OpenL Web Studio User's Guide

Exigen Process Backbone® 7.0.2

Exigen® Decision Services

Document number: TP_EPB_7.0.2_OpenLWS_User_1.2_JO
Revised: 09-29-2008
EXIGEN CONFIDENTIAL – FOR AUTHORIZED USERS ONLY

Important Notice

Trademark

Support contact information			
Telephone:		Website:	http://www.exigengroup.com/
1-866-4EXIGEN	North America	Fax:	1-506-674-4014
1-506-674-4025	All other countries	Email:	support@exigengroup.com
1-800-508-192	Australia (within Australia only)		
0-800-448-898	New Zealand (within New Zealand only)		

Table of Contents

Preface

This preface is an introduction to the *OpenL Web Studio OpenL 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 Web Studio is closely related to OpenL Tablets technology. For information on OpenL Tablets, 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 Web Studio

This section introduces the main OpenL Web Studio concepts.

The following topics are included in this section:

What is OpenL Web Studio?

OpenL 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 Web Studio. OpenL 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 table wrappers, and running Ant scripts, users must use Eclipse.

Working with Projects in OpenL Web Studio

OpenL Web Studio is intended for a multi-user environment. It provides a centralized storage of rule projects called **design time repository.** Design time repository is stored on the OpenL Web Studio server and can be accessed by all users. However, users cannot modify projects directly in design time 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 design time repository creates a copy in user's workspace, a specific location on the OpenL 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 design time 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 design time repository. A new version of the project is created in design time 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 design time 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 design time repository. Closed projects can be browsed in repository editor but are not available in rule editor.

OpenL Web Studio Components

OpenL Web Studio consists of the following main components:

OpenL 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 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 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 design time 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 design time repository • checking out, checking in, opening, and closing projects • modifying project structure and properties • managing project versions and dependencies • copying and deleting projects in design time repository • managing and tracing project deployments For detailed information on using rule editor, see Chapter 4: Using Repository Editor.

OpenL Web Studio components	
Component	**Description**
Design time repository	Centralized storage of rule projects accessible by all OpenL Web Studio users. Projects uploaded to design time repository are visible to other users. Design time 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.
Production time repository	Centralized storage of final rule projects delivered to the production environment where solution applications use them. Projects can be deployed to production time repository from design time repository using deployment projects. A **deployment project** is a specific OpenL Web Studio project type. It stipulates which rule projects and project versions must be deployed to production time repository. Deployment projects are saved and versioned so that developers can identify which specific rule project versions are deployed. In the context of Exigen Process Backbone, the Exigen Studio run-time repository is used as production time repository.
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 Web Studio supports a security mechanism restricting access to certain product functions based on user access rights. Each OpenL Web Studio user is identified by a unique name. Users can have varied levels of access in OpenL Web Studio. For example, system administrators usually have full access to all OpenL Web Studio functions, whereas business users only have access rights to modify business rules.

Usually, when a user opens OpenL 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.

User Perspectives

This section describes how OpenL Web Studio is employed by different user roles.

The following user roles are described in this section:

- Business User's Perspective
- Developer's Perspective

Business User's Perspective

The following is a typical procedure in OpenL Web Studio from a business user's perspective:

1. Open OpenL Web Studio in the web browser.

   Rule editor appears.

2. In rule editor, select the required module.
3. If the required module is not available in rule editor, proceed as follows:
   - Switch to repository editor, locate the required rule project, and open it.
     OpenL Web Studio creates a working copy of the selected project in user's workspace in read only mode.
   - Switch back to rule editor and select a module in the opened project.
4. Browse module tables as required.
5. If rules in the module must be modified, check out the project.
   As a result, rules in the module become editable. Other users cannot check out the project while it is checked out by the current user.
6. Modify module tables as required.
7. If required, run unit tests to ensure data validity.
8. Check in the modified rule project so that the changes are visible to others.

The following diagram shows the involved OpenL Web Studio components and activities from a business user's perspective:

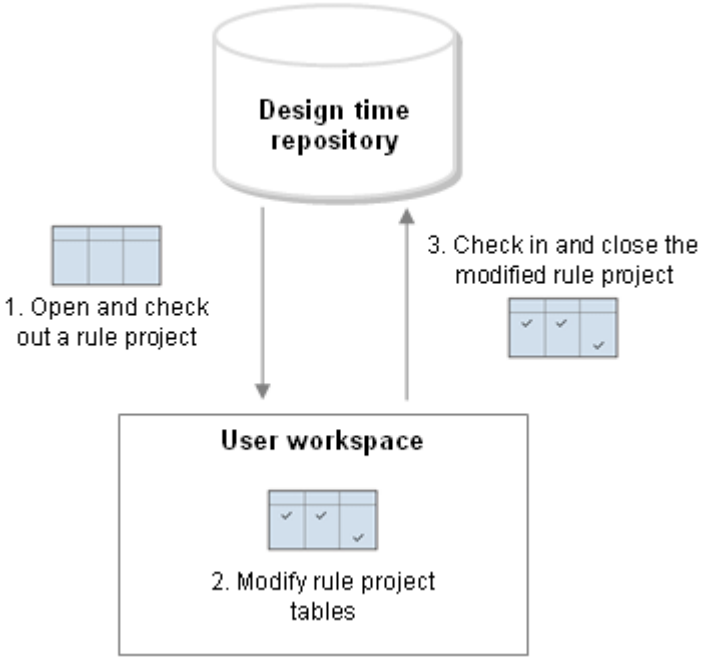


Figure 1: Business user activities in OpenL Web Studio

Developer's Perspective

Developers use OpenL Web Studio for the following main activities:

- Import projects from Eclipse to design time repository.
- Modify technical attributes of decision and data tables.
- Create and run unit tests.
- Measure performance using benchmarking.

- Manage projects in design time repository.
- Deploy projects from design time repository to production time repository.

The following diagram shows the involved OpenL Web Studio components and activities from a developer's perspective:

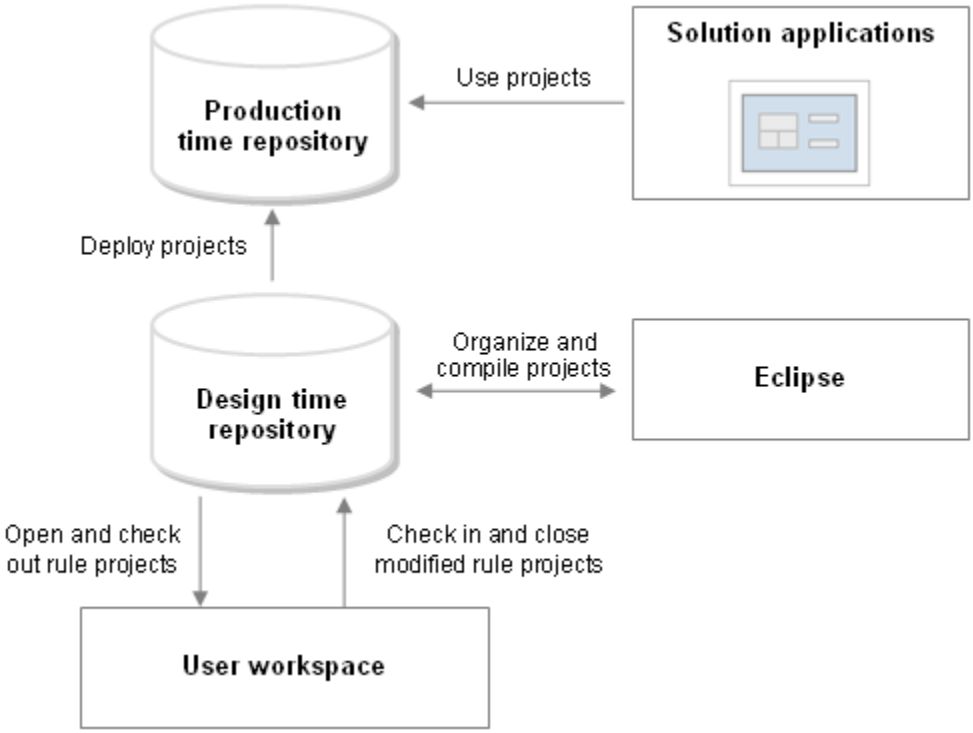


Figure 2: Developer activities in OpenL Web Studio

Chapter 2: Getting Started

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

The following topics are included in this section:

