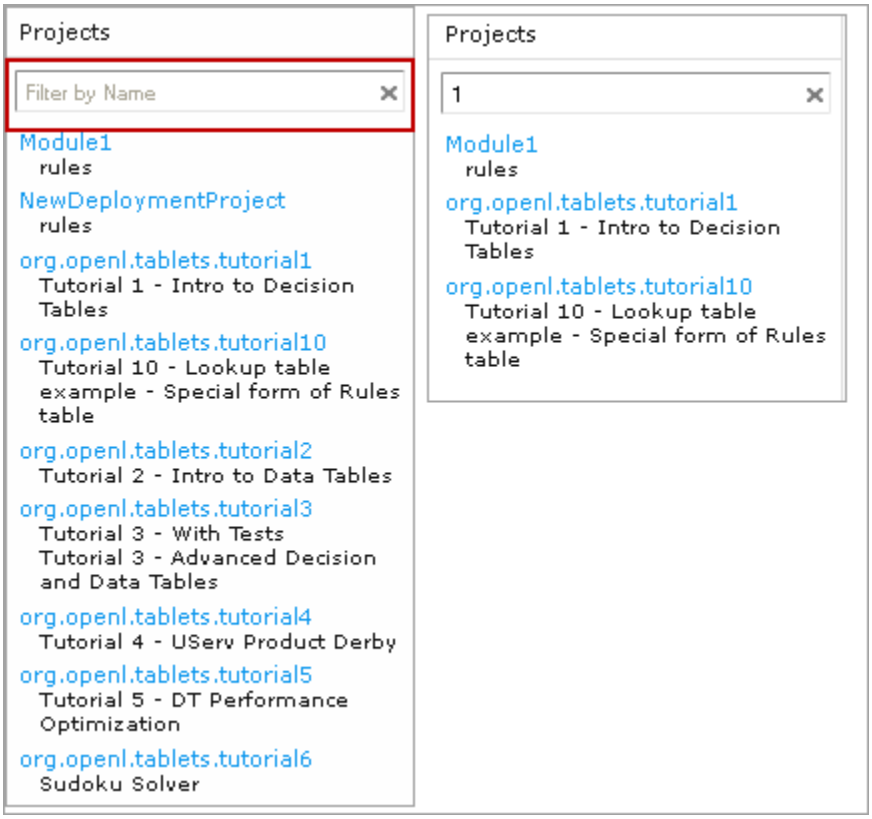


Figure 11: Filtering projects by Name

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

Opening a Module

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

Managing Projects

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

- Checking Out and Checking In a Project
- Uploading Projects to Design Time Repository
- Reverting Project Changes

Checking Out and Checking In a Project

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

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

Reverting Project Changes

OpenL Tablets Web Studio provides functionality allowing users to revert project changes to a specific date. To perform reverting, click **Revert Project Changes** in the project toolbar, select the date of modify the project and click Revert.

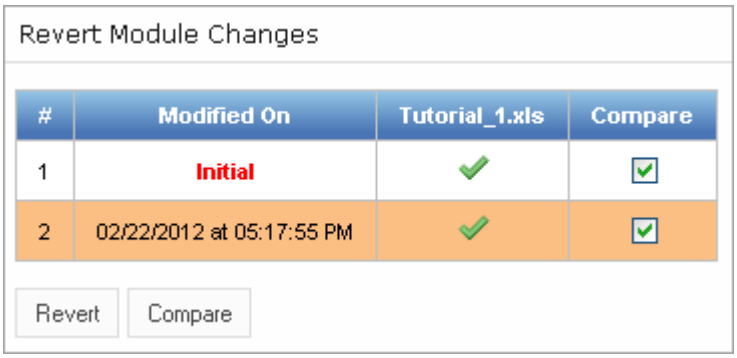


Figure 12: Revert project changes

Viewing Tables

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

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

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

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

Modifying Tables

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

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

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

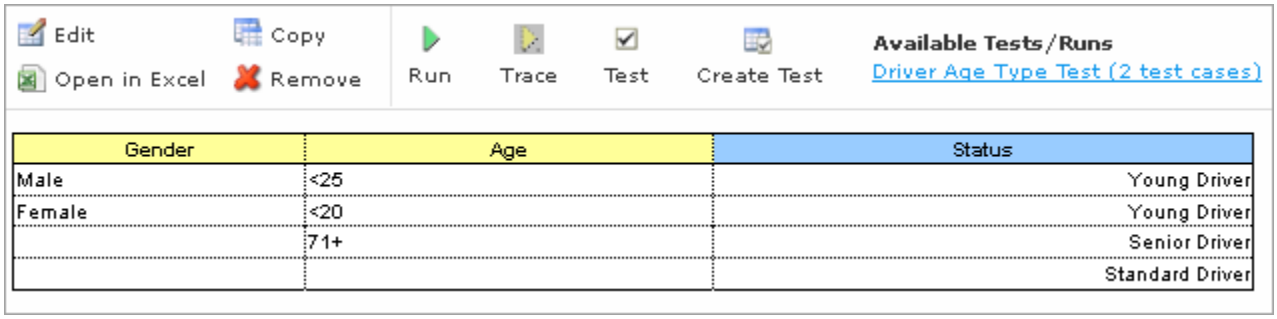


Figure 13: Table opened in OpenL Tablets Web Studio

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

- Above the table, click the **Edit Table** button.
- Right-click anywhere on the table and click **Edit.**

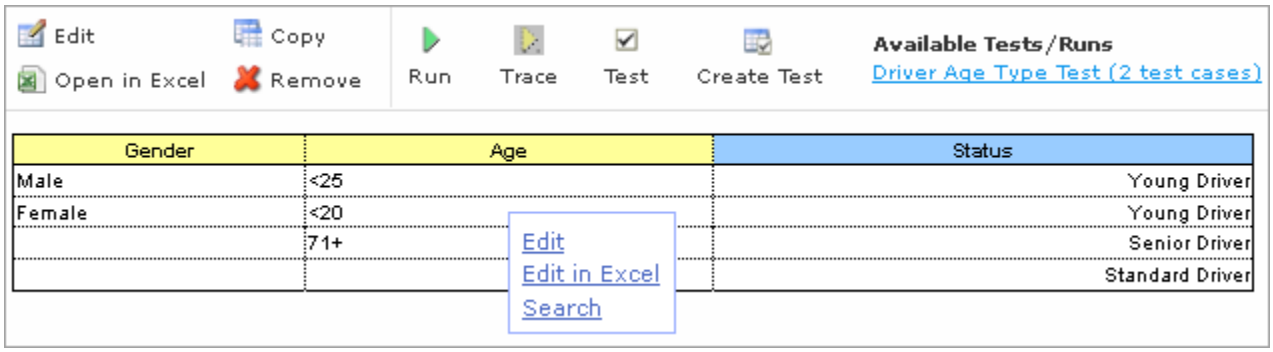


Figure 14: Switching to edit mode

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

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

The table is switched to edit mode.

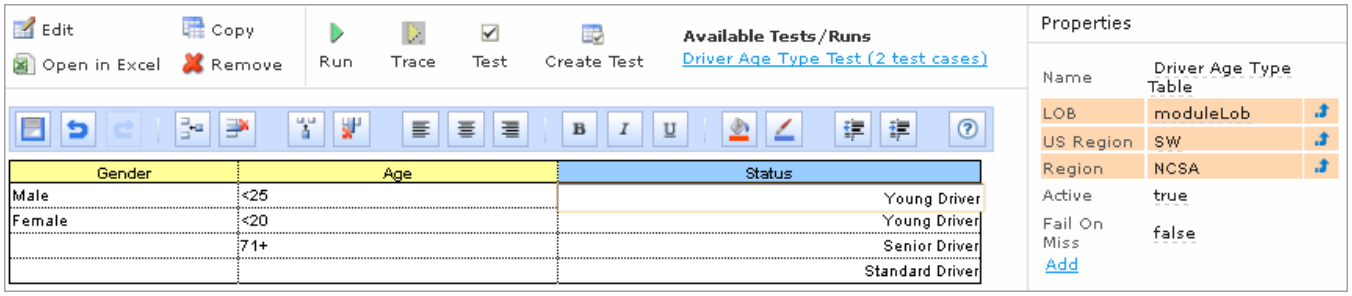


Figure 15: Table in edit mode

The edit mode provides the following buttons above the table:

Table editing buttons	
Button	**Description**
	Saves changes in table.
	Reverses last changes.
	Reapplies reversed changes.
	Inserts a row.
	Deletes a row.
	Inserts a column.
	Deletes a column.
	Aligns text in currently selected cell with left edge.
	Centers text in currently selected cell.
	Aligns text in currently selected cell with right edge.
	Make the text **bold**
	*Italicisize* the cell text
	Underline the cell text
	Setl fill color
	Set 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 cell is selected.

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

3.

4.

Performing a Search

OpenL Tablets Web Studio provides search functionality allowing users to perform a search across data in all module tables.