Logging In to OpenL Web Studio

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

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

   The OpenL 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 3: 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 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:

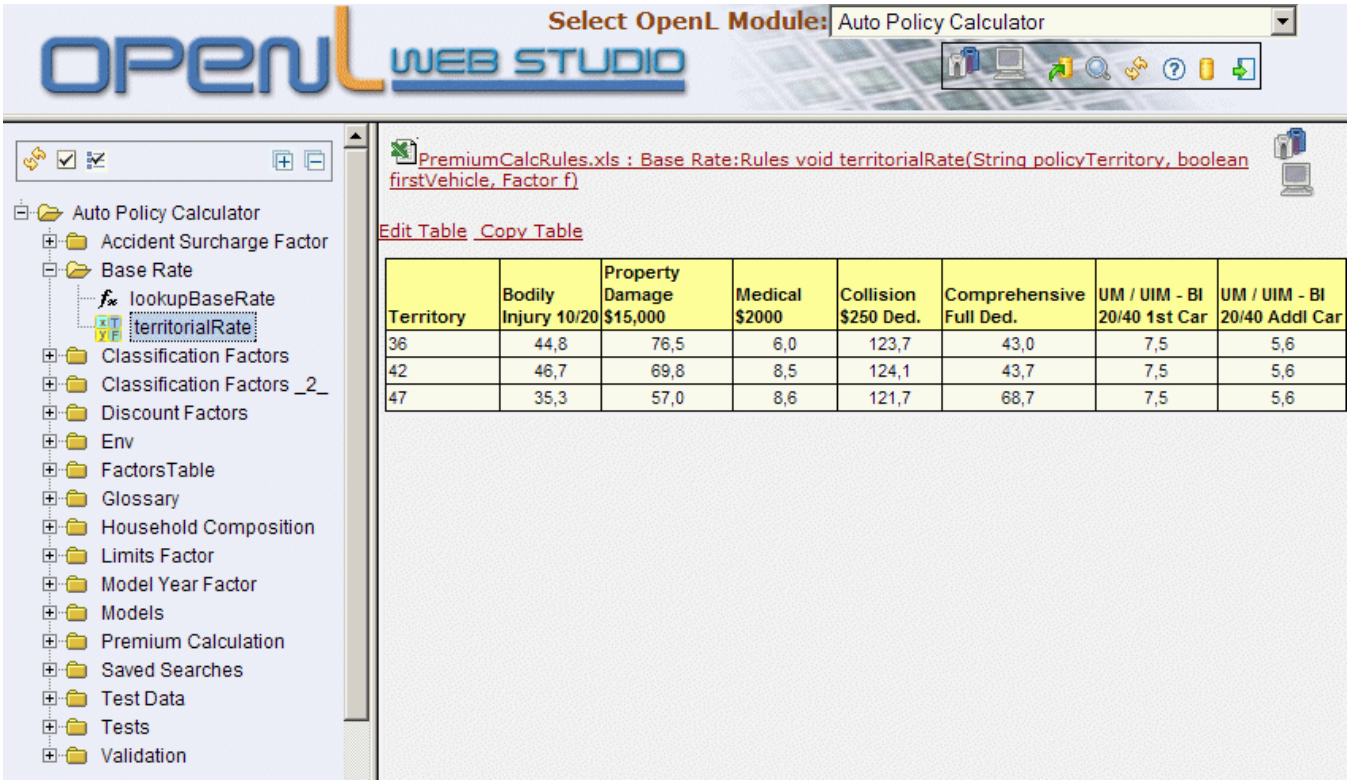


Figure 4: OpenL Web Studio rule editor

Rule editor displays one module at a time. To switch between modules, the user must select a module in the **Select OpenL Module** 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 right 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 upper part of the window contains a toolbar with the following buttons:

Rule editor toolbar buttons	
Button	**Description**
	Switches rule editor to business view. If a business view is already opened, this button switches between different business views. For information on view modes, see View Modes.

Rule editor toolbar buttons	
Button	**Description**
	Switches rule editor to developer view. For information on view modes, see View Modes.
	Opens a window for uploading projects from user's workspace to design time repository. For information on this operation, see Uploading Projects to Design Time Repository.
	Opens the search window. For information on performing searches, see Performing a Search.
	Refreshes OpenL Web Studio with latest changes in user's workspace.
	Opens OpenL Web Studio help.
	Switches user interface to repository editor. For general information on repository editor, see Repository Editor.
	Logs the user out of OpenL Web Studio.

View Modes

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

Project display modes in rule editor	
Mode	**Description**
Business view	Project view is 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. The following is an example of a module tree displayed in business view:

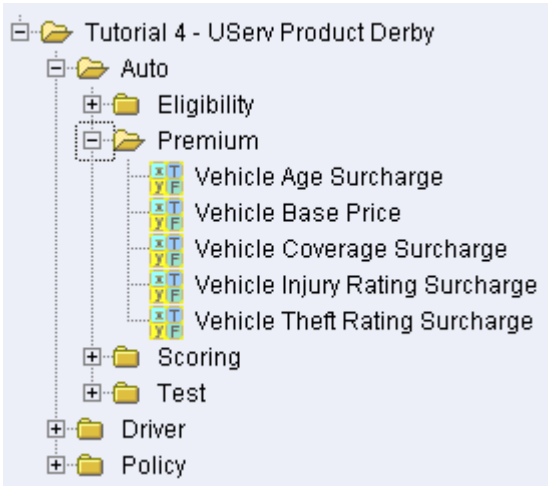


Figure 5: Module tree in business view

OpenL Web Studio hides various technical table details when a table is opened in business view. The following is an example of a table opened in business view:

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

Figure 6: Rule table in business view

OpenL Web Studio provides three slightly differing business views, mainly in the depth of the module tree. To switch between the different business views, a user must repeatedly click the business view button.

The user can switch to the business view by clicking **Business View**  **.**

Project display modes in rule editor	
Mode	**Description**
Developer view	Project is displayed in a way convenient to developers with module tree elements organized by type rather than logic. The following is an example of a module tree displayed in developer view:

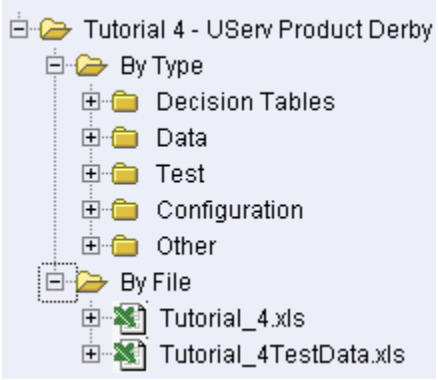


Figure 7: Module tree in developer view

OpenL Web Studio shows various technical table details important for integration with code when a table is opened in developer view. The following is an example of a table opened in developer 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 8: Rule table in developer view

User can switch to developer view by clicking **Developer View** .

Repository Editor

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

Figure 9: OpenL 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.

User can switch to rule editor by clicking .

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 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:

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 **Select OpenL Module** 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 describes the following tasks that can be performed on projects in rule editor:

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**  **.**

Uploading Projects to Design Time Repository

If user's workspace contains local projects not available in design time repository, the projects can be uploaded to design time repository directly from rule editor.

To upload local projects to design time repository, in the toolbar, click **Upload projects to repository**  **.**

Any local projects present in user's workspace are displayed in a table.

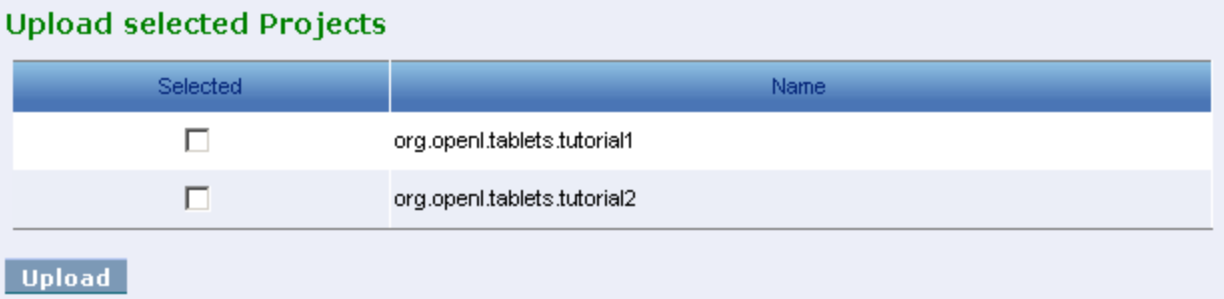


Figure 10: Uploading local projects to design time repository

Select check boxes for projects to be uploaded and click **Upload.**

Note: The same result can be achieved in repository editor by copying local projects with the same name. For information on copying projects, see Copying a Project.

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.
	Data table.
	Data type table.
	Method table.
	Unit test table.
	Run method table.
	Environment table.
ab	Table not corresponding to any preceding types. Such tables are considered comments.

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.

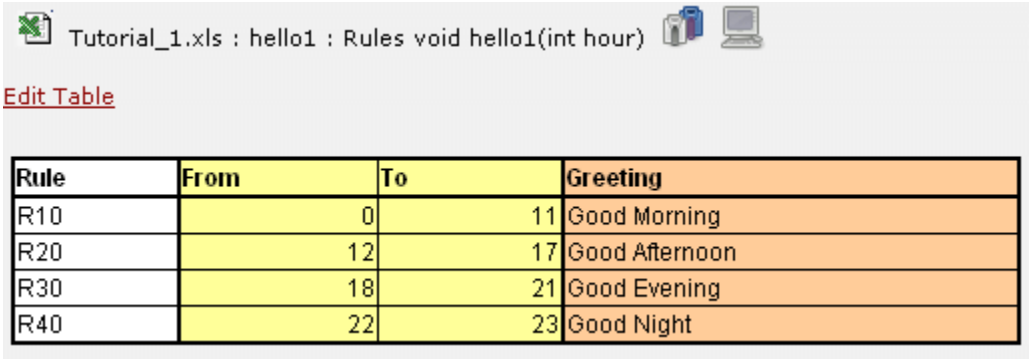
To view contents of a particular table, in the module tree, select the table. The table is displayed in the right pane. If the project is not checked out, the table can be viewed but not modified.

Modifying Tables

OpenL 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 right pane in read mode.



Rule	From	To	Greeting
R10	0	11	Good Morning
R20	12	17	Good Afternoon
R30	18	21	Good Evening
R40	22	23	Good Night

Figure 11: Table opened in OpenL Web Studio

3. If required, to switch to the business view or developer view, click the appropriate button above the table.
4. To switch the table to edit mode, perform one of the following steps:
   - Above the table, click **Edit Table.**
   - Hover the mouse pointer over the table and click **Edit.**

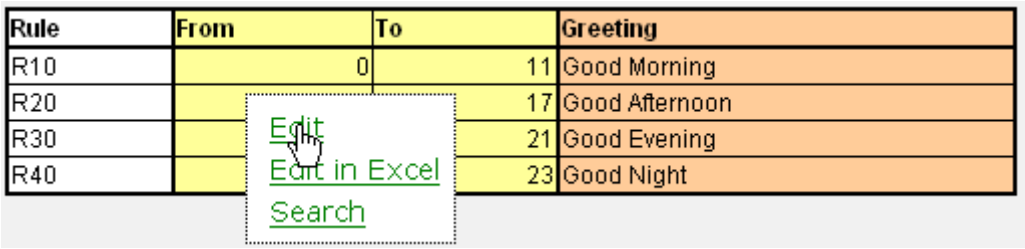
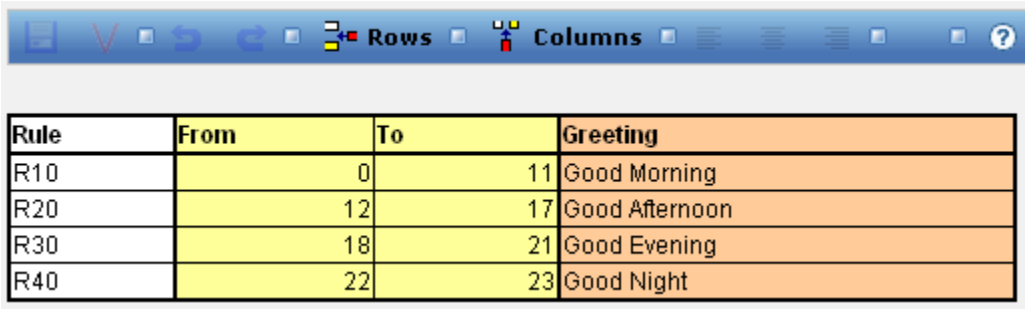


Figure 12: Switching to edit mode

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

Note: Alternatively, the table can be modified in Excel. However, such functionality is not always available in the solution and it does not offer the advanced editing and testing tools provided by OpenL Web Studio.

The table is switched to edit mode.



Rule	From	To	Greeting
R10	0	11	Good Morning
R20	12	17	Good Afternoon
R30	18	21	Good Evening
R40	22	23	Good Night

Figure 13: Table in edit mode

The edit mode provides the following buttons above the table:

Table editing buttons	
Button	**Description**
	Saves changes in table.
	Validates correctness of entered data if validation is available.
	Reverses last changes.
	Reapplies reversed changes.
	Opens menu for managing rows. By using menu options, user can add, delete, or rearrange rows.
	Opens menu for managing columns. By using menu options, user can add, delete, or rearrange columns.
	Aligns text in currently selected cell with left edge.
	Centers text in currently selected cell.
	Aligns text in currently selected cell with right edge.