There are the following search modes in the Web Studio:

- Simple Search
- Business Search
- Advanced Search

Simple Search

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

To perform a simple search, do one of the followings:

1. In the search field, enter the word or phrase and click **.**
2. If a table is displayed, right-click on the cell containing a value you want to search and click **Search.**

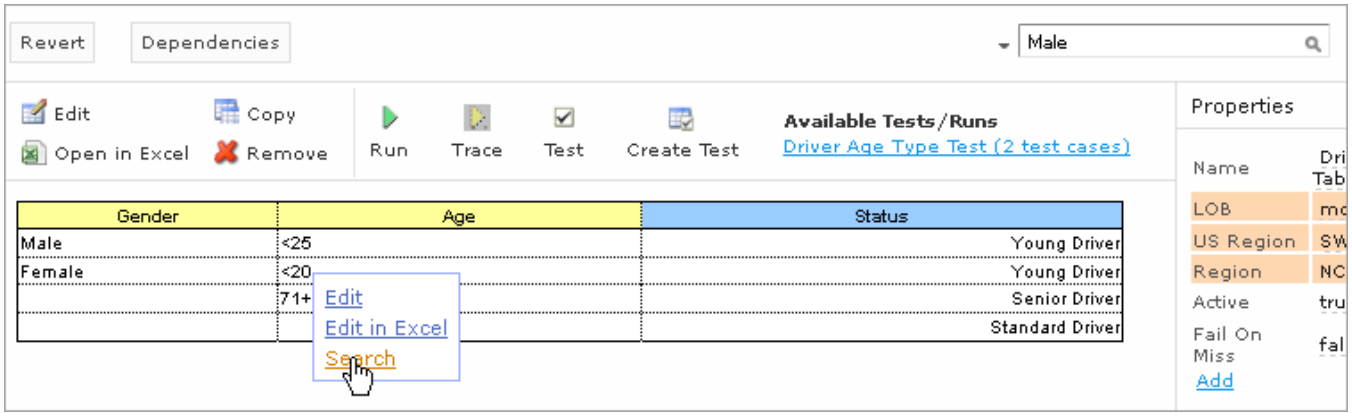


Figure 16: Starting the simple search

OpenL Tablets Web Studio displays search results in form of links to Excel files containing the entered text.

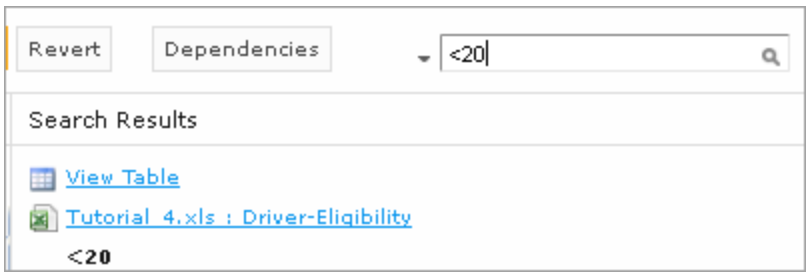


Figure 17: Search results

Business Search

Business search by particular table properties.

5. To switch to the business search mode, click an arrow to the left of search window and select **Business Search.**

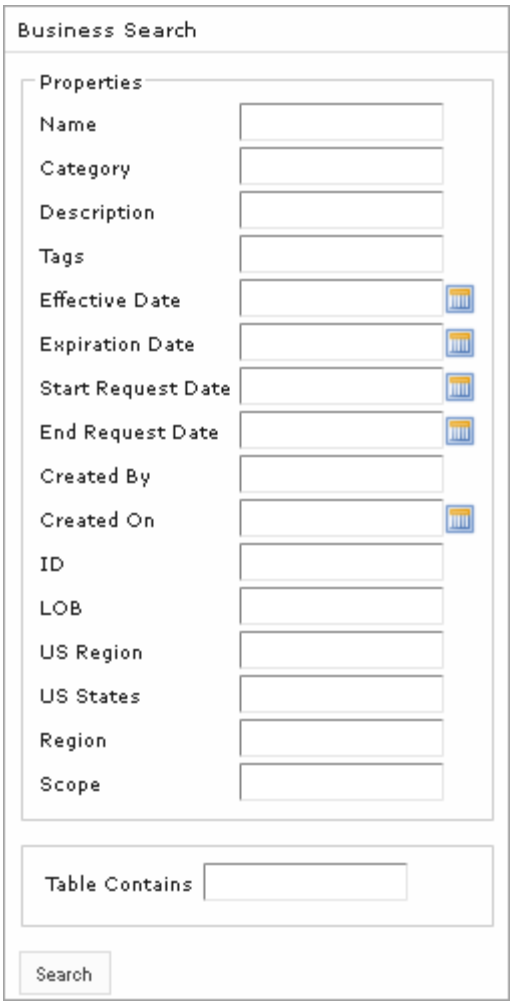


Figure 18: Business search

OpenL Tablets Web Studio displays search results in form of links to Excel files containing the entered text.

Note: search by using “Tables Contains” field is possible only in case at least one Business property has already been defined

Advanced Search

Advanced search allows the user to narrow the search by specifying criteria for tables where the search is to be performed. In addition, advanced search provides controls for saving certain search criteria for future use.

1. To switch to the advanced search mode, click an arrow to the left of search window and select **Advanced Search.**

   The advanced search window appears containing various controls for specifying search criteria.

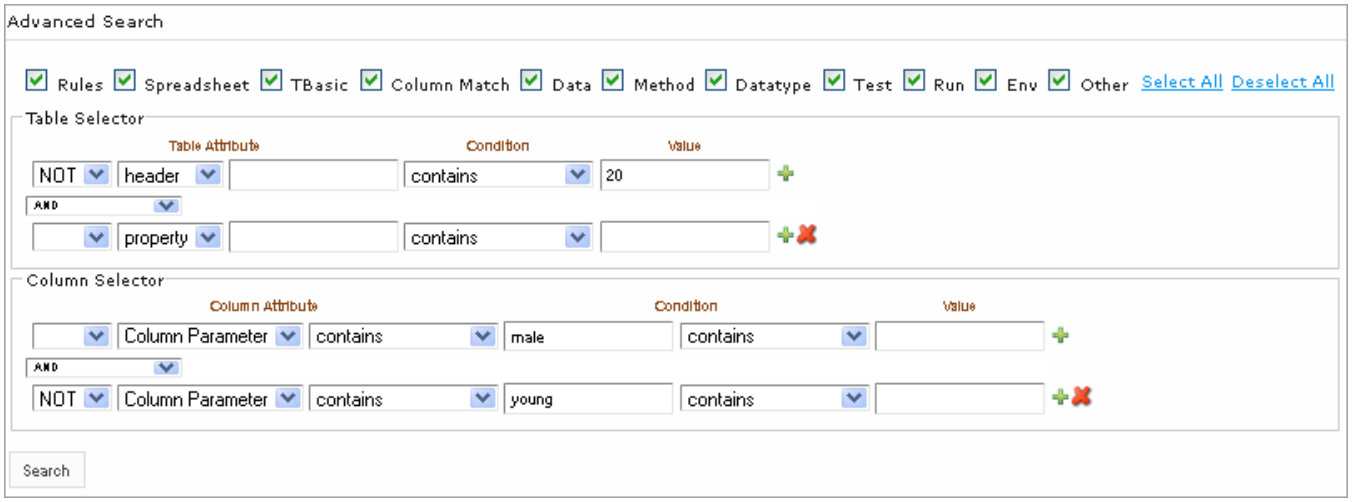


Figure 19: Advanced search

2. In the **Table Selector** section, enter or modify search criteria for tables in which the search must be performed.
3. In the **Column Selector** section, enter or modify search criteria for columns in which the search must be performed.
4. Click **Search.**

OpenL Tablets Web Studio displays tables matching the search criteria below the advanced search window.

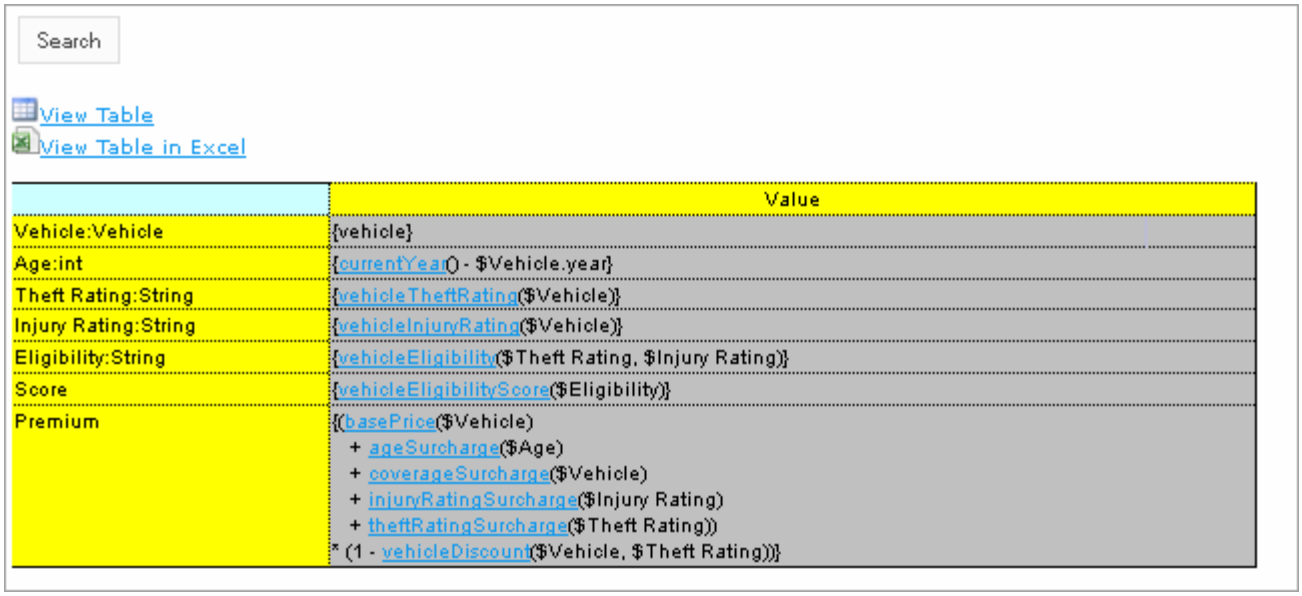


Figure 22: Advanced search results

Creating New Table

OpenL Tablets Web Studio allows creating new tables of the following types:

- Datatype
- Datatype Alias
- Test Method
- Properties

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

.