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  .

Performing a Search

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

To open the search window, perform one of the following steps:

- In the toolbar click **Search**  **.**
- If a table is displayed, hover the mouse pointer over the table and click **Search.**

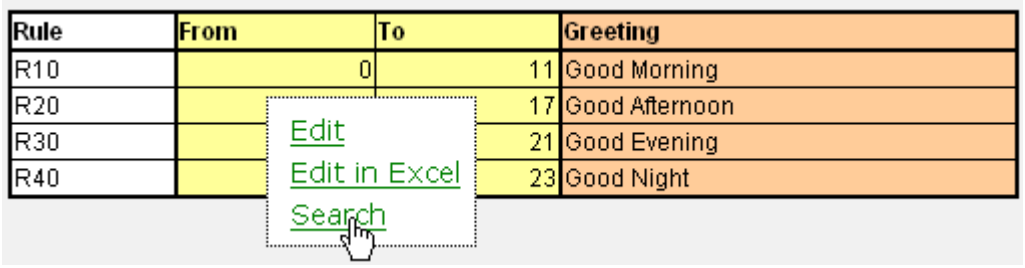


Figure 14: Opening a search window

The search window appears.

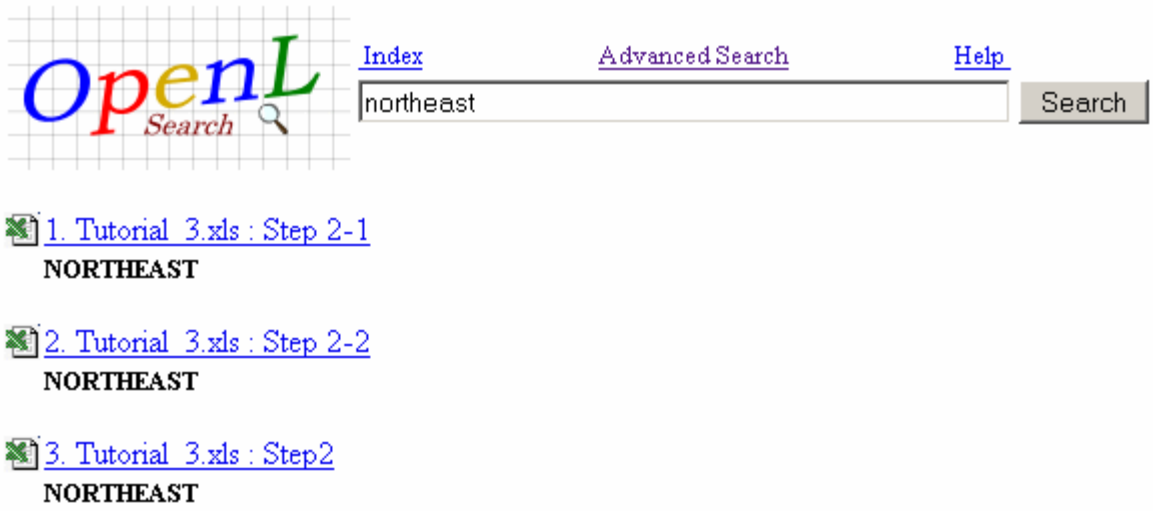


Figure 15: Search window

The search window provides the following search modes:

- Simple Search
- Advanced Search
- Index

Simple Search

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

To perform a simple search, proceed as follows:

1. Open the search window as described in Performing a Search.
2. In the search field, enter the word or phrase and click **Search.**

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

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.

To perform an advanced search, proceed as follows:

1. Open the search window as described in Performing a Search.
2. To switch to the advanced search mode, click **Advanced Search.**

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

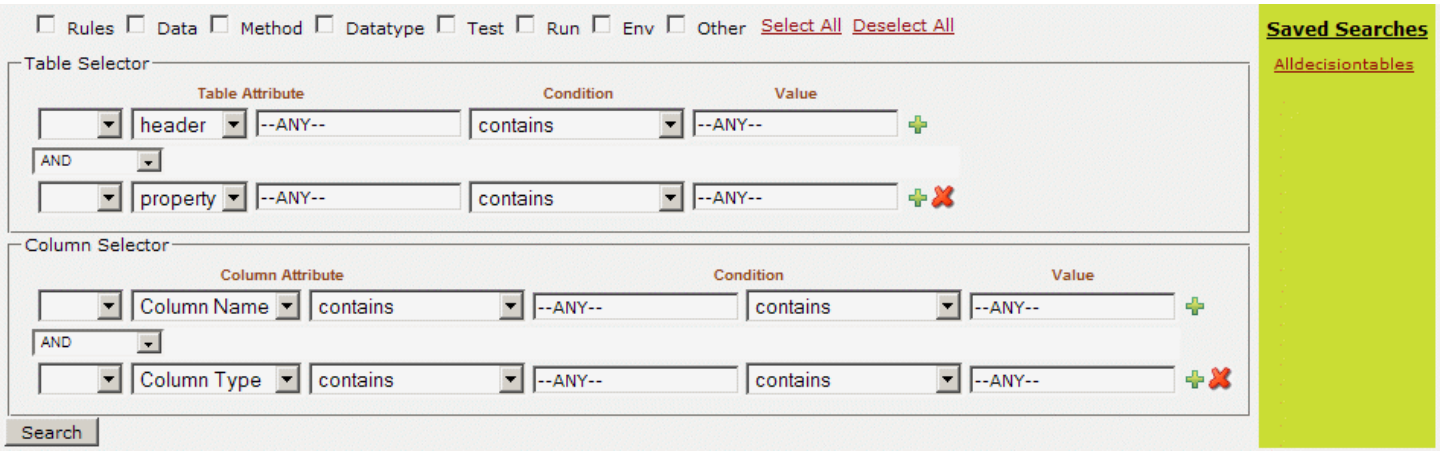


Figure 16: Advanced search

If search criteria was previously saved, it is displayed on the right side. A user can load previously saved search criteria by clicking the appropriate name in the **Saved Searches** list.

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

   OpenL Web Studio displays tables matching the search criteria below the advanced search window.
6. Optionally, to save the search criteria for future use, click **Save this search** and enter the name for the collection of search parameters.

   The user can switch to table editing mode directly from the advanced search window by hovering the mouse button over a table in search results and selecting **Edit.** For information on modifying tables, see Modifying Tables.

Index

The index displays an alphabetized list of all entities available in the module.

To display the index, proceed as follows:

1. Open the search window as described in Performing a Search.
2. To switch to the index mode, click **Index.**

   The index is displayed.