Datatype table wizard

To create new Datatype table, follow these steps:

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

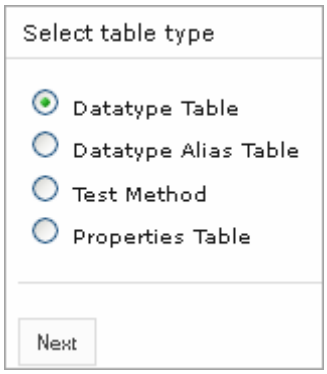


Figure 20: Table wizards

2. Input Business Name and Category

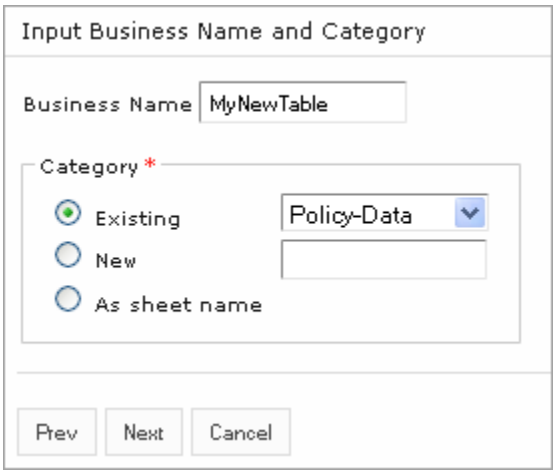


Figure 21: Datatype table wizard – Step 1

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

Figure 25: Datatype table wizard – Step 2

4. Define type fields.

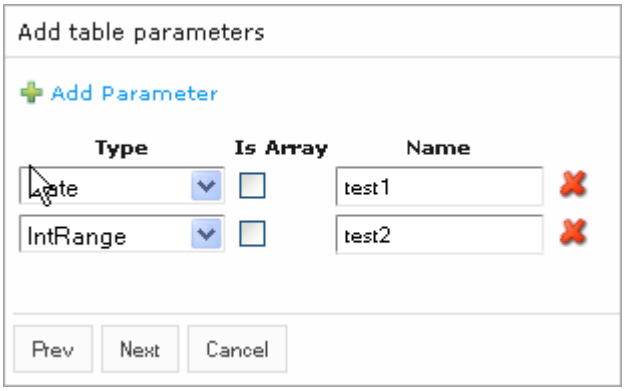


Figure 26: Datatype table wizard – Step 3

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

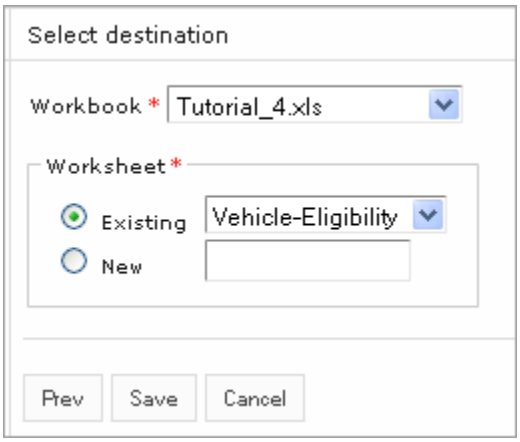


Figure 27: Datatype table wizard – Step 4

Chapter 4: Editing and Testing Functionality

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

The following topics are included in this section:

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.