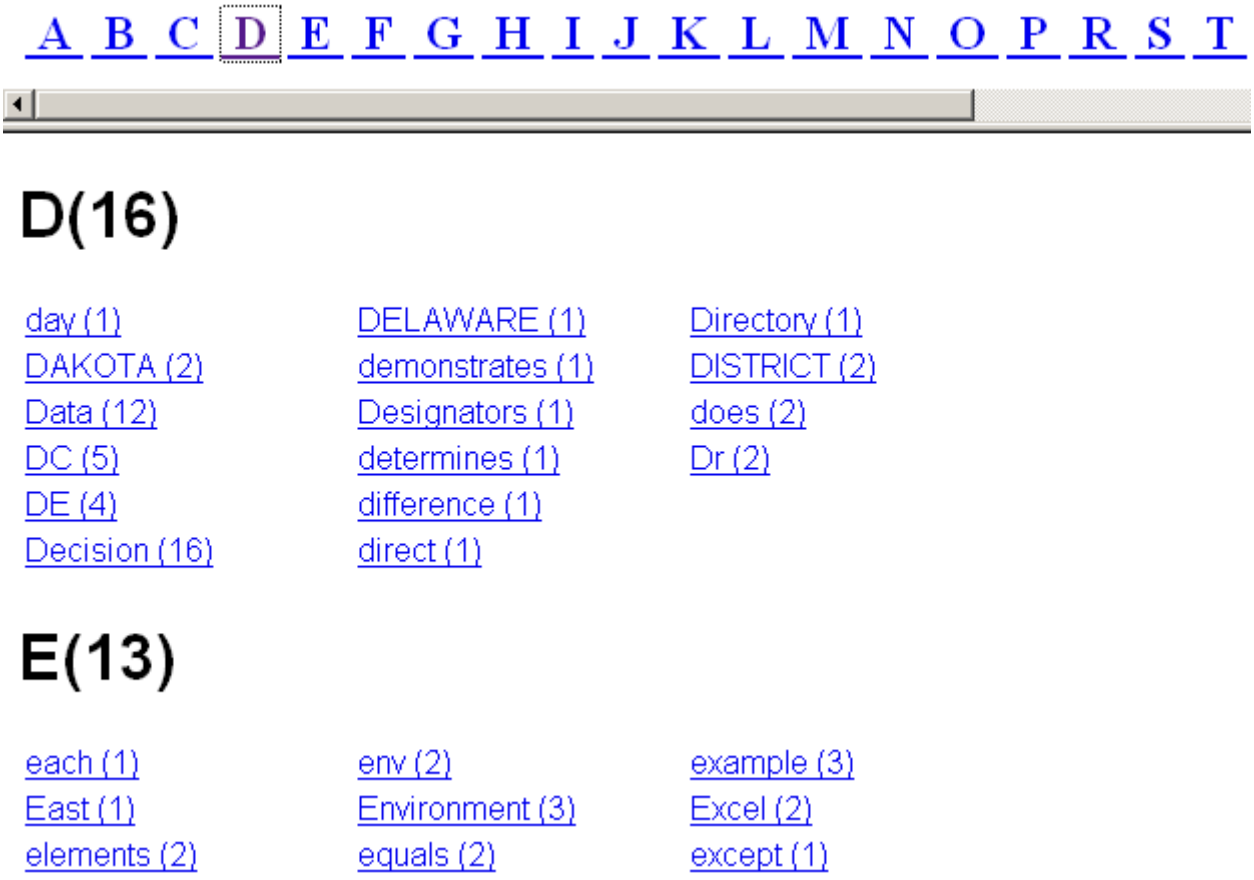


Figure 17: Index

3. To perform a simple search for a particular index entry, click the entry link.
   OpenL Web Studio displays search results in form of links to Excel files containing the index entry.

Chapter 4: 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**
Rules Projects	Contains OpenL Tablets rule projects.
Deployment Projects	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.

<table>
<tr><th colspan="2">Project icons in repository editor</th></tr>
<tr><th>Icon</th><th>Description</th></tr>
<tr><td>  </td><td>Project is closed by current user but checked out by another user.
Current user cannot check out the project.</td></tr>
<tr><td>  </td><td>Project is opened by current user but checked out by another user.
Current user cannot check out the project.</td></tr>
<tr><td>  </td><td>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.</td></tr>
<tr><td>  </td><td>Project is marked for deletion.
In OpenL Web Studio, 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 time repository. After this phase, the project cannot be restored.
For information on erasing a project, see Erasing a Project.</td></tr>
</table>
</td></tr>
</table>

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 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, proceed as follows:

1. In the project tree, select **Rules Projects.**
2. In the right pane, click **Upload Project.**

The **Upload Project** window appears.

Figure 18: Uploading a project

3. In the **File** field, select the ZIP file containing the rule project.

   Warning: The ZIP file must contain only project folders but not the root folder.
4. In the **Project Name** field, enter name by which the project must be represented in design time repository.
5. Click **Upload.**

Creating a Project

OpenL 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 **Rules Projects.**
2. In the right pane, click **Create New Project.**

   The **New Project** window appears.

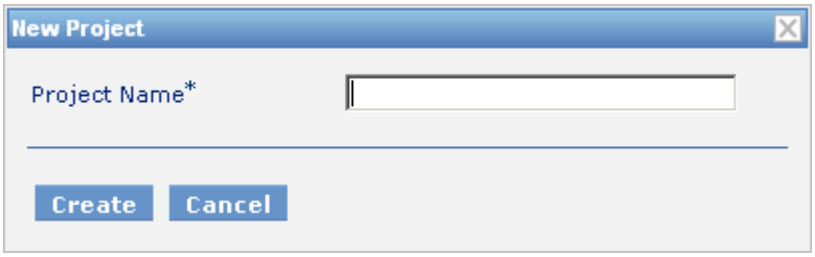


Figure 19: Creating a new project

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

   An empty project is created in design time repository. Initially, it does not contain any folders or files. Its structure must be manually constructed.
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.

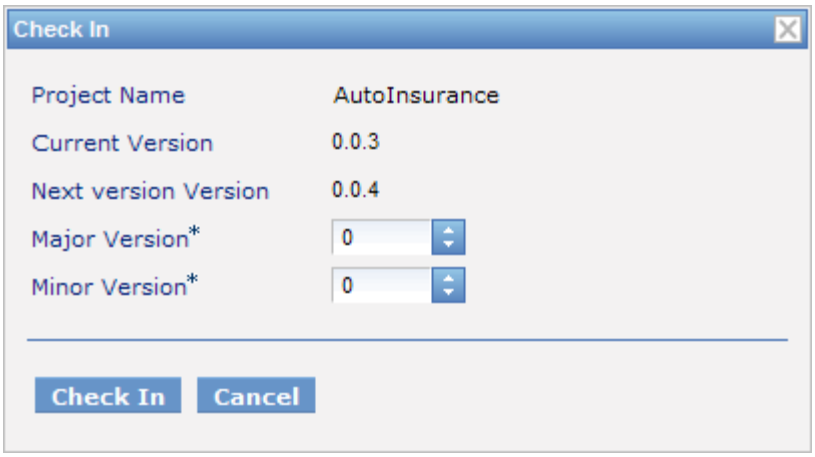


Figure 20: Checking in a project

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 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 21: 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.

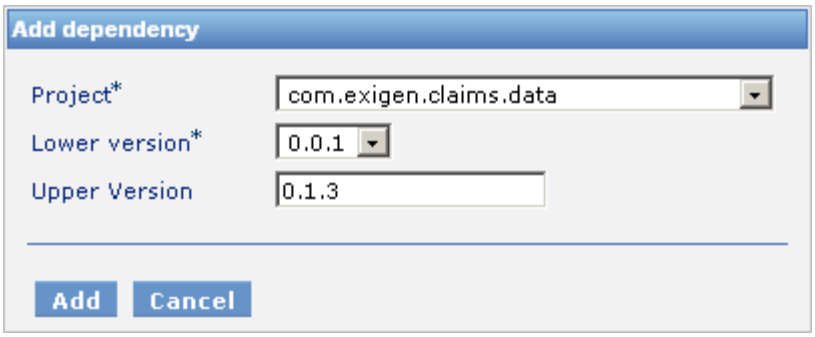


Figure 22: Defining a new dependency

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

When a project is checked out, its properties and contents can be modified.

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.

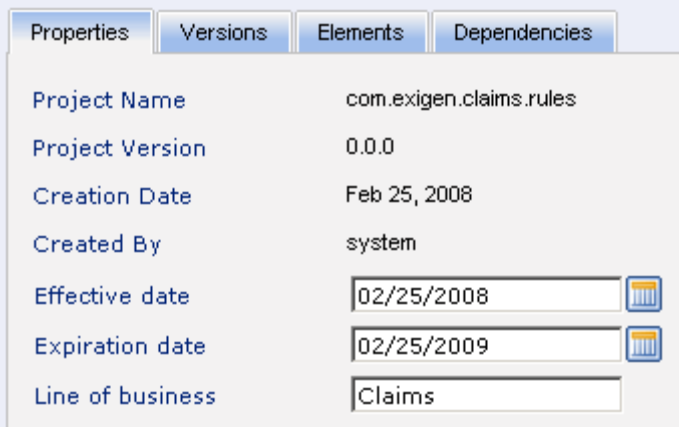


Figure 23: Project properties

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

The following properties can be modified by a user:

Manually edited 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.

The properties in the preceding table are used by the OpenL Tablets rule engine at run-time to determine which rules are valid for a particular request.

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 24: Uploading a file

4. In the **File** field, select the file to be uploaded.
5. In the **File name** field, enter the name of the file to be used in design time repository.
6. 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.**

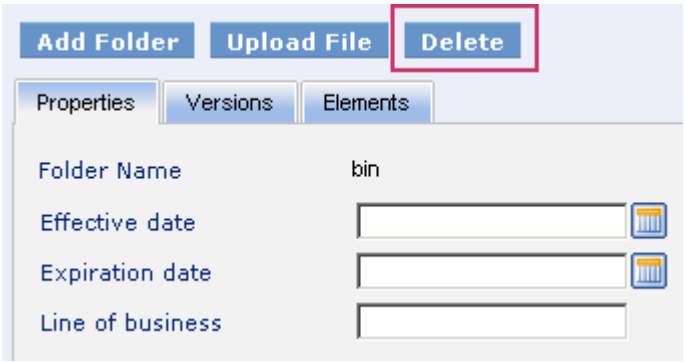


Figure 25: Deleting a project element

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

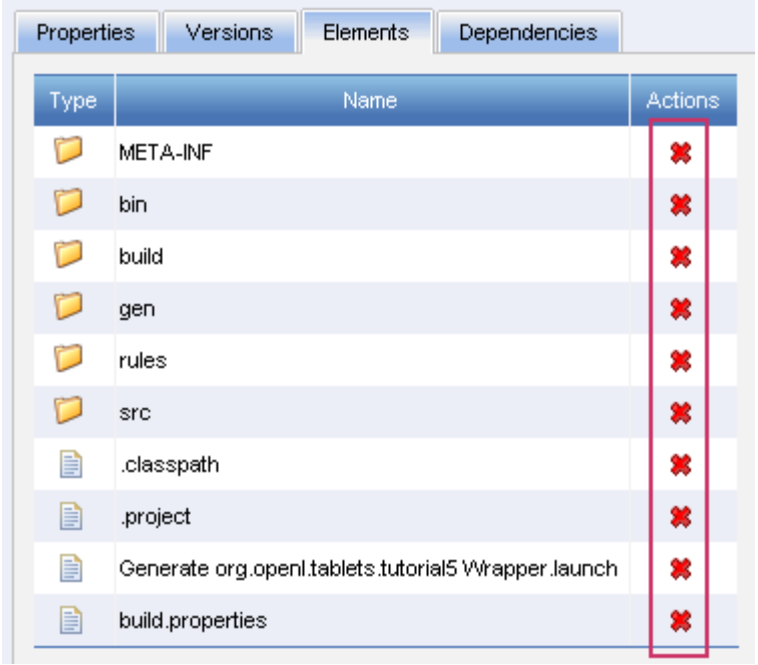


Figure 26: Deleting project elements in the Elements tab

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 the same or 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 **Rules 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.

Erasing a Project

Erasing a project permanently removes it from design time repository.

Warning: Erased projects cannot be restored.

To erase a project, proceed as follows:

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

Deploying Projects

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

The following topics are included in this section:

- Creating a Deployment Project
- Defining Deployment Project Descriptors
- Deploying a Deployment Project
- Opening Deployed Projects
- Redeploying Projects

Creating a Deployment Project

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

Deployment projects are listed in the project tree, in the **Deployment Projects** 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 **Deployment Projects** category.
2. In the right pane, click **Create New Deployment Project.**
3. In the **New Deployment Project** 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 27: 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.

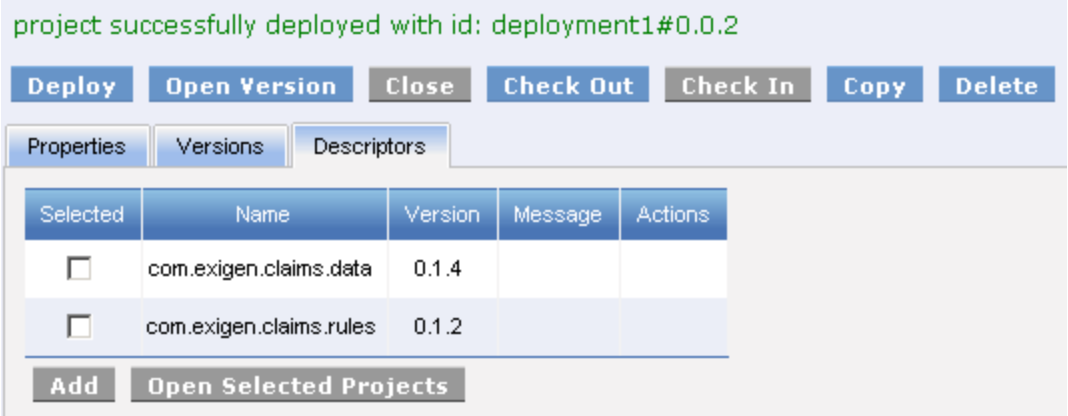


Figure 28: Deployment message

Opening Deployed Projects

Deployment projects provide the means for tracking the deployment history of project versions. OpenL 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 Selected Projects.**

   The selected project versions are opened in repository editor.

Redeploying Projects

OpenL 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.