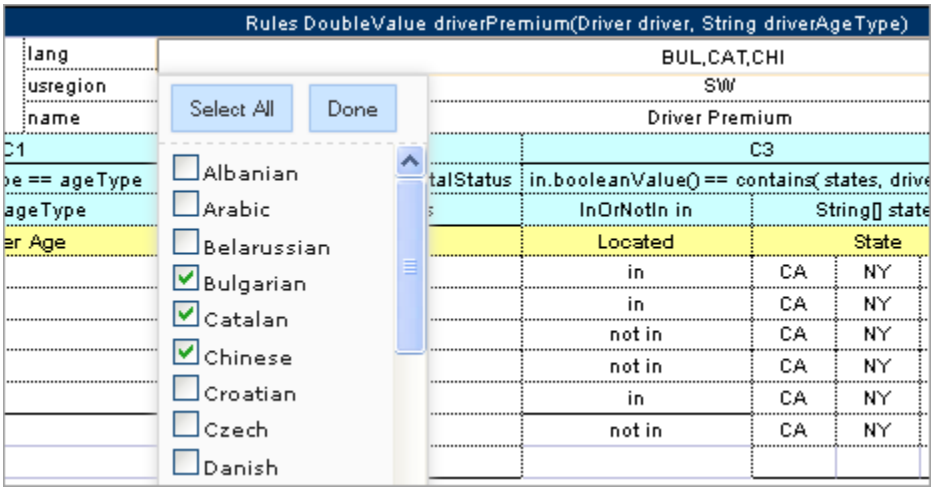


Figure 228: Multi editor in Web Studio for editing comma separated arrays

Default and Inherited Properties

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

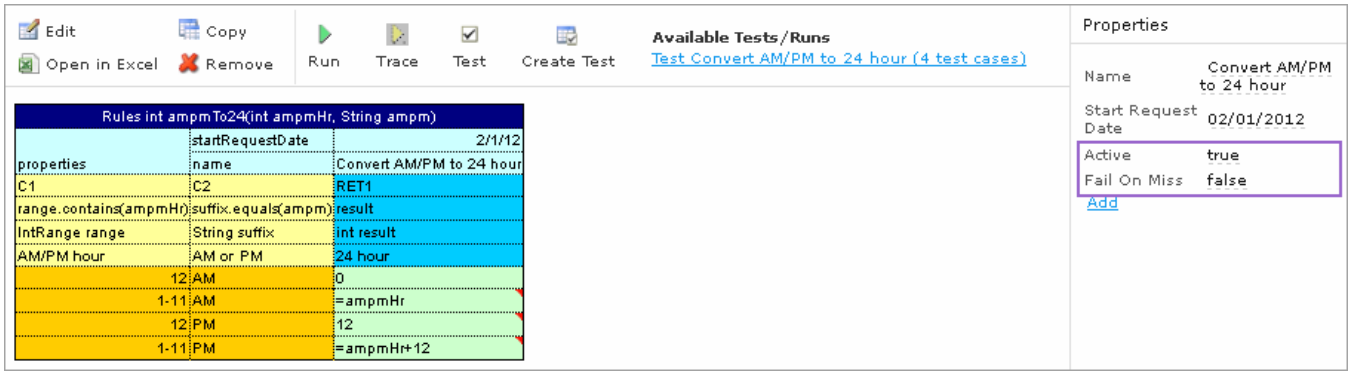


Figure 239: Default table properties example

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

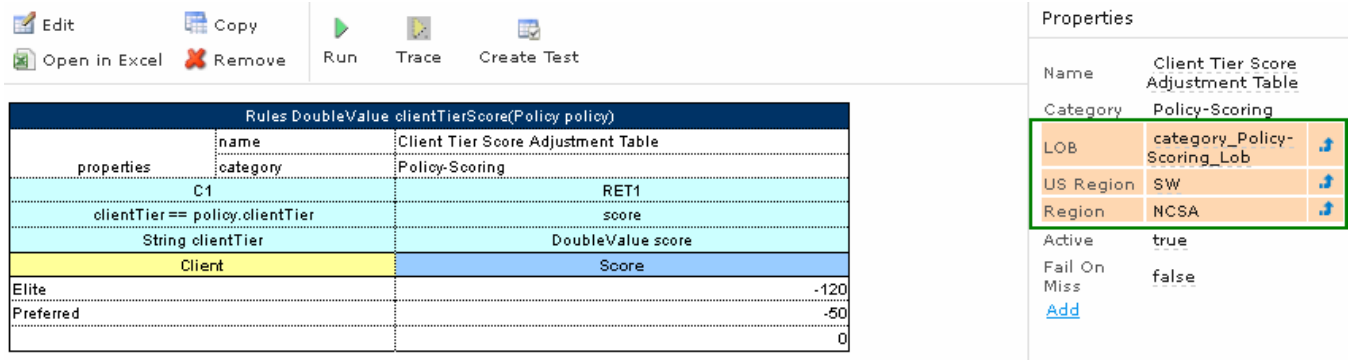


Figure 30: A category level properties example

System Properties

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

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

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

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

Figure 31: A system properties example

Properties for Particular Table Type

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

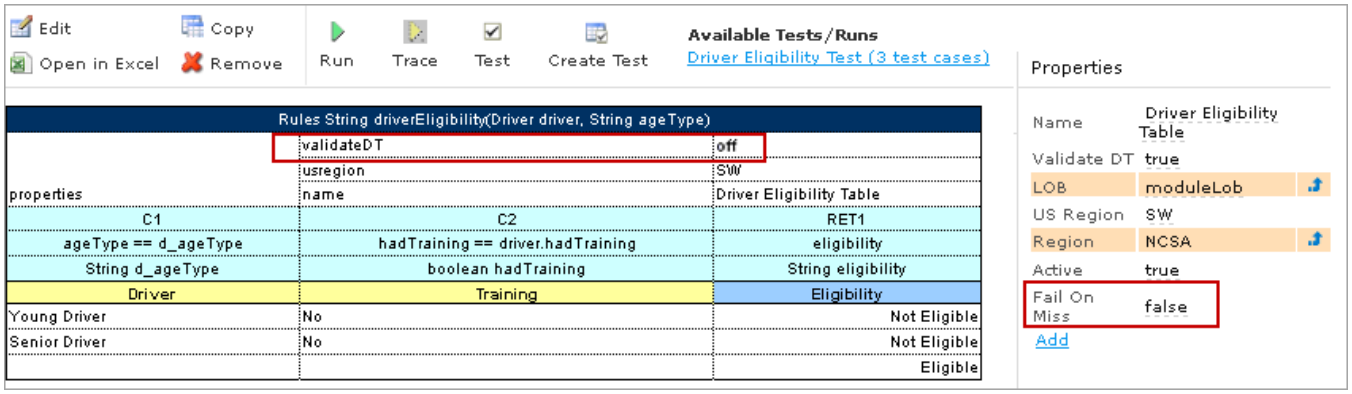


Figure 32: Properties for Decision table type

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

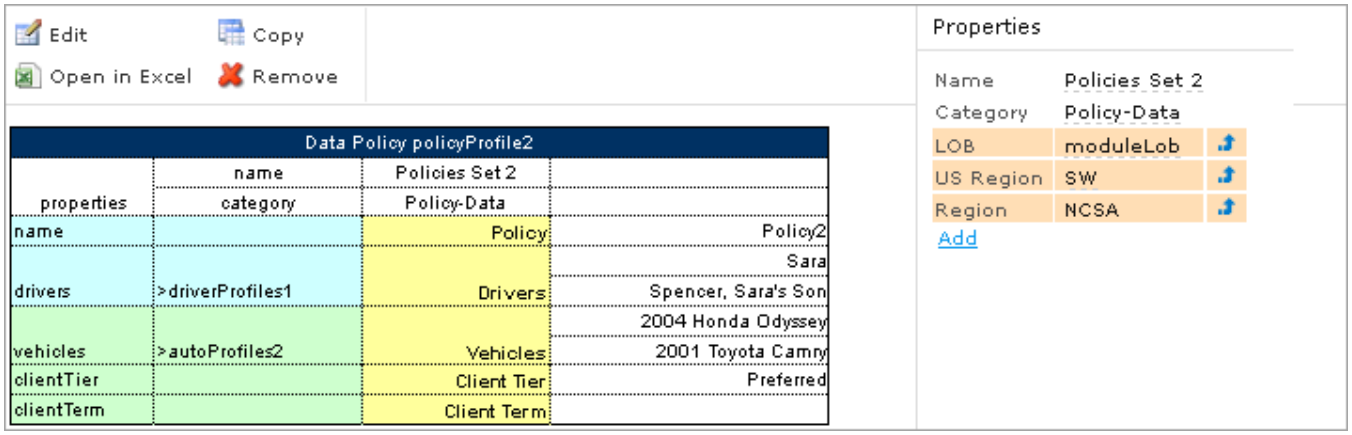


Figure 243: There are no properties on Data table that are defined for Decision table only

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

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

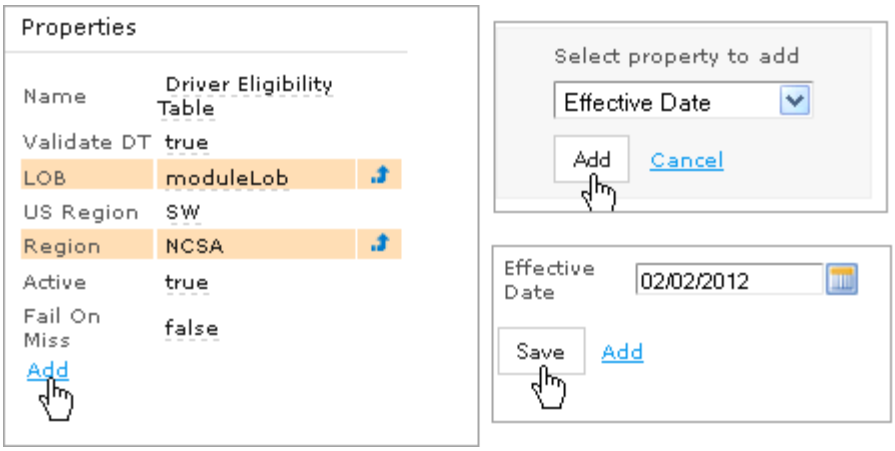


Figure 34: Adding a new property for the current table

Copy as New Version

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

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

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

Figure 255: 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.

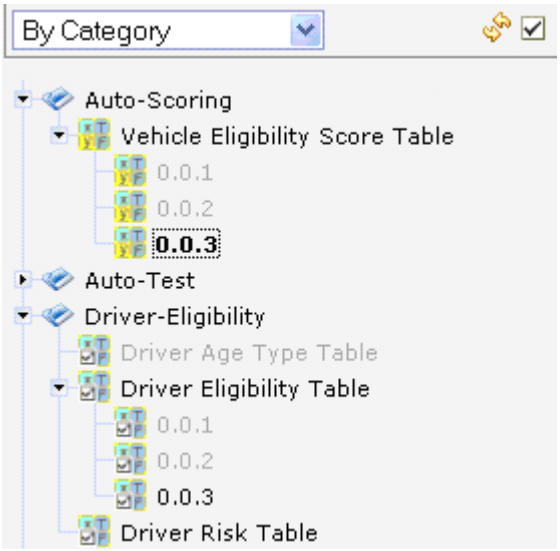


Figure 266: Displaying table versions in the module tree

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

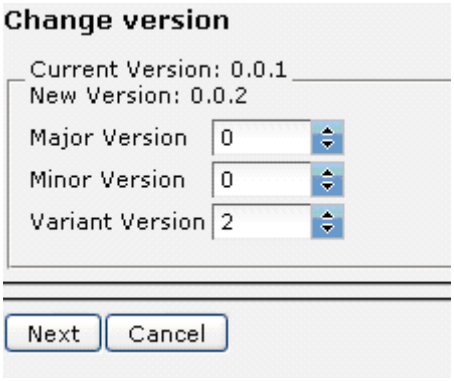


Figure 277: Entering a new version number

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)		
C1	C2	RET1
range.contains	suffix.equals	result
IntRange range	String suffix	int result
AM/PM hour	AM or PM	24 hour
12	AM	0
1-11	AM	=ampmHr
12	PM	12
1-11	PM	=ampmHr+12

Testmethod ampmTo24 ampmTo24Test		
ampmHr	ampm	_res_
Hour	AM/PM	24 Hr
3	AM	3
12	AM	0
12	PM	12
3	PM	15

Figure 288: Decision table and its unit test table

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

Navigation

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

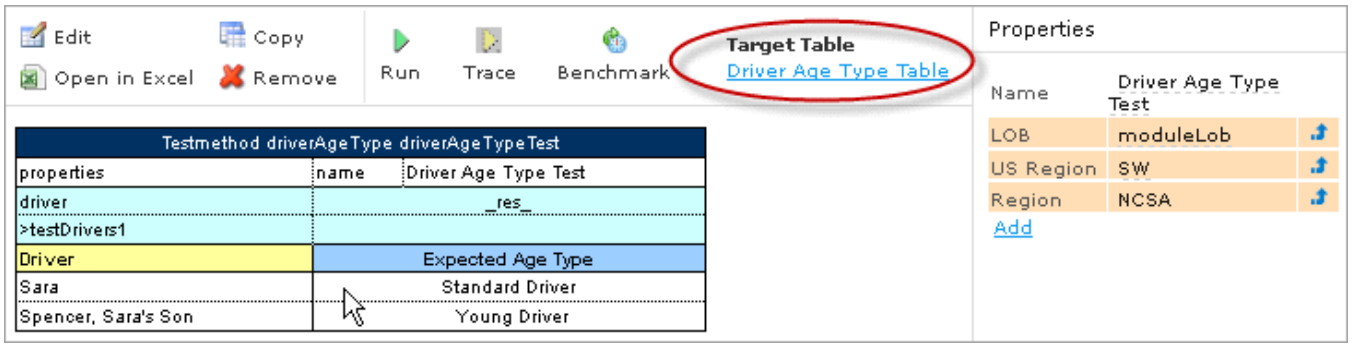


Figure 39: Navigation link to target table

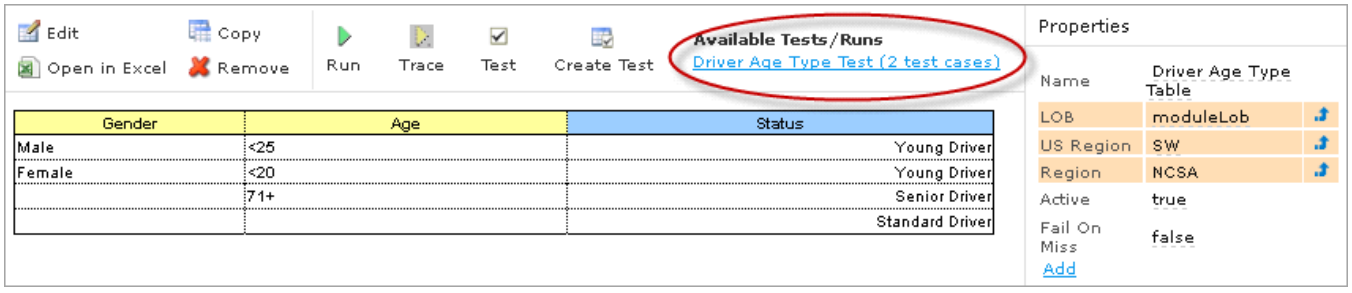
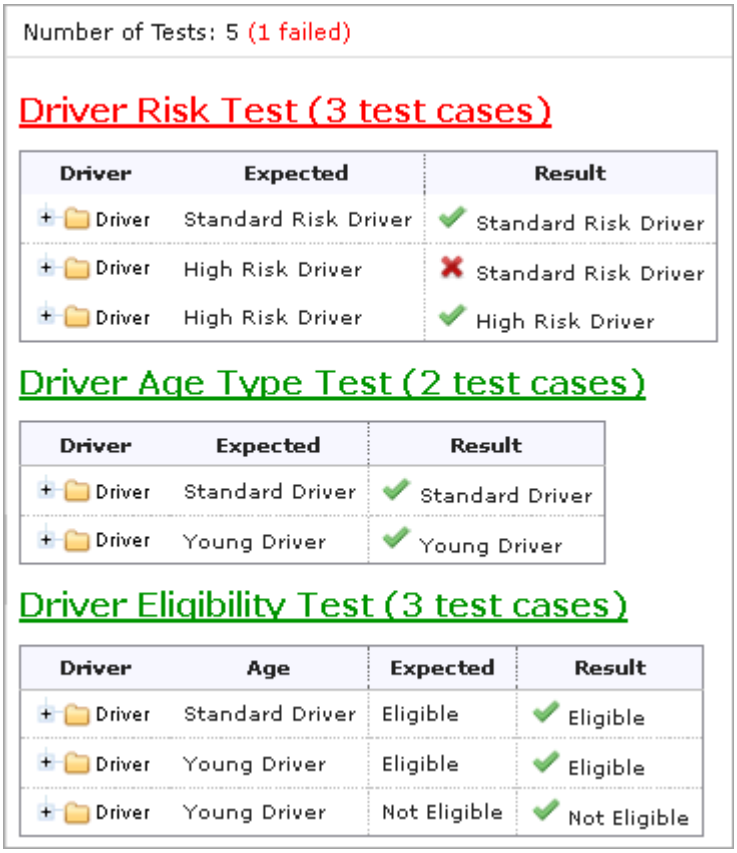
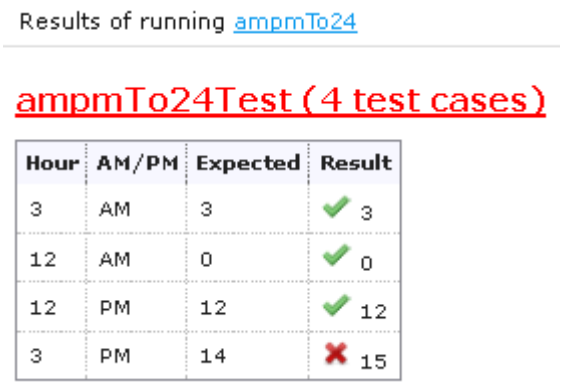


Figure 40: Navigation links to available checks and tests

Run Tests

The following methods can be used to run unit tests:

Methods for running unit tests	
Method	**Description**
Execute all project tests at once	System automatically executes all test runs in every unit test in project and displays a summary of results. To run all project tests, click the Run **All Tests** icon above the module tree in rule editor**.** Test results resemble the following:  *Figure 41: Results of running all project tests* Failed tests are represented by the mark. Passed tests are represented by the mark.

Methods for running unit tests	
Method	**Description**
Execute all tests for a single decision table	System executes all test runs for one particular decision table. To execute all test runs for one particular decision table, in rule editor, in the module tree, select the decision table and, in the upper part of the middle pane, click **Test** ☑**.** Test results resemble the following:  Failed tests are represented by the ✖ mark. Passed tests are represented by the ☑ mark.

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	14	✖ 15

Figure 42: Results of executing all test runs for one decision table

Creating New Test

Web Studio provides convenient way to create new test table.

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

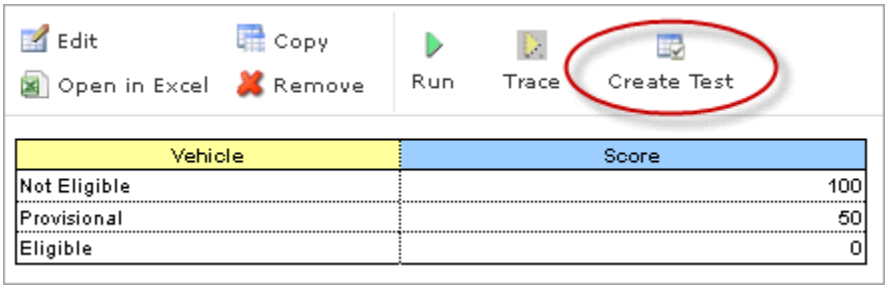


Figure 43: Create new test table

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

Tracing

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

- all unit tests
- decision tables and method tables with attached `Runmethod` data

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

To display the trace view, in rule editor, open the required table and, in the middle pane, click **Trace** .

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.

The trace view resembles the following:

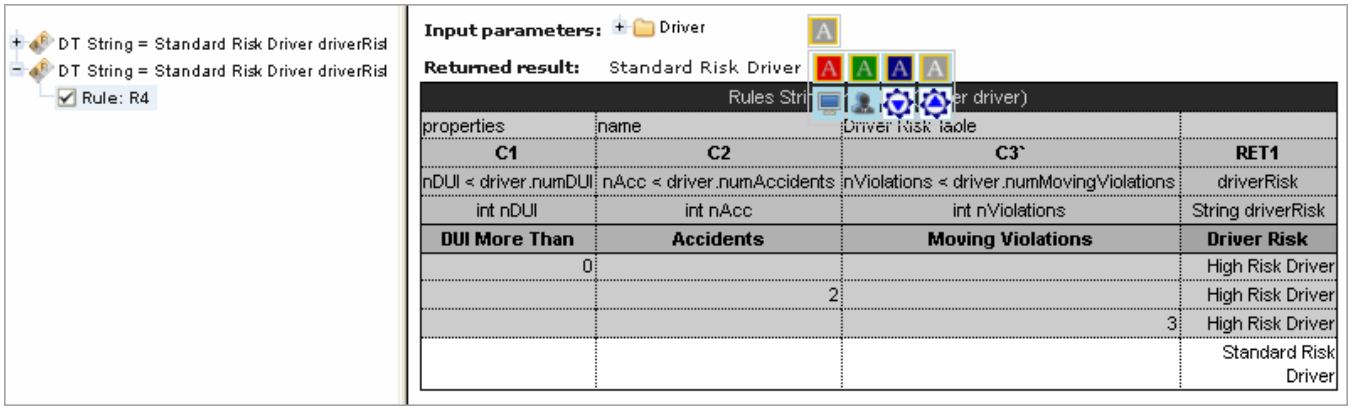


Figure 294: Tracing a rule

The left side displays a tree consisting of decision tables as tree nodes and fired rule rows as tree leaves. In addition, the view displays the actual parameters used in the particular method call.

If an element in the tree is selected, the corresponding decision 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 using controls above the decision table.

Benchmarking

OpenL Tablets Web Studio 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 containing appropriate elements that can be run and also next to every appropriate method.

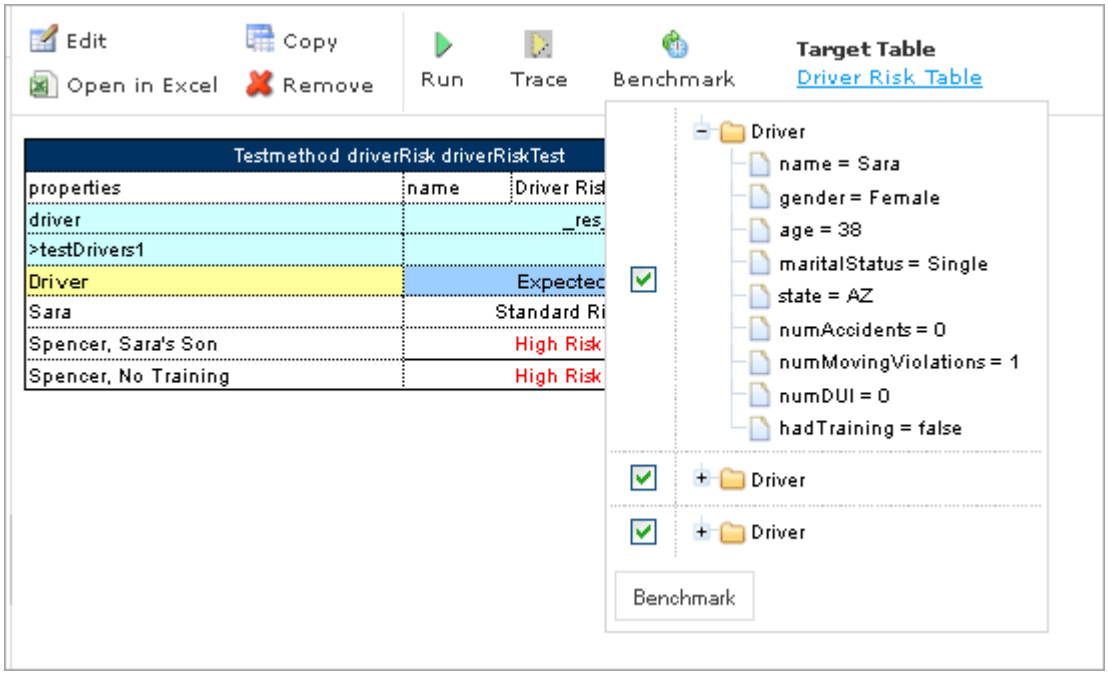


Figure 305: Controls for measuring performance

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

Results of benchmarking Test Driver Premium

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

Compare Delete

Figure 316: Benchmarking results

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

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

Comparison results are displayed below the benchmarking table.