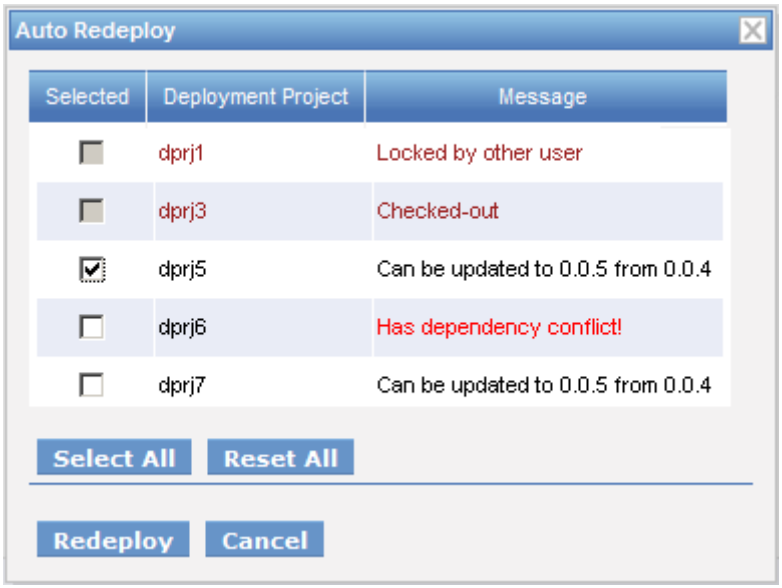


Figure 29: Redeploying a project

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.

```
Deployment project 'dprj5' successfully updated
Project 'dprj5' successfully deployed with id: dprj5#0.0.6
```

Figure 30: Redeployment results

Comparing Project Versions

OpenL 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.

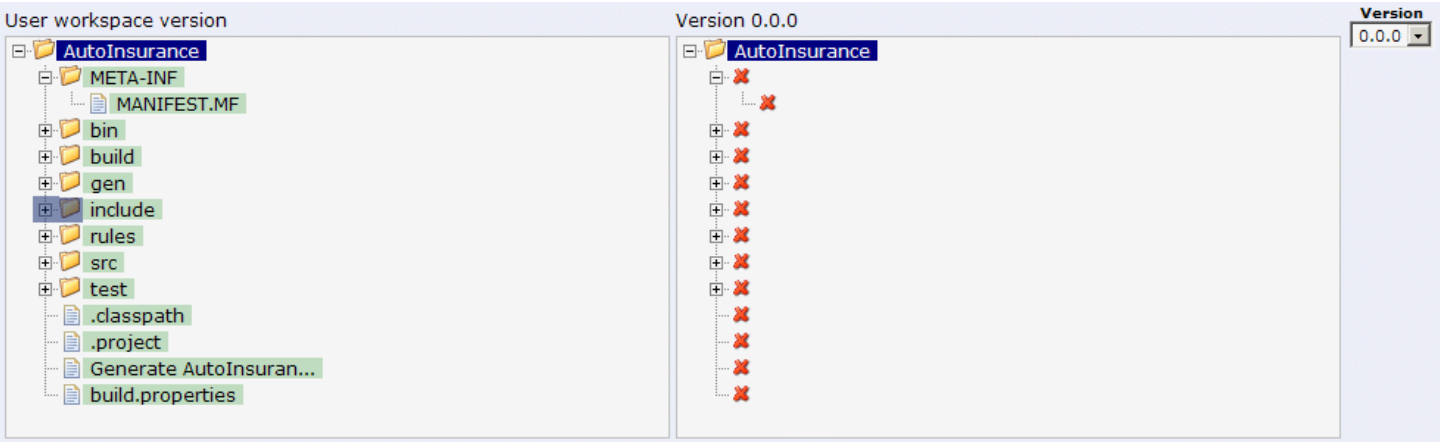


Figure 31: Comparing project versions

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.

Chapter 5: Advanced Functionality

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

The following topics are included in this section:

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 32: Decision table and its unit test table

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

To run unit tests, the following methods can be used:

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, in rule editor, above the module tree, click **Run All Tests** ☑**.** Test results resemble the following:

Tests: 9 (1)
Units: 44 (1)

Test Region Indexed

state	Expected	Result
MD	NORTHEAST	☑ NORTHEAST
OH	WEST	✖ CENTRAL
KY	SOUTHEAST	☑ SOUTHEAST
WA	WEST	☑ WEST

Test Region

state	Expected	Result
MD	NORTHEAST	☑ NORTHEAST
OH	CENTRAL	☑ CENTRAL
KY	SOUTHEAST	☑ SOUTHEAST
WA	WEST	☑ WEST

Figure 33: Results of running all project tests

Failed tests are represented by the ✖ mark. Passed tests are represented by the ☑ mark.

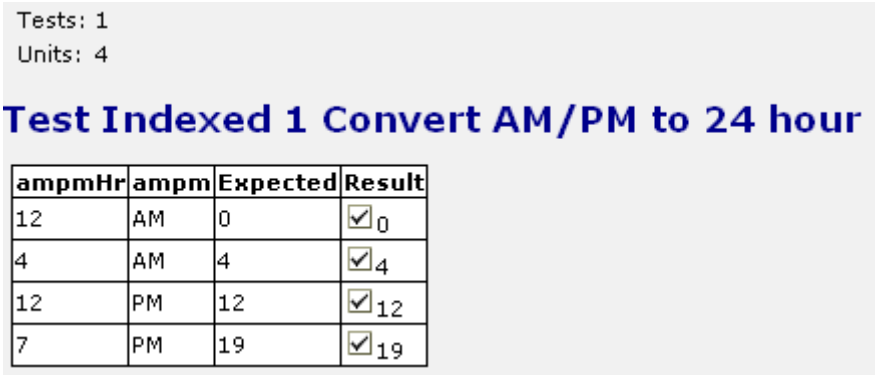
Methods for running unit tests	
Method	**Description**
Execute all tests for a single decision table	System executes all test runs for one particular decision table. To execute all test runs for one particular decision table, in rule editor, in the module tree, select the decision table and, in the upper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

Figure 34: Results of executing all test runs for one decision table

Validation

OpenL Web Studio provides controls for checking the validity of all decision tables in the opened module. To validate all rules in the module, click **Validate Project** above the module tree.

If errors are detected, a corresponding message is displayed and erroneous elements are identified by a red cross in the module tree.