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

Figure 327: Comparing benchmarking results

Chapter 5: Using Repository Editor

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

The following topics are included in this section:

Browsing Design Time Repository

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

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

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

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

Project icons in repository editor	
Icon	**Description**
	Project is closed by current user but checked out by another user. Current user cannot check out the project.
	Project is opened by current user but checked out by another user. Current user cannot check out the project.
	Project exists only in user's workspace but not in design time repository. Other users do not see this project. User can delete the project or upload it to design time repository as described in Uploading Projects to Design Time Repository.
	Project is marked for deletion. In OpenL Tablets Web Studio, deletion of a project takes place in the following phases:

Phase	Description
Deleting a project	Project is removed from user's workspace and marked for deletion. In this phase, the project can be restored using the undelete function. For information on deleting a project, see Deleting a Project.
Erasing a project	Deleted project is permanently removed from design time repository. After this phase, the project cannot be restored. For information on erasing a project, see Erasing a Project.

Filtering the Project Tree

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

To filter the project tree, proceed as follows:

1. Above the project tree, click **Filter.**
2. In the pop up window, enter a list of file extensions, separated by semicolon as follows:

xls;properties;txt

3. Click **Apply.**

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

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

Uploading a Project

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

To upload an archived rule project to design time repository:

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

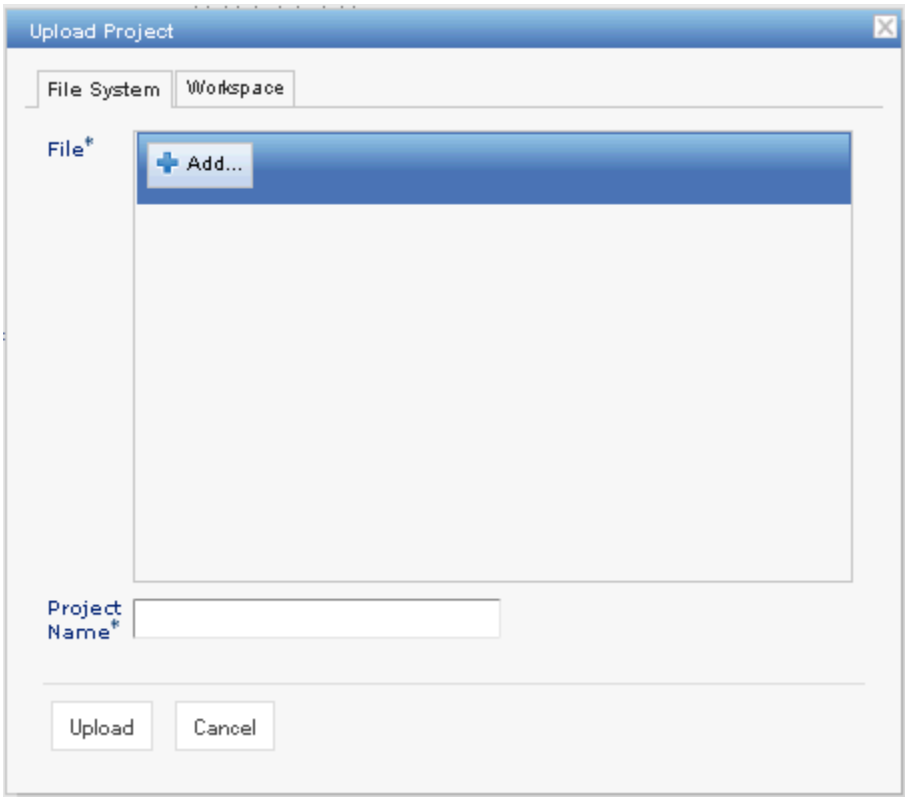


Figure 338: Uploading a project

2. In the **File** field, select the ZIP file containing the rule project or excel file containing OpenL tables.
3. In the **Project Name** field, enter name by which the project must be represented in design time repository. You can also open the **Workspace** tab and select checkboxes for projects to be uploaded.
4. Click **Upload.**

Creating a Project

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

To create a new project, proceed as follows:

1. In the project tree, select **Projects.**
2. In the top line menu, click **Create New.**

   The **New Project** window appears.

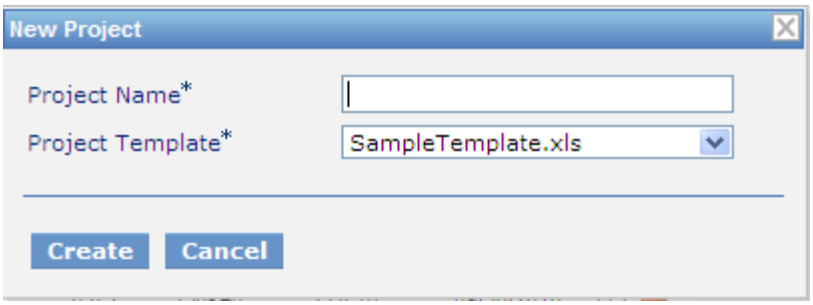


Figure 349: Creating a new project

3. In the **Project Name** field, enter the project name and click **Create.**

   New project is created in design time repository. Initially, project structure corresponds to selected project template but can be constructed manually.
4. To construct the project structure, add folders and upload files as described in Modifying Project Contents.

Opening a Project

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

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

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

Closing a Project

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

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

Checking Out a Project

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

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

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

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

Checking In a Project

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

To check in a project, proceed as follows:

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

   The **Check In** window appears.

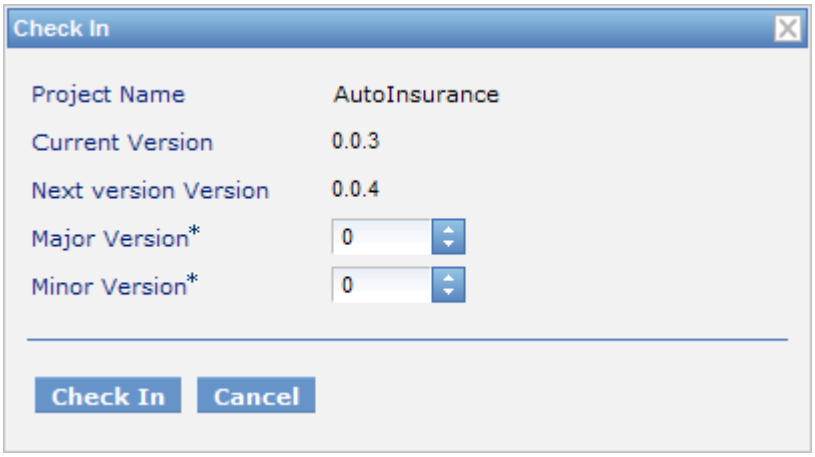


Figure 50: Checking in a project

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

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

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

Defining Project Dependencies

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

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

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

Figure 51: Defining dependencies

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

3. To define a new dependency, click **Add.**
   The **Add dependency** window appears.

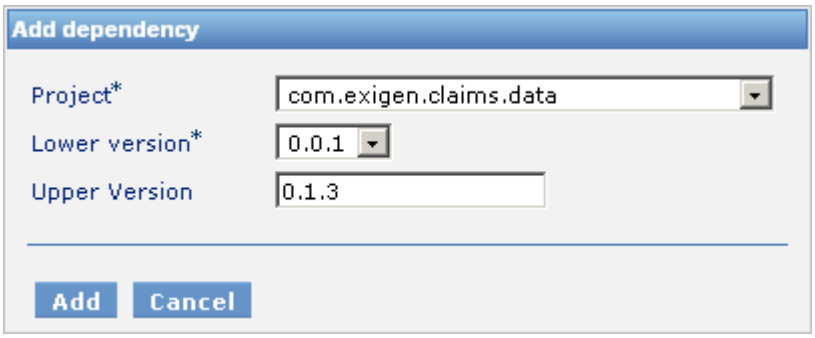


Figure 52: Defining a new dependency

4. In the **Project** list box, select the required project.
5. In the **Lower version** list box, select the oldest allowed version of the referenced project.
6. Optionally, in the **Upper Version** field, enter the latest allowed project version.

If the **Upper Version** field is empty, any project version above the one specified in the **Lower version** field is allowed.

7. Click **Add.**
8. Repeat this procedure to add as many dependencies as required.

Modifying a Project

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

The following topics are included in this section:

- Modifying Project Properties
- Modifying Project Contents

Modifying Project Properties

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

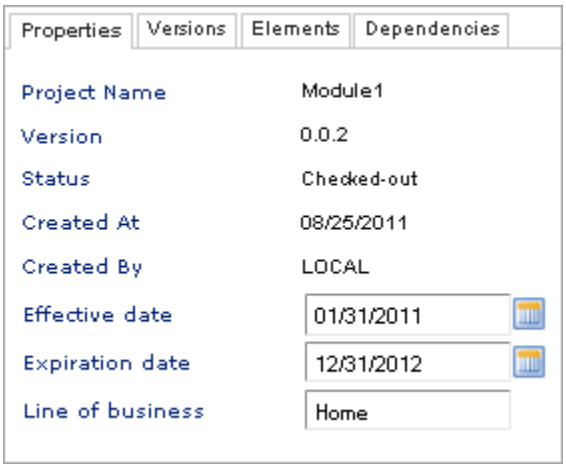


Figure 53: Project properties

By default, the following editable properties are displayed:

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

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

To add more properties to UI, proceed as follows:

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

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

   An example is as follows:
   ```
   props.attribute1 = Attribute_Label_Name 1
   props.attribute2 = Attribute_Label_Name 2
   ```

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

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

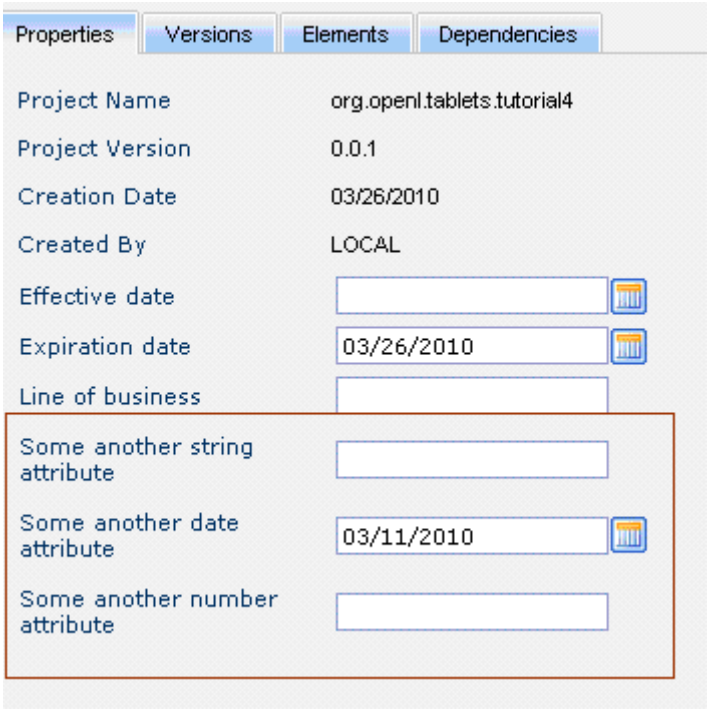


Figure 354: Project UI with properties 1, 6, and 10 added

Modifying Project Contents

This section describes modifying the physical structure of a project.

The following topics are included in this section:

- Creating a Folder
- Uploading a File
- Deleting a Folder or a File

Creating a Folder

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

1. If the project is not checked out, check it out as described in Checking Out a Project.
2. In the project tree, select the parent folder in which the new folder must be created.

   To create a root level folder, the project name must be selected in the project tree.
3. In the right pane, click **Add Folder.**
4. In the **Add Folder** window, enter the folder name and click **Add.**

Uploading a File

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

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

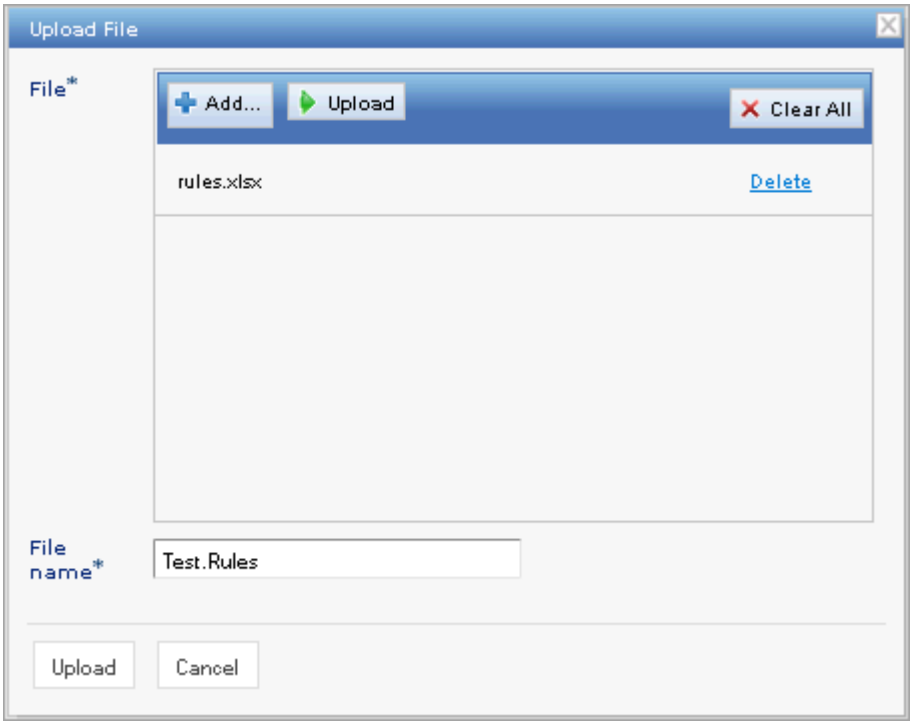


Figure 365: Uploading a file

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

Deleting a Folder or a File

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

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

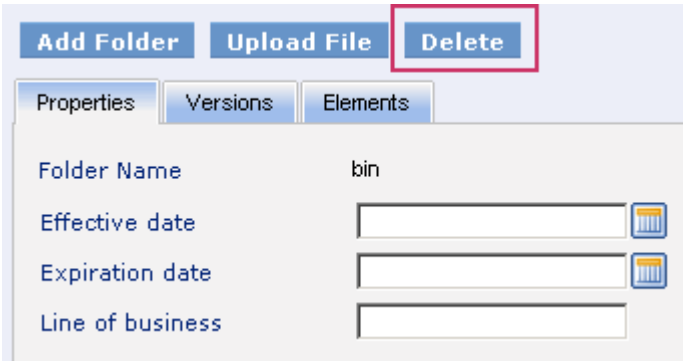


Figure 376: Deleting a project element

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

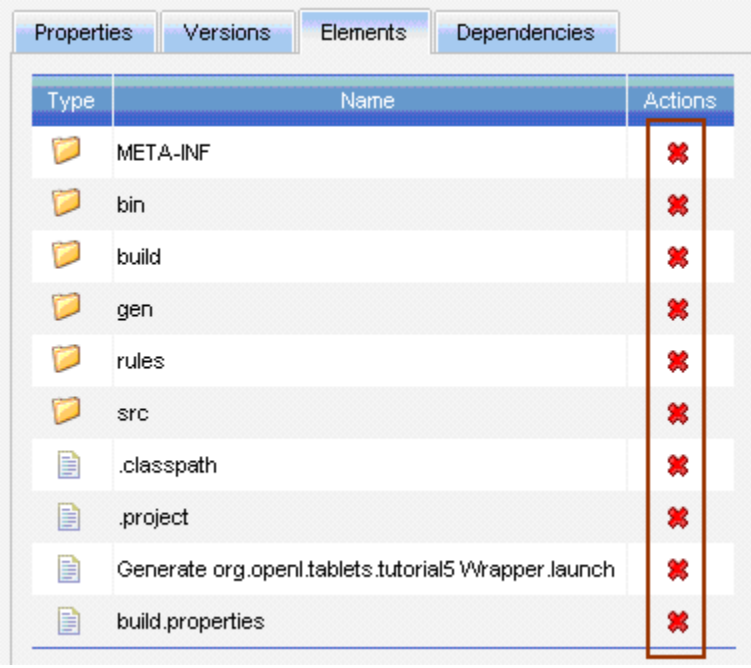


Figure 387: Deleting project elements in the Elements tab

A confirmation window appears.

3. In the confirmation window, click **Delete.**

Copying a Project

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

To copy a project, proceed as follows:

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

Removing a Project

Removing a project is executed in the following phases:

- Deleting a Project
- Erasing a Project

Deleting a Project

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

To delete a project, proceed as follows:

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

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

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

Erasing a Project

Erasing a project permanently removes it from design time repository.

Warning: Erased projects cannot be restored.

To erase a project, proceed as follows:

1. Delete the project as described in Deleting a Project.
2. In the project tree, select the project and, in the right pane, click **Erase.**
3. In the confirmation window, click **Erase.**

Deploying Projects

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

The following topics are included in this section:

Creating a Deployment Project

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

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

To create a deployment project, proceed as follows:

1. In the project tree, select the **Deployments** category.
2. Click **Create Deployment** in the top line menu**.**
3. In the **New Deployment** window, enter the deployment project name and click **Create.**

   The new deployment project appears in the project tree.
4. Define deployment project descriptors as described in Defining Deployment Project Descriptors.