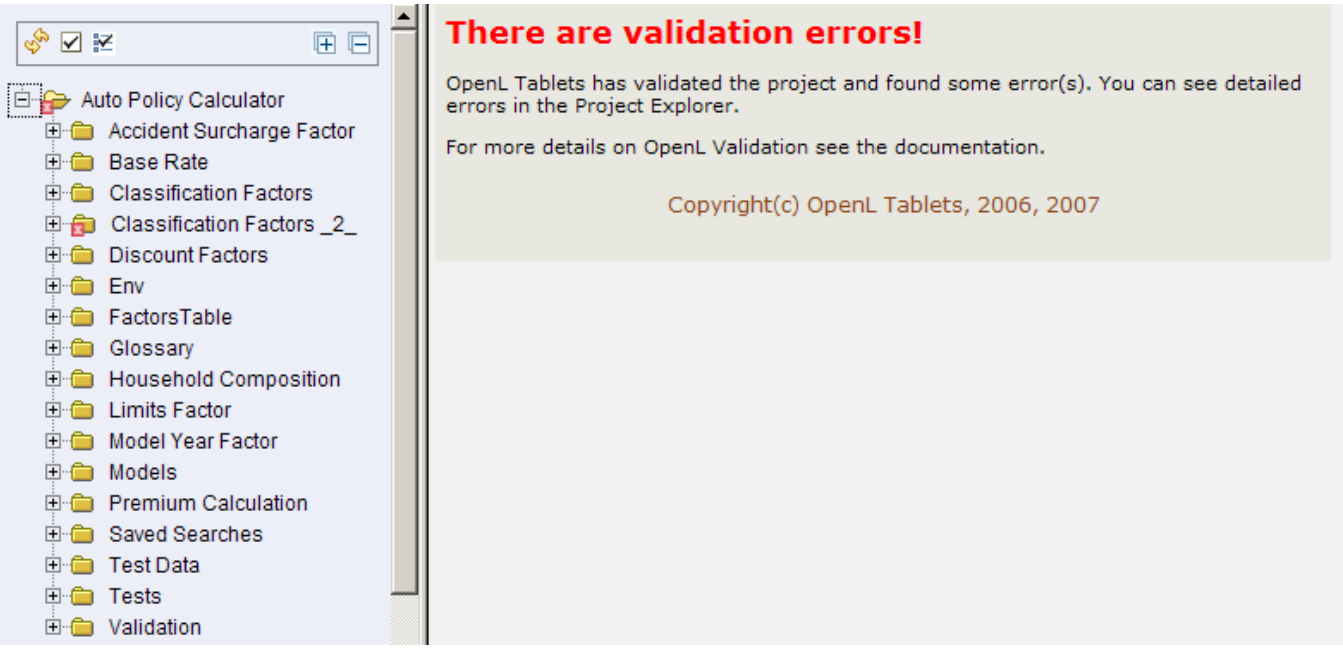


Figure 35: Validation results

Tracing

OpenL 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 right pane, click **Trace** .

The trace view resembles the following:

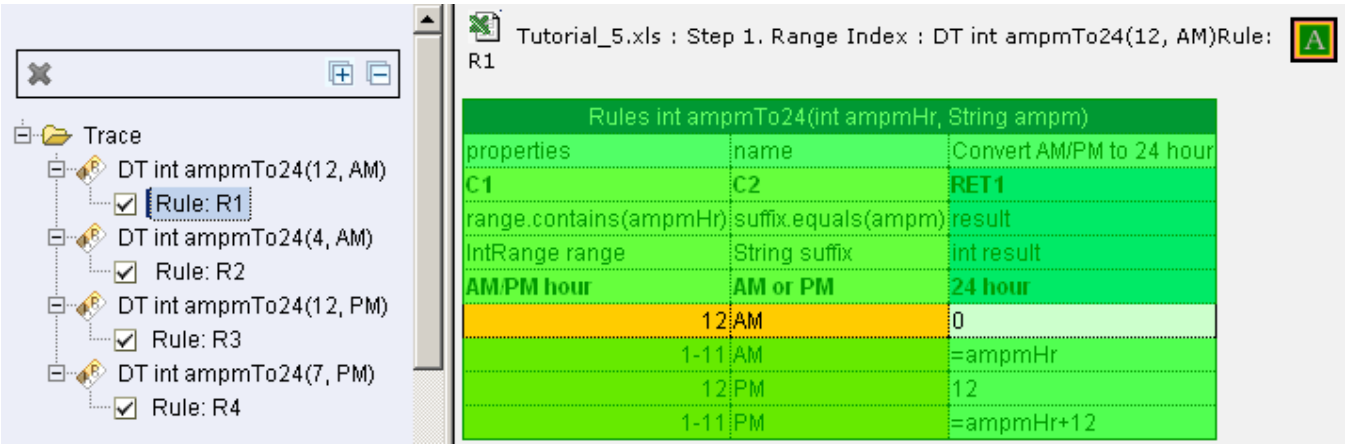


Figure 36: Tracing a rule

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 Web Studio provides benchmarking tools for measuring execution time for all appropriate OpenL Tablets elements. 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.

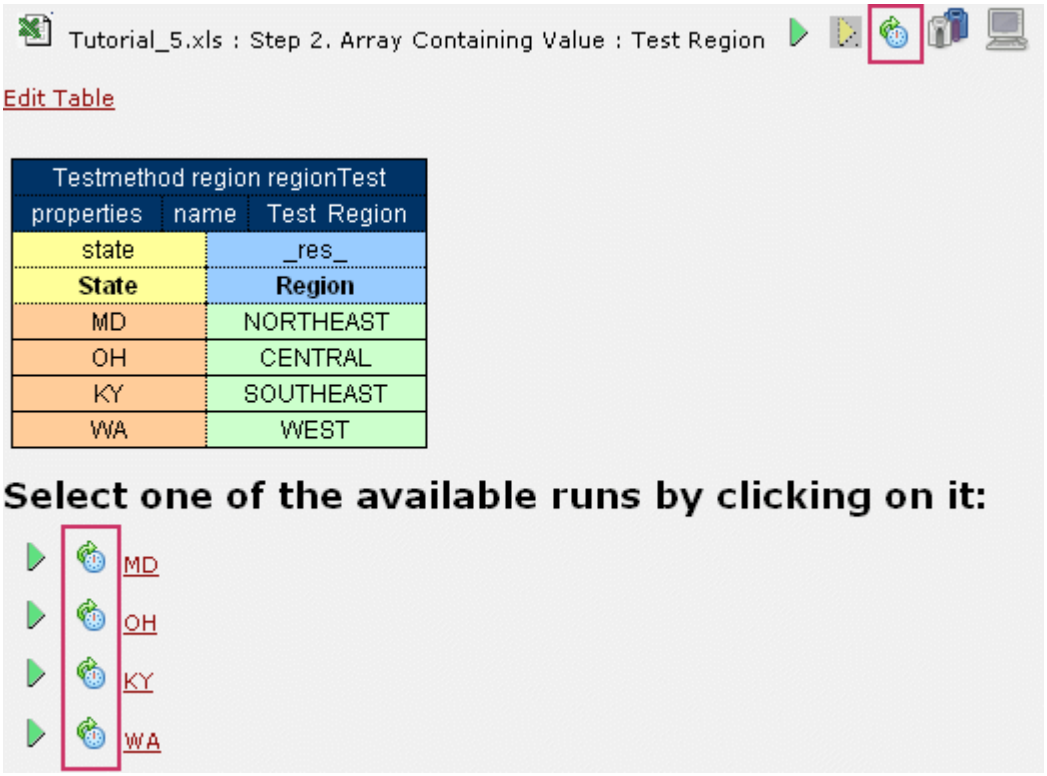


Figure 37: Controls for measuring performance

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	☑

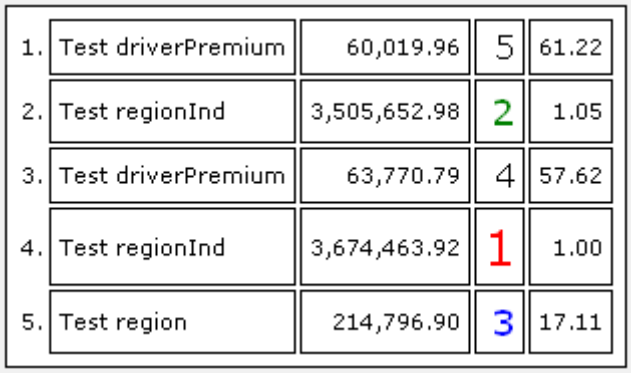
Compare Delete

Figure 38: Benchmarking results

OpenL 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.

To identify the most time consuming methods, benchmarking results can be compared. To compare results, in the results table, select the required check boxes and click **Compare.**

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 39: Comparing benchmarking results