Defining Deployment Project Descriptors

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

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

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

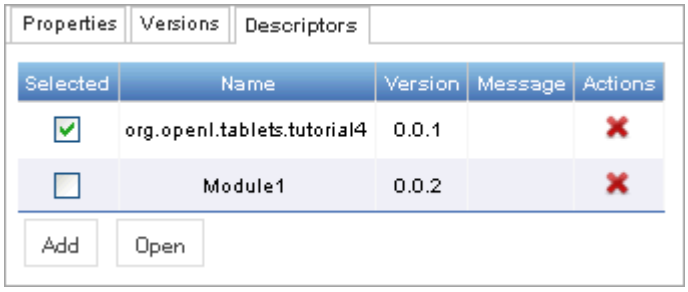


Figure 398: Deployment descriptors

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

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

Deploying a Deployment Project

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

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

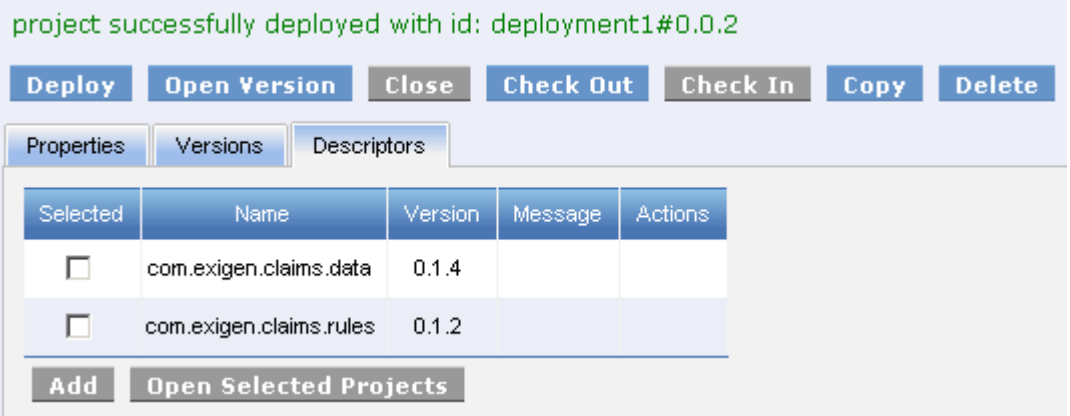


Figure 409: Deployment message

Opening Deployed Projects

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

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

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

   The selected project versions are opened in repository editor.

Redeploying Projects

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

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

1. In the project tree, select the modified rule project.

2. In the right pane, click **Redeploy.**

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

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

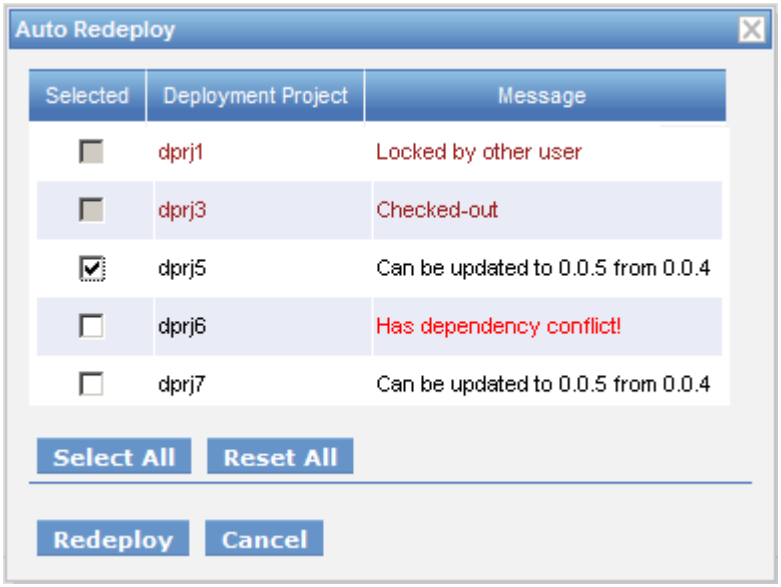


Figure 60: Redeploying a project

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

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

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

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

   Update and redeployment results are displayed in the user interface.

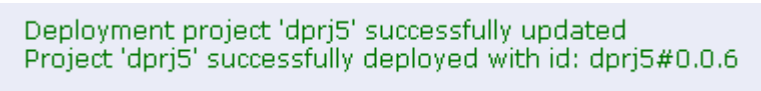


Figure 61: Redeployment results

Comparing Project Versions

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

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

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

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

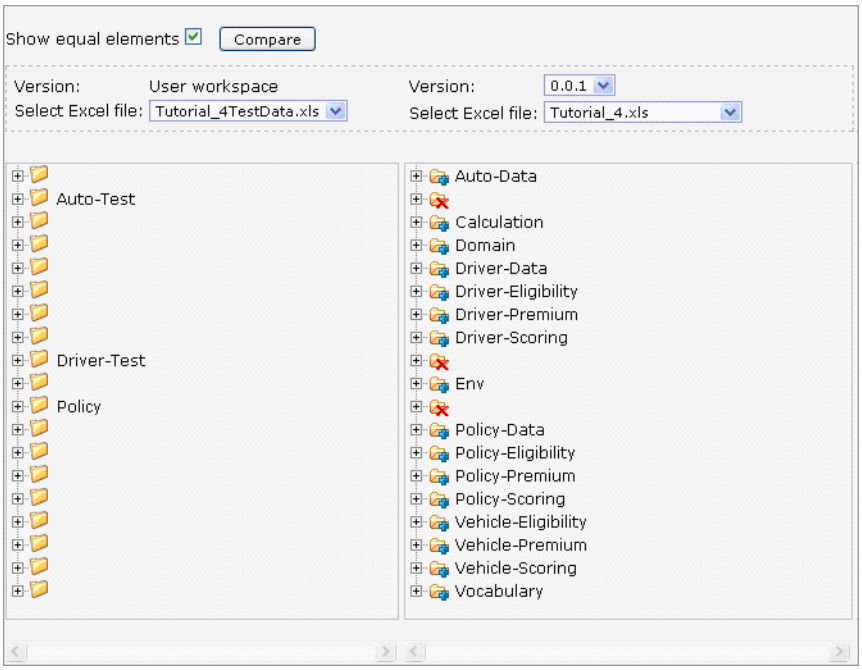


Figure 62: Comparing project versions

   Green entries indicate new elements and red crosses indicate deleted or nonexistent elements.
3. To compare the current project version with a different version, in the **Version** list box, select the version number.

Using Administration Tools

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

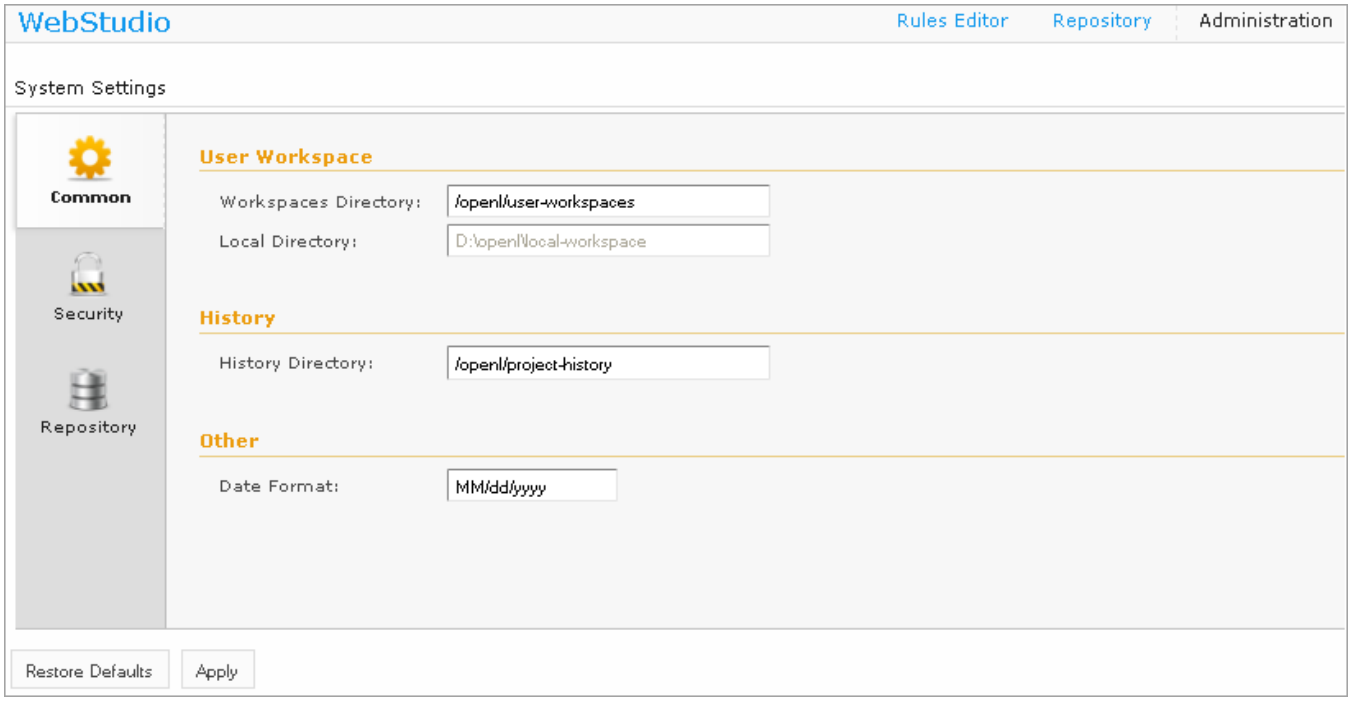


Figure 63: OpenL Tablets WebStudio Administration

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

The following topics are included in this section:

- Common settings
- Security settings
- Repository settings

Common Settings

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

User Workspace

Defines where the user's projects should be located:

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

History

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

Other

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

Security Settings

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

Repository Settings

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

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

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

Index

U

V